Clinical decision support systems: A discussion of quality, safety and legal liability issues

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

ABSTRACT

Developers of Clinical Decision Support Systems (CDSSs) have to date been more concerned with the efficacy of systems (e.g. measurable improvements in clinical outcomes) than with safety (e.g. potential for harmful side-effects). In future CDSS developers will be required (by the courts etc.) to acknowledge a “duty of care” covering all aspects of design, development and deployment. Experience in the transport, power and 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 an integral part of routine patient care, and before the trust of healthcare professionals, patients and other stakeholders can be gained. No single method will be sufficient for safe development and deployment; 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.

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) such as patient monitoring and reminder systems, prescribing systems, treatment management and workflow management systems 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 patient care we must also anticipate the possible risks of introducing these systems. Most new medical technologies entail new risks (e.g. unanticipated side-effects of drugs) and 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.

In the context of CDSSs, there is also a further longstanding and unanswered issue concerning legal liability: if a decision support system gives bad advice, who will be held responsible? The software designers? The providers of the medical knowledge used by the system? Or the end-users, the healthcare professionals who are responsible for the final clinical decision? No-one seems to know: so far as we can establish, there is no case law to establish the relevant precedents in the USA, Europe or elsewhere. The medical informatics community should itself anticipate possible legal liabilities that might result from the use of their technologies 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. Further, 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.

1 The nature of this paper is that it includes a number of summaries of methods, options, constraints and so forth. These are presented in a number of bulleted lists and we hope that this does not seriously impact readability.
QUALITY AND SAFETY ENGINEERING

CDSS developers have much to learn from current quality practices in software engineering, particularly in software safety engineering. It is well known that software products are increasingly developed within a “development lifecycle”, which covers the design, implementation and ongoing maintenance of software, particularly software that is intended for use in 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 [IEC 61508] as a basis for establishing best practice in the design and development of safety-critical systems. However, neither the International Standards Organisation nor the International Electrotechnical Commission have the authority or resources to enforce their standards (e.g. by any audit or certification process) so the current position is that the R&D community and industry must police themselves.

In addition, no current standard can unequivocally 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 organisation 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). In short, safety is seen as a trade-off between maximising safety and investing a level of resources 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 in this area, 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 health professionals who may use it in good faith in their clinical practice.
- This duty of care will probably extend to commercial licensees/suppliers of these technologies (i.e. even if they 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 have been developed according to generally agreed quality and safety standards.
- Disclaimers that attempt to limit liability by, say, restricting “proper” use of CDSSs will have limited status in law and would not by themselves be sufficient to protect a developer or supplier from legal proceedings.
- Suppliers may be able to limit their exposure by taking out insurance to cover for awards in the event of a mishap. However, this would not insulate them from other costs, such as a damaged reputation.

The goal of those developing CDSSs must be to maximise the quality and safety of this new technology, thereby minimising the risk of adverse events and exposure to legal action. All CDSS developers would wish to see their work put to effective use for the benefit of patients, but since absolute safety can never be guaranteed with any technology, they 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 best practice during all stages of development. Currently, there appear to be no generally accepted standards of practice that clinical software developers or suppliers of CDSSs can seek to comply with.

We have identified four primary approaches to quality and safety for CDSS technologies so far reported in the literature.

1. The use of rigorous software engineering techniques to ensure the integrity and reliability of the technology platform.
2. The adoption of systematic development life cycles for creating and maintaining the medical content of an application and its associated scientific evidence base.
3. The application of explicit safety and hazard management techniques within the applications where this is possible.
4. The provision of comprehensive documentation to provide for quality and safety reviews by end users, technology licensees etc.

(See also discussions in Fox and Bury, 2000; Fox et al, 2001).

The position with regard to legal liability issues is less clear. Many CDSSs that are currently available (e.g. through the web) seem to depend substantially on disclaimers for their legal protection. Typical 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 another “The Software is provided “AS IS”, without any warranty as to quality, fitness for any purpose, completeness, accuracy or freedom from errors”.

Given existing consumer protection legislation in many countries, 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.

In practice, it is likely that the degree of exposure for an organisation making CDSS technologies or applications is manageable, though this will obviously depend on common norms of litigation in different countries and markets. Nevertheless, to comply with our duty of care, 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 document we consider a range of options available 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 1-4 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 which are 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 applications. At the delivery end of the process, placing strong constraints on who can have access to CDSSs and under what circumstances will inevitably curtail their use and thereby limit their benefit.

Our proposal abandons the idea that “one size fits all”; that a single approach to quality and safety can be adopted for the development of all applications. Rather we assume that a more flexible framework will be needed based on the ALARP principle. This places a clear duty of care on developers and suppliers while permitting them to establish reasonable rules for limiting the resources to commit to 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 methodical investigation of the hazards and operational problems to which a technological system can give rise, 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. The purpose of the analysis is to classify the proposed application into one of a number of categories. We propose criteria for four risk levels, but these should be viewed as tentative and are offered for discussion only as a basis for further refinement.

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 could be 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, pre-existing 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 future problems that will require management (e.g. by failing to record or inform other carers about actions taken)

Risk level 4:

There are no identifiable consequences for patient mortality or morbidity of 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 (the software which is used to build the clinical application) and/or the specific clinical application (the medical knowledge content). The following quality methods are applicable to both:

1. Systems should be designed, implemented, tested and documented using generally recognised 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 be nominated to sign off 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 judgement, 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.

