A Study of a Health Enterprise Information System Part 8 - HIT REGULATION PROPOSALS

Jon Patrick, PhD
Health Information Technology Research Laboratory
School of Information Technologies
The University of Sydney

Abstract

The regulation of information systems has been resisted by the industry until recently when the American Medical informatics Association called for its introduction in 2010. This paper suggests topics for legislation that emerge from this study. Proposal for regulations about the development of systems includes the exhaustive testing of variable configurations, provision of mechanisms for forensic analysis of program code and a log of the staff who have made changes, provision of bug and error logs with public reporting of their remediation, and validation of primary key and foreign key integrity. Proposals about regulations for the management of data include preserving data in the form in which it is captured, data loading be defined by a formal process and validation strategy with reporting of error rates, validation of data imported from external organisations. The need for effective engagement of users in specification and human factors assessment is also proposed. A proposal about the removal of the "learned intermediary" defence for software manufacturers is also advocated.

1. Introduction

The debate on the need for regulation of the information systems that are used to record patient health care data has until very recently been subdued with very few voices urging for more regulation. The position held by the industry corporations has been that software can rarely cause problems for the patient as the responsibility for actions taken by doctors is determined by their role as a "learned intermediary". Therefore the doctors have to take all responsibility for patient misadventure. Concomitantly some vendors require purchasers to agree to vendor policies that prohibit disclosure of HIT system design flaws and weaknesses, errors, bugs, and other hazards [1]. Only in 2010 have professional societies like the American Medical Informatics Association (AMIA) come out publicly to say this position needs to be changed [2]. The Health Information Technology Policy Committee, an Amercian federal advisory committee, has recommended certification to the Office of the National Coordinator (ONC)[3]; and the ONC has proposed the establishment of certification programs "for purposes of testing and certifying health information technology."[4]

Other efforts to identify the dangers to patient care that are introduced by clinical information systems (aka EMR) have been stymied by a lack of a substantial number of tangible cases where such dangers can be proven to have unsatisfactory outcomes for patients. The proponents of regulations have argued that this has occurred due to restrictive practices of information sharing introduced in vendor contracts [1], whilst the industry manufacturers and its clinical supporters deny the existence of any substantial evidence and therefore any need for regulation. In the USA one comment was "FDA regulation should exempt most clinical software systems and focus on those systems posing highest clinical risk, with limited opportunities for competent human intervention."[5]

The debate has principally been on issues to do with patient safety and misadventure at the socio-technical layer of the work processes, such as problems with interfaces and workflow. This essay uses the design issues inside the software as a pointer for introducing regulations that improve the quality of software and the public's confidence in it and patient safety. Other potential regulations motivated by other sources are not addressed here. Each item of relevance is first described and then followed by a regulation proposal.

2. System Configuration and Database Maintenance

Large software systems are highly complex usually with 3 levels of organisation: storage, configurable parameters, and screen design. This complexity takes a large amount of time to learn and subsequently a software programmer cannot know the full consequences of any configuration decision due to interactions between the three levels. There is no way by which the configuration methods can be validated as they are performed directly on the client system. Hence an operation may work correctly for some time on one part of the system but not another. But when it does fail it can cause problems as broad as total failure across multiple hospitals to locally problems such as incorrect data being stored or retrieved for a single patient.

From the point of view of testing for such problems it is inadequate to perform a roll-out consisting of testing functions to perform in one scenario. This is especially so in an environment where complexity of the system is so great and an understanding of its behaviour is beyond the comprehension of a single person or even a small team. All possible pathways through the software, especially “corner cases”, should be regression tested. Any assumption that if it works for one then it will work for all should be disregarded, e.g. the same function with different variations for different hospitals.

It is clear that a substantial level of configurability in the software enables a corporation to claim that they can deliver any configuration a customer requires. While this might be literally true it is generally not practically true due to limitations in: the shortage of trained staff skilled enough to construct the required design, the time span in which the deliverables are needed, and the budget available to pay for the requirements.

