Ockham's Razor of Design: An Heuristic for Guiding Design and Development of a Clinical Information Systems Generator

Jon Patrick

Health Information Technology Research Laboratory
School of Information Technologies
University of Sydney

(http://www.it.usyd.edu.au/~hitru)

jonpat@it.usyd.edu.au

Peter Budd

Health Information Technology Research Laboratory
School of Information Technologies
University of Sydney

(http://www.it.usyd.edu.au/~hitru)

pbud3427@mail.usyd.edu.au

ABSTRACT

This paper presents a rationale, created from first principles, for the design criteria for the architecture of clinical information systems. The criteria are developed according to the heuristic axiom of Ockham's Razor, presented here for the first time and operationalised in the form of three principles; Generalization, Minimalization and Coverage. The minimal set of characteristics of the application is defined as – the context, roles of physician, the nature of the data, the business processes, the change management program, and the criteria for success. This set of characteristics is developed to demonstrate how they define the details of a proposed architecture, which can be used to create not a clinical information system (CIS) but a generator of clinical information systems. The generator creates CISs that have the characteristics of complete user control of the design of all input and output presentations and workflow with real-time adaptability, ad hoc analytics with natural language processing of all free-text, all information automatically coded in a lingua franca terminology(s) of clinical choice, and native level interoperability between CISs An example of a CIS generated for the Trauma Service of an Australian hospital is described. The "generative" model is presented as a superset of the strategies for building best-of-breed and enterprise systems and thereby representing a new paradigm in Clinical Information System development. The parsimonious model is demonstrated by creating CISs requiring only 11 objects for designing the users interfaces along with their business logic.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Methodologies.

General Terms

Design

Keywords

Ockham's Razor, clinical information systems, generative

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IHI'10, November 11–12, 2010, Arlington, Virginia, USA. Copyright 2010 ACM 978-1-4503-0030-8/10/11...$10.00.

1. INTRODUCTION

"The No. 1 challenge to being a CIO, they say, is getting clinical and administrative staff to actually use, on a regular basis, the I.T. that CIOs and their staff have installed… The I.T. department has at times been a spectator as physicians, nurses and pharmacists fight cage matches over order sets, nomenclature and workflow issues. (Zack Martin, 2007) [1]

This work is motivated by the widespread experience of failures to produce optimal and interoperable information systems within the hospital and health community settings. The words of Zac Martin, a well-known commentator on these problems, are occasionally used herein to capture the feelings of a large majority of the health community as experienced over a long period of time. Concomitantly, this work is informed by a history of health informatics stretching back to the 1960's which has led to an impasse in Clinical Information Systems (CIS) being unable to share information and serve the health community in the light of increased expectations about their functionalities.

We define a CIS as an information system designed to collect data about the health of patients to support clinicians in their work of caring for patients. CIS has other synonyms such as EMR (USA) and EPR(UK) to name a few. Numerous systems have been designed, constructed and implemented in the health domain, with the many of them failing to be adopted in any meaningful way by clinicians at the point of care (Berg, 1999)[2]. The reasons for their failure are multi-faceted, not simply technical but also societal and organisational. This impasse revolves around the inability of traditional information systems design to be a suitable fit for the complex, working environment of the hospitals, unless there is strict enforcement from administrators and massive changes to clinician workflow to fit in with the requirements of the new systems. In the ensuing 10 years nothing seems to have change as in 2008 Greenhalgh et al [3] described the similar failings in the NHS of the United Kingdom with the added complications of coming to terms with differing aspirations between various management and clinical groups. They produced 8 critical factors at micro-, meso- and macro-levels that effected the outcome of implementations of Enterprise systems. In 2009 Greenhalgh [4] published a meta-narrative study of 24 systematic reviews and 94 primary studies to draw the conclusions that the scale of EPR implementations needed to be moderated against the needs of local clinical and patient communities. In 2009, Patrick [5] published a controversial essay about a case study of a system

roll out of Firstnet into Emergency Departments (EDs) in Australia that had led to only a limited utilization of its functionalities and a claim by clinicians that the development was significantly flawed.

McAlearney (2005)[6] conducted a systematic review of the CISs developed for hospitals and found their failings stemmed from similar problems:

- A focus on tasks the computer does well, such as dealing with quantified information, well defined vocabularies, great speed over repetitive tasks, etc. They were built with a view to making the system design easier, which in turn forced clinicians to change their work practices to fit the system. This resulted in the clinicians task becoming more difficult.
- The CISs were rigid, built off-site and incredibly expensive both in time and money to change. Many desired functions of CISs were passed over because of time, funding and retraining concerns.
- The systems didn't take into account properly the reuse of data contained in the CIS across the hospital.

The work of Beyer, *et al* (2004)[7] also highlights interoperability and inter-institutional support as being major drivers of system failure.

CISs of the current era are built on the need to support access to the Electronic Medical Record (EMR). The work of the physician has changed in many ways since the pioneering of these systems. Now it is time to review the basis of their design through the lens of changed work contexts and processes of physicians, and technological advances in software engineering. One of the greatest changes is an increased need "to know" by all health professionals engaged in the care of patients.