A computer-based representation of medical knowledge cannot, in principle, be proved to be clinically comprehensive or objectively valid; it can only attempt to formalise 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, and this can be achieved, at least partially, by automatic means. In addition content will need to be endorsed by professional clinicians for the foreseeable future so it should also be a requirement that the content is humanly legible, so far as possible, and can be effectively reviewed by appropriate specialists and end-users.

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). Methods for quality control of medical knowledge bases may include:

1. Automated analysis to find internal inconsistencies, gaps, redundancies, ambiguities etc. (e.g. based on syntax-directed verification techniques).
2. Peer review by competent individuals. The review may include static assessment of content (e.g. reading the knowledge base) and dynamic assessment (e.g. testing the performance of the application against standard patient datasets).
3. All content should be available in a legible form for review by the end users of the system, both in static form (e.g. as text) and dynamic form (e.g. as explanations of any decision or recommendation which is made for a specific patient).
4. Provision should be made for end-users to report queries and problems to the application developers as easily as possible.

Safety and hazard management

Safety is not the same as quality. 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 offered may not take into account atypical patient circumstances (e.g. unusual combinations of conditions; local lack of resources). In a clinical application in which there are safety considerations, therefore, an explicit protocol should be adopted which provides some assurance that the design and implementation of
Quality and safety of clinical decision support systems: A draft protocol for discussion

Such systems minimises avoidable hazards to patients or others, and makes provision for managing unavoidable hazards that are known.

In the remainder of this discussion we set out a range of techniques for quality and safety assurance without attempting to decide in detail which methods are appropriate at which risk levels, which we believe can only be established as a result of open discussion.

SAFETY BY DESIGN

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

- A comprehensive HAZOP may be carried out alongside the software requirements specification to detail situations or events that might be associated with increased patient mortality or morbidity. Each such situation represents an obligation on system developers to accommodate the possible hazard in the 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 principle hazards, design choices and associated safety arguments, which have been considered in developing the CDSS (see section on documentation below).

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 the possibility of adverse events occurring in clinical operation. Systems could, for example, be misused or used in inappropriate or unforeseen situations. For these reasons, further obligations should be added to minimise, 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 right of access

When an application could potentially be used inappropriately, an important option is to be able to limit access to suitably qualified users. There are many possible access control options, including the following in increasing levels of stringency:

1. Unrestricted open access.
2. Limiting access to users who explicitly accept terms and conditions of use.
3. Limiting access to a specific class of user whose qualifications can be verified (e.g. medical practitioners whose current professional registration has been confirmed).
4. Limiting access to named individuals and/or organisations 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 keep a detailed record of the operational use of some applications, including:

- A clinical audit trail. The application may record all medically significant information and decisions which may need to be reviewed by appropriately qualified auditors in order to assess the appropriateness of all recommendations, decisions taken, clinical orders issued etc.
- An operations audit trail. The application records all significant internal operations and external transactions (data acquisition, system messages etc.) in a time-stamped format and using a mechanism that is secure.

“Guardian” functions

For certain classes of application it may be desirable and practical to include within the application:

- The capability to monitor the use of a decision support system in order to flag atypical and/or possible adverse events
- The capability to intervene if it appears that an inappropriate decision or action is to be taken, 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”. The safety case will normally include:

- A description of the method and scope of the HAZOP analysis that has been carried out.
- 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 to make the application safe.
- Revision history.

A summary of the safety case should be made accessible to end-users, ideally in electronic form 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 organisation, 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 only concerned with marketing, customer liaison and so forth.

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

In short a safety culture should be established within the development team 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 organisations 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.

DOCUMENTATION

An important element of a quality and safety protocol is the provision of comprehensive and clear documentation for the system to guide end-users in understanding its behaviour and its correct operation, and technical staff who are charged with support and/or maintenance.

We suggest that all developers should consider including the following basic documentation:

- Statement of clinical purpose of the application (e.g. diagnosis of headache, prescribing for hypertension, management of depression)
- Characteristics of the population for whom the application is intended (e.g. patient group, age, gender etc).
- Inclusion and exclusion criteria to determine the patients who should be managed using the application.
- Context of use (such as out-patient department, emergency room, primary care, pharmacy, patient’s home)
- A description of the conditions under which the application may be used (e.g. during the patient encounter; before/after the patient sees the doctor; self-care etc.)
- Assumptions about availability of ancillary equipment, services etc..
- Designated users and/or skill levels required (e.g. qualified physician, nurse, receptionist, pharmacist, patient).
- Any associated documentation, on computer or in paper documents, which was used in the preparation of the application.
- Evidential status - one of: evidence-based (with references), consensus (who), local policy (where), individual clinical opinion (who), prototype (status), together with any further comments or significant qualifying remarks.
- Version control: a unique number for the version, build and release, together with its creation date and last revision date.
• Review date: A date at which the application should be reviewed for confirmation, revision or removal from service.
• Authors: Names, affiliations and contact details.
• Safety case

Finally we suggest that this documentation would ideally be easily accessible to all end-users of the application, at any time from within the runtime environment, to permit their assessment of the fitness of the system for the intended purposes.

CONCLUSIONS

Management of quality and safety of clinical decision support systems is an important but difficult challenge requiring technical, professional and organisational commitment. A policy that is overly lax could lead to patient harm and ethico-legal challenges, while one that is overly stringent will be a disincentive to developing such technologies and achieving the potential for improved patient care that they offer.

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.

[Leveson, 1995] is recommended to readers who require a comprehensive and authoritative review of current practices in software safety engineering, including general medical software.

The authors welcome comments on any aspect of this discussion.

John.Fox@cancer.org.uk
rt@openclinical.org

REFERENCES


ISO 9000. ISO 9000:2000 family of international quality management standards and guidelines. (See also http://praxiom.com/)