For all its versatility, configurability brings its own difficulties, for example:

1. the user base asks for different variations of the system so that each new roll out has an increasingly more complex system to verify for the impact of each new change,
2. the changes in the system become more and more dependent on the knowledge of the programmers who have worked on the system, and their tasks increase in difficulty with added diversity and modifications,
3. The configurability functions do not have defined scopes that enable reliable prediction of how they will interact within other configurable functions.

Regulation Required: All possible pathways should be tested and configuration variables be such that their consequences can be correctly predicted. Regression testing with proper specification of corner cases should be included.

The configurability functions do not have defined scopes that enable reliable prediction of how they will interact within other configurable functions, e.g. one case is known where the standard values of gender male/female/unknown were programmed into the data stores to have the values 1/2/3 respectively. A programmer changed the ordering sequence in the interface so that all males and females were switched in display order and therefore incorrectly reported in the presentation layer viewed by clinicians.

Regulation Required: Content captured in the presentation layer should be retained unmodified in the representation and storage layers no matter what subsequent computation and re-presentation is performed. This will prevent changes in original values being lost whilst not inhibiting effective re-presentation processes such as graphing. Scale changes, units changes and origin shifts and other numeric transformations in the presentation layer of the patient episode may need to be prohibited under particular circumstances.

Database configuration maintenance can be ongoing on a daily basis. Some vendors have been known to accept reported bugs and fix them, but users have not been advised of changes, and so are unaware of the wide sweeping effects they might have on their system. It has been pointed out that voluntary revelation of changes to the system would be tantamount to admitting that something was

incorrect might open the software manufacturer to a liability and remove the protection they possess by the learned intermediary shelter written into contracts.

Regulations required:

1. Manufacturers to produce a log of all changes and their potential consequences. The reporting system can be designed to be no-fault, as in many critical incident reporting mechanisms.
2. The extent of testing performed on a bug fix be reported to the customer.
3. Define the scope of learned intermediary so that it can't be used as in defence of the system producing misleading information, or system based malfunctions.
4.

3. System Development Processes

The large effort required to develop complex software systems means that often the documentation is a low quality. At the same time the scale of the design is so great that consistency of meaning can't be maintained. Hence there is duplication and redundancy in the underlying storage structures that leads to inconsistent storage design decisions between developers. When these two designs come together then the data is not reliably stored and so not reliably retrieved.

Regulation Required: Forensic analysis of the implementation requires mechanisms in the software to record the behaviours of the programmers building and adapting the system. It should be possible to identify the changes made by individual programmers, and so determine responsibility for behaviours.

Processes for auditing the actions of designers and developers in maintaining a system need to be equally auditable as the processes of data entry and retrieval imposed on the clinical staff.

4. Management of Orderables

Each orderable or procedure in any clinical information system is made up of multiple elements (such as the clinical category, order entry format, order alert information, etc.). A process for setting up the management of orderables has to be initiated for any system installation. In general there are between 20 and 30 elements that make up each orderable and require definition in the setup process. Most elements are required to successfully place an order, but not all. Missing elements can occur either when the orderable is first built, or the element is wiped out later by a maintenance revision. In the normal course of events when an order is made by a doctor it will need to be completed by someone else, e.g. a nurse might have to administer a drug. The order would remain “uncompleted” until the nurse signs off the order as “completed”. In one cited case study completed orders were reverting to ordered but uncompleted, causing a patient to receive two doses of insulin and so suffer an overdose reaction. This demonstrates that each of the processes for setting up the orderables systems effect the specific orders but not everything in the system. However, one mistake in the order entry fields can affect every orderable that uses it.

Regulation Required: Processing functionality created by an installation process needs a formally defined validation process to ensure they operate correctly and do not lose data, and properly record and present to the user the entry of an order and the completion of its execution.

5. Management of External Knowledge

External knowledge is content that is collected elsewhere and embedded in the information system by some form of loading process. Typical examples are drug tables, drug interactions, values and ranges associated with orders, etc. The source of this knowledge can be from system internal repositories or external to the software manufacturer. It is well understood in the industry that this knowledge has errors in it from time to time either from the original source files or from bugs in the loading programs. For example some data is loaded from csv files which are generally known in the industry to carry faults that are difficult to automatically detect such as invalid values and missing values. In known cases values such as incorrect dosage have been identified, where the concentration rate e.g. "Aspirin 325mg" has the

dosage embedded in the name of the drug yet the field in the database where this value is stored is incorrect with a value of say, 1000mg.