The single unique limitation of CISs in hospitals that fails the needs of the staff is that the many independent CISs in a hospital cannot exchange data. Clinicians in the workplace commonly complain that this results in the following needs being inadequately met:

- The need for physicians to have all relevant data available for decision-making.
- The need to aggregate data readily for answering research questions.
- The need for ready access to individual case data and aggregated data for audit investigations.
- The need to automatically process the written language of clinical reports and notes to enhance decision making.
- The need for aggregated data for the administration of the hospital in planning and logistical management.

For the physician on the hospital floor, the redesign of CISs needs to deal primarily with their lack of ready access to data. One class of solution to this theme is to take an enterprise–wide perspective and bring all systems into some form of singularity. This is well presented in the Mayo Clinic's Enterprise Data Trust which is extensive in the existing clinical, research and business information systems it accesses and the unification of data dictionaries it accords to enable semantic interoperability with the Clinic's operations for data analysis and decision making [8]. In some sense the opposite type of system is best-of-breed which can be represented in one of its form by the work of an i2b2 team [9] who built a specialized system from exporting excerpts of the patients' record into a tailored research system and subsequently cleaned the records for research analytics. Their work gives detailed attention to the underlying data models of their processing solution. We will present arguments later that this should be an unnecessary strategy for creating CISs.

We argue that both these strategies have their drawbacks and that a new paradigm is required to serve the many masters of medical record processing, as, for the designer a larger set of considerations than revealed in these case studies come into play. Namely they must design a system that gives users access to their data in a manner that is optimal for them, and that the system has the longest possible life maintainable at the lowest possible cost. All these criteria of both the users and the designers must be used in the justification for a new class of information system. This paper returns to the first principles of design to present a justification for specific design criteria as well as exposing its design methodology for critique.

The aim in this paper is to move onto a new generation of information technology that emerges out of the past, but has learnt the lessons it provides. This we call Health Information Technologies and it is informed in part by contemporary understanding of the methods of computer science and software engineering rather than that of the period of the 80s which was informed by the methods of prescriptive systems design, programming and data processing.

The Generative Clinical Information Management System (GCIMS) is a new type of architecture for clinical information systems. This system is built with a different philosophical position to current CISs with its own model of the purpose of such a system and the different roles of functions of the people who use it. It aims to define the next generation of architectures. Building effective electronic support for the health system requires two essential ingredients to maximise the life and utility of the solution and minimise its cost. Firstly, the approach needs to be principled, that is, it has to work from a set of principles and seek to reaffirm those principles at each step of design and check against them at each decision point in the process of model development. This approach provides guidance on the most important criteria for making decisions, resolving competing solutions at decision points, and ensuring a concrete and consistent understanding of why certain decisions are made. Second, develop an accurate and useful model of the clinical information system that is specifically defined in a manner that makes the model testable against the design principles in a tangible and verifiable manner.

2. Approach and Foundation Principles - Beliefs and Values

The intellectual background for this work is not drawn from the general theory of design itself but rather from the minimalisation principles in Shannon's Information Theory (Shannon, 1948)[10] which were brought to a culmination of scientific method in Wallace's Minimal Message Length Encoding (Wallace, 2005)[11], now considered to be the closest method we have to a computational form of Ockham's Razor. The application of these general ideas by the authors, to clinical systems is motivated by the incessant claims of all members of the health community as to the unsatisfactory state of the current CIS technology.

So as to construct a new architecture unshackled by both the immediacy of the views of the purposes of past systems and by their dominance of current practice we must expose the fundamental principles of our design methodology. The first step in this process is to define the fundamental heuristics or axioms of the methodology and then develop a set of reasoned principles by

which the design of a clinical information system can progress and against which that design might be assessed.

The design criteria should be founded on characteristics both intrinsic to the nature of design and on operational properties and the needs of the ultimate owners of the systems that are produced. We offer the following principles as the most appropriate for proceeding with this task. In following the traditions of scientific endeavour expounded in Ockham's Razor which states:

Numquam ponenda est pluralitas sine necessitate
[Plurality must never be posited without necessity]
(Ockham's Razor, Wikipedia)[12]

We propose an equivalent core principle which we name Ockham's Razor of Design (ORD):

A design should use the minimal number of entities with their maximal generalisations.

This principle, whilst posited as a unitary idea is decomposable into three derivative principles for the purpose of applying it to the process of design, that is operationalising it as a method for guiding and evaluating a design; we have labeled them as the Principles of **Generalization**, **Minimalization** and **Coverage**.

Generalization is the principle of constructing design concepts described at the most general level possible. This criterion ensures that the system described covers the widest possible range of exemplars thus ensuring the greatest life duration for the system. One of the main reasons CISs fail or fall into decay and moribundity over time is because they have failed to be designed for exemplars that had not been considered. The counter argument is that accepting too much generalization initially delays the development of a system, increases costs, and complexity as the system development attempts to support more and more diverse needs. This is a misconception of generalization, which is the task of identifying the things that are truly common and the ways in which they are not common. It has to be accepted that some system entities are generic on one attribute and different on others. In such circumstances the question to be asked is – "Is the common attribute a defining part of the system, which is common to all other classes included by the system, or is it common on a minor attribute that carries very little weight in considering the design, in which case it is excluded from the design generalization."

Minimalization is the principle that limits design to identifying the smallest set of features that give coverage to the greatest range of functionality desired in the system. To fulfill this criterion the relationships between various functionalities need to be established and consideration given to their dependencies whether they be mutual or singular.

Coverage is the principle that the set of entities required to define the design cover the needs of the design application. This is necessary to ensure that the design is complete.

The combination of Generalization and Minimalization yields the most general and therefore smallest description of a system with the greatest functional coverage. This is a desirable property of any design because it focuses on the most crucial elements that serve the greatest purpose. Achieving success in designing a system to fulfill these criteria will provide an elegance of solution, a solution description that is comprehensible to both the engineers who have to build it and the users who have to use the system, and subsequently the next generation of engineers who have to maintain and extend the system in response to revised workplace processes. This last point is most crucial to the successful extension of the life of a working system as the more accessible the fundamental properties of the design of the system the longer the life it will achieve.

In the spirit of optimizing the balance between these three criteria of design, the Ockham's Razor of Design can be more explicitly stated as:

"entities should not be multiplied beyond the minimum set of generalisations to cover the scope of the task"

2.1 Assembling Generic Requirements

The generic requirements to be detailed have to be taken from four topics that fix the scope of the performance the system has to fulfil:

Once these criteria are elucidated the task of this work has to be to define the system by following these steps:

1. Define the architecture of the CIS and justify it against the requirements.
2. Define the work processes embodied by the architecture.
3. Decompose the processing systems into units of specific functioning roles to the extent that their functionality is readily understood by an experienced software engineer.
4. Define a construction approach that prima facie would satisfy a test of implementability.

The remainder of this paper endeavours to follow this design plan to ultimately justify a new approach to the architecture and functioning of CISs.

2.2 Putting the Heuristics to Work – Identifying the Generic Entities

The axiomatic heuristics are the guiding principles used in developing a set of design criteria that can be used to assess the nature of individual design decisions for their usefulness in both quantitative and qualitative ways. The generic entities to be detailed have to be taken from topics that fix the scope of the performance the system has to fulfil. These separate into three types, Beliefs & Values, Characteristics of Design, and Matters of Implementation. Beliefs & Values have already been discussed in the Approach and Foundation Principles of this paper.

The Characteristics of Design consist of the following generic entities:

- Social Context – Complex System – the social system in which the work is conducted
- Roles of Users – Physicians and their proxies at work.
- The Nature of the Data - The form of data which has to be handled in the CIS, which is predominantly Text in Time and Location.

The Matters of Implementation consist of the entities:

- The Business Processes – Captured in Forms
- Change Management Program - Pathway to Adoption (skill, will, and time to do it)
- Criteria for Success – Acceptance: Satisfy very busy and conscientious people

3. The Characteristics of Design

3.1 Social Context - Complex Systems

Our philosophical standpoint is to view a hospital community as a complex system in the sense of complexity theory and hence assign to it the following characteristics and constraints:

1. No central control;
2. Self-organising communities persuing local optimisation with non-deterministic behaviour of autonomous members;
3. Open System – people, organisations and ideas cycle in and out;
4. Limited view – no-one can see all of it at once;
5. Multiple levels – with fuzzy boundaries;
6. Emergent Properties – features emerge from the wholism of the system.

CISs are built as engineering artefacts - they consist of inputs of what is considered important, and, outputs which are considered to be wanted in advance of being known to be useful. An evaluation of the complex systems model of a hospital community provides some insight into the design frames for a Clinical Information System such as:

1. No central control: communities/people have to be able to create their own CIS;
2. Self-organising communities: have to be able to change their CIS at will as their needs evolve;
3. Open System: old data and models need to be retrievable (archived but still accessible) as much as new models introducible;
4. Limited view: the complete CIS for the whole system has to be able to be assembled from all the CISs of the individual communities and yet its usability by the whole community has to appear as an emergent property;
5. Multiple Levels: the definition of different degrees of granularity for system description. Granularity as a property of system entities might change with different contexts created by self-organising communities;
6. Emergent Properties: constitute the coordinated outputs of the system contingent on the wholism of the system design;

The two basic tensions in this model are:

- satisfying the needs of the individuals and local communities versus those of the whole system e.g.

"The No. 1 challenge to being a CIO, they say, is getting clinical and administrative staff to actually use, on a regular basis, the I.T. that CIOs and their staff have installed." ... "Twenty-eight percent of respondents said staff buy-in was their biggest challenge (Zack Martin, 2007).

- satisfying the needs to have completely dynamically changeable CISs versus an engineering demand to predetermine and design all functionalities.

"The pharmacy wants to call Tylenol one thing and doctors want to call it something else..."And when it comes to workflow, every group of clinicians want to do things differently. At some point CIOs are wondering how they can merge what everyone wants into one system." (Zack Martin, 2007)

3.2 Roles of Users – Physicians At Work

There are many professionals involved in hospitals whose work activities operate in a complementary manner with the physician - most crucially nurses - but also scientists, and allied health professionals (physio, OT etc). For the purposes of this paper they are all considered as part of the physician team or are surrogates or proxies and so encompassed under the category of physician.