Regulation Required: Validation of data in external knowledge sources needs to be mandated. This could take a number of methods: initial capture of the data needs to be done by double entry keying with the computation of key stroking similarity scores and thereby concomitant computation of the error estimates of the content. Storage of the information should be in database management systems that provided for automatic checking of domain ranges and primary key values and foreign key references. Released clinical information systems must be certified to have been loaded from the certified versions of the knowledge database. When advised by third parties of errors in this knowledge the supplier should be obliged to correct the data and distribute it to all customers of that data within 24 hours.

A second class of problem with external knowledge is the proper management of its storage. In one system for managing medicines there were NUMERIC and TEXT identifiers (IDs) across multiple tables that act as a link between the different tables to form the whole of the record about a medicine. As the IDs act in this way we would expect them to be declared as a Primary Key (PK) in the one primary table and then always referred to later in other tables as a Foreign Key (FK). This would ensure that the in-built functions of the Database Management System (DBMS) could be used to maintain the integrity of the record identifiers. In particular that deletions to any element across the tables should be checked for validity before it is executed. In this system the key for linking is not set up as a PK-FK relationship so one cannot rely on the integrity of the full description of a medication defined to be managed by the DBMS. Also failure to maintain a correspondence between PK and FK values of IDs leads to broken links in the chain of description of a medicine entry. Subsequently there will be incomplete information within the medicines and all orderable records when loaded into the customer application.

Regulation Required: Methods of data validation across linked tables be defined and actual validation accuracies be reported so that estimates of subsequent errors are known.

6. General Observations

In line with the above, and in keeping with the essence of this study there is the need for effective user involvement, workshops, and user engagement strategies. Given the paucity of these activities in this case study there would appear to be a need for the regulations to make them obligatory. Certainly, supplier that couldn't support them should not be engaged. Users have to include all levels from management to, especially, nurses, who seem to get left out and yet use the system most heavily. There probably have to be different strategies and mechanisms for different groups. Despite the cost it has to be funded; and it needs to be controlled by somebody other than the company and probably outsourced by someone from hospital management who has sufficient authority that they cannot be ignored. This really is a job for ethnographers, social psychologists, and Human Factors specialists, and not just experts in screen design.

References

1. Koppel R, Kreda D. Health care information technology vendors' "hold harmless" clause: implications for patients and clinicians. JAMA 2009; 301:1276e8.
2. Kenneth W Goodman, Eta S Berner, Mark A Dente, Bonnie Kaplan, Ross Koppel, Donald Rucker, Daniel Z Sands, Peter Winkelstein, for the AMIA Board of Directors. Challenges in ethics, safety, best practices, and oversight regarding HIT vendors, their customers, and patients: a report of an AMIA special task force. J Am Med Inform Assoc (2010). doi:10.1136/jamia.2010.008946
3. **Office of the National Coordinator for Health Information Technology.** Health IT Policy Committee: Recommendations to the National Coordinator for Health IT. Washington, DC: http://healthit.hhs.gov/portal/server.pt?open1/4512&objID1/41815&parentname1/4CommunityPage%20&% 20parentid1/437&mode1/42&in_hi_userid1/411673&cached1/4true (accessed 1 Oct 2010).
4. **Federal Register Online.** Washington, DC: GPO; 2010 [updated 2010 March 10;

DOCID:fr10mr10-17]. Volume 75, Number 46, Proposed Rules, pages 11327-11373. http://edocket.access.gpo.gov/2010/2010-4991.htm (accessed 1 Oct 2010).

5. Miller RA, Gardner RM. Recommendations for responsible monitoring and regulation of clinical software systems. J Am Med Inform Assoc 1997; 4:442-57; also summarized in: Miller RA, Gardner RM. Summary Recommendations for Responsible Monitoring and Regulation of Clinical Software Systems. Ann Intern Med 1997; 127:842-5.