We view the work of the physician on an axis of four roles that each need to be served in different ways by the CIS. They are:

1. Physician as Point-of-Care Clinician: where information retrieval of the patient record is the most important function required from the CIS;
2. Physician as Researcher: where the function of the CIS is to support the analysis and reuse of data over time in knowledge based systems and to build the substantial case for Evidence Based medical practice;
3. Physician as Administrator/Manager: with the same functions as the Physician-researcher but required for planning, logistics, and general operations management;
4. Physician as Auditor: where a physician compares the results of their care with an ideal care plan and analyses the differences. This is a role that is frequently performed intertwined with the role of Clinician, but also it is more formally separated in the work of patient safety and clinical excellence professionals. Evidence Based Medicine is driven heavily by this role.

3.3 Community Wholism

We view a complex system such as a grouping of health communities either by natural aggregation or by governmental administration as made up of many communities some within other communities.

Members at the "top" of the communities may not actually be part of the communities below them, for example, the CEO of the Health Service (HS) is a member of the HS community but not necessarily of any community within the HS. However such a person in the administrator role has a key interest in being serviced by all the information systems within the communities.

3.4 Knowledge Wholism

Knowledge is the general body of knowledge physicians receive and develop through their educational and practice processes. It is used in a limited way in most CISs, the common manifestation being through Decision Support Software. However subtly it also is implicitly used in the Research and Auditor Roles when retrievals and reports from the CIS are designed to evaluate such knowledge.

3.5 Fundamental Data Content Classes & Data Types – Text in Time & Location

There are three fundamental classes of data content that need separate consideration: Patient Content (History & Socio-demographics), Medical Measurements, and Medical Knowledge. These content classes come generally in a mixture of 3 data types: numerical quantities, categorical attributes, and natural language. While images are a formal method of data capture they are most often converted into one or more of the three basic types. These classes can be further divided in many ways into a more detailed classification if needed, but the basic divisions define the functional purposes that exist at the point of maximal generalisation.

1. Patient Content: socio-demographics including the personal background of the patient and their clinical history;
2. Medical Measurements: including information from monitors, pathology laboratory, imagery, direct observation, diagnosis, etc;

3. Medical Knowledge: includes the physician's personal knowledge and skills as well as content present in medical literature and computational surrogates such as thesauri, taxonomies, ontologies and classifications (TTOCs), and decision support systems.

These data content classes carry intrinsically different demands on the processing of the CIS and thereby not only require different storage requirements but also different secondary software to harness their value, e.g. viewers to see X-rays, SNOMED encoders for natural language, natural language parsers, statistical functions for record aggregation, database management systems for record storage, etc.

It is often asserted that organizational structure constitutes a class of data content that needs to be captured in a CIS. This paper does not find that it is necessarily an explicit part of system description, but rather it arises as an emergent property from defining the functional needs of the various communities within the whole of their inter-relationships.

The ultimate constraint on these classes is that they have to be situated in Time and Space, which are the intrinsic axes of clinical information. Hence any CIS needs to define time and location as data native to any clinical record and allow for analysis over these two axes.

3.6 Outputs Maximise Genericity: Not Inputs nor Storage

The fundamental service of the Individual Systems and the Whole System have to be defined on the functions of the outputs NOT the inputs NOR the storage of the system. The definition of this service is created by the various roles of the community members: clinician, researcher, administrator, and auditor. The single generic function of all these user roles is data analytics and not data retrieval. It is a common belief that the point-of-care clinician only needs data retrieval and so most CISs are designed on that premise. However we argue that this is not the case and the clinician might, for example, use a decision support system or refer to a text book or clinical guideline (whether on-line or not). This paper further argues that such an action is the action of an auditor and for a clinician to take such an action they are shifting from the role of point-of-care clinician to auditor. The difference may be immaterial to the clinician in the performance of their patient care but it is vitally important in understanding the appropriate manner in which to define the most generic CIS.

Hence the fundamental objective of the CIS has to be the delivery of uncompromised, totally comprehensive Data Analytics. There are a number of consequences from this position:

- Firstly, any questions for which there is an answer within the system should be stateable in the data analytics language.
- Second, any stateable question should be computable by the underlying processing engine.
- Third, data retrieval is only a by-product of data analytics and it is NOT a point of focus for the design of system architecture.

4. Matters of Implementation

4.1 Premises for an Architectural Design

The premises set out in this argument need to direct the architecture of the clinical information system to fundamental tenets of design for such a system. We propose that these are:

- A CIS has to be indistinguishably both many systems and one WHOLE system;

"Getting staff buy-in is the biggest challenge facing the I.T. initiatives of many CIOs. But in many instances, that buy-in is contingent on information systems working smoothly and seamlessly with clinical workflows. And that task requires systems integration, or interfacing, an elusive goal for many health care facilities. And the biggest challenge to systems integration is trying to keep data flowing smoothly across applications while at the same time piling on more technology."
(Martin, 2007)

- A CIS has to have the delivery of data analytics as its fundamental raison d'être;

"Health care CIOs and other I.T. leaders plan to sink more money into applications that increase clinician data access and help reduce medical errors... Of those respondents who expect their I.T. budgets to increase in fiscal 2007, 51% said the primary factor for those budget increases is to purchase technology that increases clinician access to information."
(Martin, 2007)

consequently,

- each system within the whole system has to be totally "owned" by the community for which it serves. Currently we know these systems as best-of-breed systems;
- the development of the "whole" system is apparently intrinsically anarchic and at the mercy of the autonomous member communities but must nevertheless have emergent properties that satisfy the needs of the members of the "whole" community, and those needs are fundamentally data analytics. Current solutions are known as enterprise systems or ERPs.

4.2 Route to Acceptance

Any new approach to building CISs will meet resistance from users who wish to minimise the time cost of clinical redesign for themselves. The architectural design must strive to achieve the least disruption to current work practices and at the same time the greatest advance in workflow, functionality and productivity so that users have minimal effort for maximal gain.

4.3 Challenges

The challenges from such a model are:

- Ensuring cohesion without forcing cohesion between collaborating communities;
- Ensuring appropriate emergent properties are achieved without specifically designing for them;
- Defining a data analytics language of the necessary power.
- Making the adoption of this new architecture acceptable to users.

From the viewpoint of current system designers another set of questions can be posed which are made redundant by the intrinsic characteristics of the proposed architecture, some of which are:

- Delivery of the functionality required to meet all major perceived needs.
- Ensuring all the work-force problems with system usage are addressed.
- Managing some mechanism whereby those whose work is slowed and made more difficult are rewarded for involvement and participation.

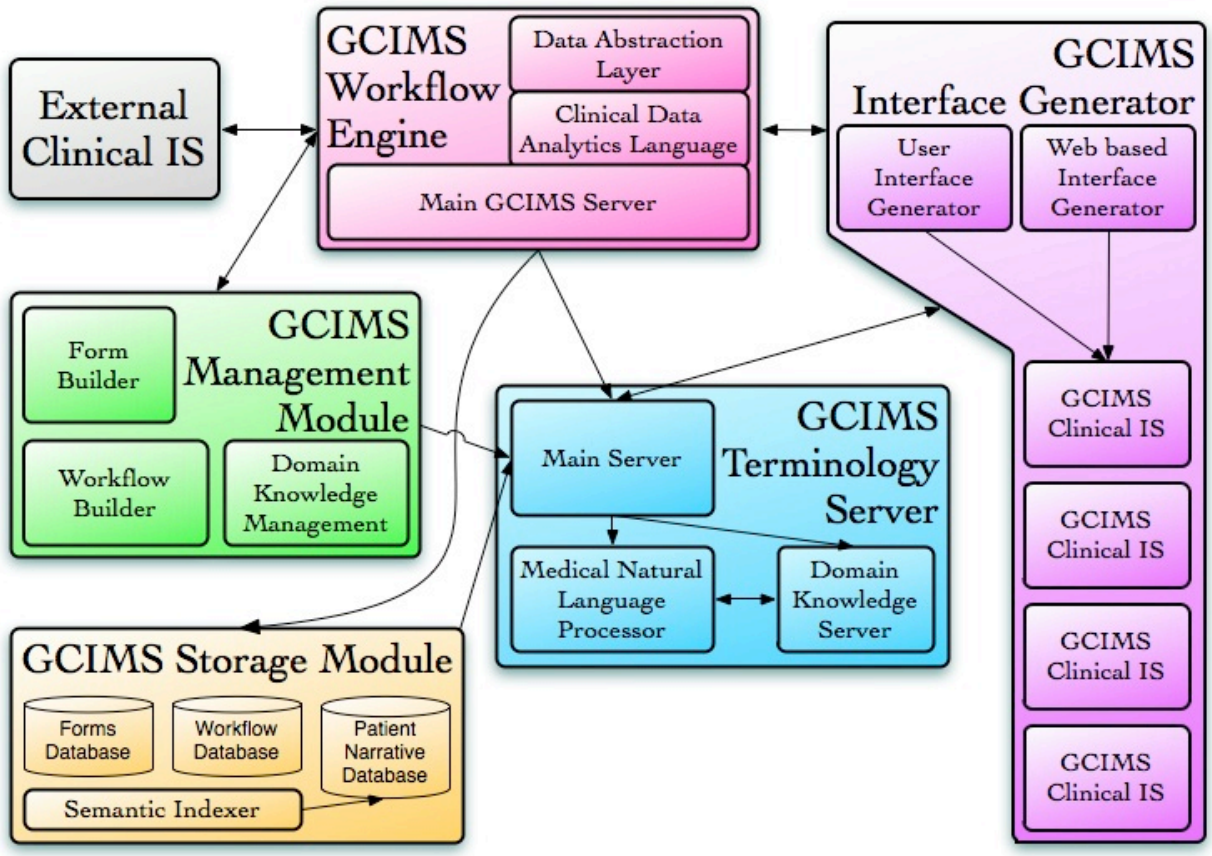

Figure 1. GCIMS Architecture

4.4 Implementation of Architectural Strategy

The integration of all these criteria produces a different perspective on the implementation strategy for clinical information systems, namely, each community has to be responsible for creating and evolving their own CIS. However, to satisfy systems wide access and reuse of ALL data there has to be a unifying coherence that they all participate in.

This view cuts across previous perceived designs such as "there will need to be a series of serious design decisions about just what is common and what is the scope of the CIS which is going to support individual requirements etc", in that there is no intrinsic necessity to fulfill this criteria, our proposed architecture can satisfy idiosyncratic requirements without performance or implementation costs beyond the marginal. It might however be useful to have administrative management of the system description to satisfy local issues of governance.

4.5 System Design of a Generative Clinical Information Management System (GCIMS)

A design to cater for the aforementioned architectural needs is a system that generates tailored information systems for individual community needs but preserves integration and access to data over all communities. For the sake of transition from a currently implemented system, the new system should be able to mimic the functions of existing systems, especially if it is a paper system.

Hence a Generative Clinical Information Management System (GCIMS) is one where all information systems are generated from the one system and all data is known and understood across all generated systems. Under such architecture an emergent property is interoperability, that is, the system doesn't have to be designed for it. It is intrinsically achieved in each local system, which has the capacity to know about all information in all other systems when it is passed to them.

This idea brings the notions of best-of-breed and enterprise systems into a different positioning as our technology of "generative systems" embraces both of them and supersedes their architectural designs. The two together constitute a subset of architectural form contained wholly with generative systems and together are less than its whole.

5. Engineering Solutions with Forms

1. A community CIS is defined by the definitions of the data that is captured within their work processes. They need to be able to control the definition of the content they collect, and its manner of presentation through suitable software that provides for FORMS and full real-time revision control of such forms.

2. Forms create the context in which data is collected. Hence data about a patient is always captured in a versioned form. The data is stored with a record of the form and its version number. The data is retrieved and presented to the user with the same version of the form by which it was captured thus retaining its context. Selection of forms for usage will have to ensure that the most current version of a form is used.
3. A community CIS thereby becomes an environment where a community can define its own forms and their content and retrieve its own forms as well as those of all other communities within the *whole* system.
4. A description of the workflow process of the community defines the route by which information must be collected and travel through the community.
5. All information defined in a community CIS must be anchored to a common data dictionary. This does not constrain dialect labeling for individual communities, however it does require the dialectal language be tied to a standard description language of all communities.
6. The data analytics functionality is:
   - common to all communities within the system;
   - its general design over the definitions of all defined data in all systems makes it universal for all user roles.

5.1 Emergent Properties

The features of the GCIMS that emerge from this approach are:

1. Perpetual real-time evolution - a GCIMS generated CIS can be changed at anytime in anyway by use of a version control system for all descriptions of all data.
2. Every community can see all the data captured by all other communities in the contexts (forms) in which it was captured. As the central server can send data collected at any one community CIS to any other community CIS along with a description of how to present it, all data records are available at all locations across all GCIMS generated systems in real-time.
3. Coherence of data descriptions between different medical dialects is maintained by a centrally managed data dictionary.
4. Patient Logical Tracking - a GCIMS generated CIS requires a workflow analysis of data processing within a community so that information is pushed from where it is collected to where it is needed. Such a system gives a logical location of each patient or "patient logical tracking".
5. Whole of System wide load analysis - The patient tracking process means that at all times the logical location of a patient is known and all such locations can by drawn on a network map indicating loadings throughout the network.

5.2 Criteria for Success

Adoptability is the criteria that a system must be able to be put in place and immediately usable by the community of workers for whom it is designed. An adoptable system will be intrinsically recognisable as of value to its users and be readily integrated into their existing workflow practices. At the same time it will self evidently demonstrate an improvement in the quality of the work it supports and thus encourage its adoption by the workers.

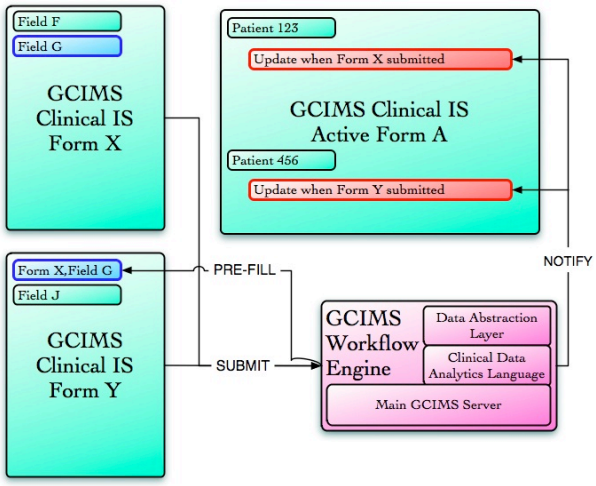

Figure 2. GCIMS Workflow Engine

Implementability is the criteria by which the practical design of the system has to be judged. A system is implementable if it requires a set of resources available for the life of the completed system at the project development stage. Also implementability is partly defined by the complexity of the connectivity pathways between system components. As the complexity increases so does the difficulty of ensuring the implemented system conforms to the specified system. Hence a controlling characteristic in the design process is to limit the complexity to comprehendible bounds.

Adaptability is the criteria that a system has to be readily adaptable. Setting a quantitative form for this criterion appears desirable but is elusive. An adaptable system allows the owners of the system to change its functionality at will. This is a tall order in a totally general world but within a defined domain it should be achievable. Adaptability is even more achievable if the language of the domain is both restrained at least to some moderate degree and unambiguously defined. Both of these conditions can be met for the most part by health CIS applications.

The key aspect of achieving an adaptable CIS is to provide a mechanism for describing functions that are non-predetermined at least in terms of the range of user functions that can be serviced. This entails creating a mechanism for users to describe their own CIS, or at least to allow them to believe they only have to create that description to get their own system. The adaptable system will then manage the CIS implementation in its own independent manner without the user needing to know.

6. GCIMS Model

The resulting GCIMS model that emerges out of our work is comprised of five sub-modules that when combined together are able to produce all the required functionality of a CIS while fulfilling the design criteria of the Ockham's Razor of Design. A graphical representation of this design is shown in Figure 1. An overview of each of the five sub-modules, paying special attention to how they encompass optimal design is provided.

GCIMS Workflow Engine

The workflow engine is the heart of the system. It coordinates all the functions of the system, acting as a bridge between the management module, the storage module, the interface generator

and external, non-GCIMS CISs. The workflow engine is implemented as a TCP-linked server connected to a cluster of processing machines in the background. The server provides functionality in addition to bridging requests between the other modules:

- The Data Abstraction Layer is responsible for hiding the particulars of external information systems from the GCIMS architecture. It provides semantic mappings into known databases and readers/writers for HL7 messages to allow semantic retrievals, aggregations and analytics of patient information irrespective of the source of the information.
- The Clinical Data Analytics Language (CliniDAL) handles analytics requests from anywhere within the system or without. The CliniDAL language is a restricted natural language that allows a wide range of ad hoc analytics based on any TTOC known to the terminology server. Its functionality is specifically on semantics, not just syntactic or morphologic as it has access to TTOC codes and their structure from the Terminology Server.
- The main GCIMS server is responsible for routing requests within the system and responding to workflow events within the system as the system changes. These responses take two forms in the system, notifications and pre-filling as shown in Figure 2.

GCIMS Management Module

The management module is responsible for the creation, building and management of the information and representations required to generate the CISs as well as managing the TTOCs stored in the Terminology Server. The three sub modules are:

- **Form Builder** is used for building the user interfaces, defining their data inputs, outputs, screen positions, the overall appearance of the interface and links the data elements to a TTOC within the Terminology Server. The user interfaces are intended to be created by non-technical users with no programming experience. It utilises a drag and drop interface and generic button types that can be configured easily to perform a wide variety of functions. It saves the created form as an FDL (Form Description Language) which is readable by the Interface Generator.
- **Workflow Builder** is responsible for defining how the data flows within the system, where fields are pre-filled from, what information triggers a notification as well as alerts and the logic behind the decision support component of the workflow engine.
- **Domain Knowledge Management** looks after the TTOCs that are currently in the Terminology Server, adding new concepts, setting preferred terms, updating mapping tables between TTOCs and, all the functionality required to update and manage the TTOCs used by the system.

GCIMS Storage Module

The storage module is responsible for storing, organising and indexing the data used by the GCIMS system. It has 3 main databases, each containing versioning control so that information can be recalled from the system in the exact appearance and context that the original data was entered. These three databases are the **forms database**, **workflow database**, and the **patient narratives database**. Each of these holds the information which is controlled by the management module and used by the interface generator and workflow engine.

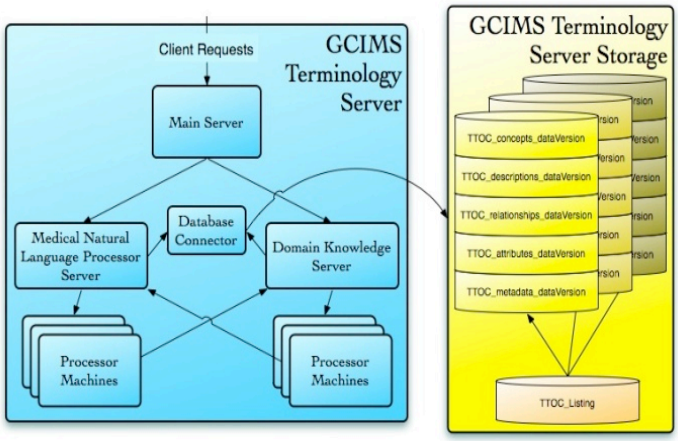

Figure 3. GCIMS Terminology Server

In addition to the three databases there is a sub module called the **semantic indexer,** which is responsible for providing an index of the data in the patient narrative database based on a generic and extensible range of indices. These indices can be based on structured features such as last name, or admitting physician as well as semantic features such as Clinical Terminology codes for diagnosis, or family history of cancer.

GCIMS Terminology Server

If the Workflow Engine is the heart of the system, then the Terminology Server is the mind. It gives meaning to otherwise generic labels on the data fields and an explicit structure that can be understood by the system for the purposes of data analytics and retrieval.

The Terminology Server is designed as a main routing server connecting to at least two sub-servers, each of which is a processing cluster, as shown in Figure 3. The structure and rationale behind the design of the Terminology Server is outlined in Patrick & Budd (2010)[10]. The two sub-servers are:

- **Medical Natural Language Processor Server** handles requests within the system for identifying medical concepts within free-text and all the associated functionality required for labelling the form fields and extracting medical information contained in the text fields of the forms.
- **Domain Knowledge Server** handles access to different TTOCs in use in the system. This access comes in the form of TTOC management and questions pertaining to the structures of the TTOCs. These questions encompass such requests as getting the children of a concept, the complete sub-graph of a concept, synonyms for a concept, known abbreviations for a concept, the relationships between concepts, etc. It also handles mappings between the different TTOCs so that analytic queries can be performed across all generated CISs without specific knowledge of any of them.

GCIMS Interface Generator

The interface generator is responsible for creating the graphical user interfaces which the clinicians interact with. It takes the FDL from the Storage Module and parses it to recreate the interface that was built using the Form Builder. As the FDL is independent of programming languages, operating system

platforms and human-computer interface devices, a number of interface generators can be built for specific devices or purposes while still building, working in and contributing to the same CIS without specific redesign of the core system.

As a case in point the original prototype of the GCIMS model had an Interface Generator written in python using wxPython, and the Trauma Information Management System (TIMS)(detailed below) required web browser access to the system within the ED. A web-based interface generator was subsequently written using AJAX that generated the interfaces in keeping with web 2.0 philosophy. The two interface generators are capable of faithfully displaying the interfaces and saving/loading patient data without concern for the location of the data. Extensions to mobile devices are trivially facilitated. Parsimony of design has been achieved with for only 11 objects needed for all interface behaviours required to date.

7. Operationalising the GCIMS Model – A Trauma Clinical Information System

Development of the GCIMS has progressed to the point where it is possible to create an operational CIS. The first system has been designed for the trauma service at the Royal Prince Alfred Hospital and is in the beta-test stage. An ED system (NEDIMS) is about 90%complete and a multi-disciplinary team system (MDIS) 50% complete. The Trauma team have designed a set of screens to support four aspects of their work:

- **Triage** to collect crucial information about a patient who has suffered some form of trauma and is likely to need prompt and urgent care;
- **Decision Support and KPI screens** used during the first immediate care of the patient;
- **Secondary Survey** to record the details of the patients care in first few hours in hospital including all the interventions;
- **Tertiary Survey** is the follow up study of the patient after the first few days in hospital to check if there are any unattended injuries or concerns by the patient.

The component system has shown the clinical staff how to build a system and manage its evolution by changing their specifications as the system developed. The component design has facilitated full dynamic control of the emerging information system without any particular requirements collection and analysis stages. Occasionally functionality requested by the users has not been available, such as clocks counting down the time remaining to treat a patient. These types of requests require a revision of the GCIMS itself and some code development to supply the functionality but this has not caused serious impediments to continuing testing and constructing other elements of the system.

Although GCIMS is designed for use by clinical staff their engagement in learning to use the designer tool has been variable. For the TIMS system the clinicians have been prepared to just say what they want and leave IT staff to do the building. The NEDMIS system has fully engaged the clinical staff with added help from a health informatician. Figures 4a and 4b show the changes in TIMS from initial paper mimic to a system optimized for parsimonious screen and system design, which is optimal for clinical workflow, and professionally and emotionally owned by the clinical team. At the same time all administrative reporting is incorporated in the screen tab Registry Report embracing best-of-breed and enterprise requirements in the one generative technology.

8. Conclusions

A model principle for the design and development of software is proposed as Ockham's Razor of Design. The application of the principle to the case of designing clinical information systems has been demonstrated and a practical demonstration shown by the creation of a Generative Clinical Information Management System. The GCIMS system in turn has been used to create a Trauma Information Systems as specified by the staff of the Royal Prince Alfred Hospital. It is undergoing beta site testing at the moment with other systems also in development.

9. References

[1] Martin, Z. (2007): The No. 1 CIO Challenge, Health Data Management, April 09, 2007, www.healthdatamanagement/issues/20070301/14738-1.html. Accessed 31-8-2009.

[2] Berg, M (1999): Patient Care Information Systems and Health Care Work. A Sociotechnical Approach, I*nternational Journal of Medical Informatics* **55**: 87-101

[3] Trisha Greenhalgh, T. Stramer, K, Bratan, T , Byrne, E, Mohammad, Y, Russell, J. Introduction of shared electronic records: multi-site case study using diffusion of innovation theory. *BMJ 2008; 337:a1786*

[4] Greenhalgh, T, Potts, H.W.W, Wong, G, Bark, P. and Swinglehurst, D. (2009) Tensions and Paradoxes in Electronic Patient Record Research: A Systematic Literature Review Using the Meta-narrative Method. **Milbank Quarterly**, 87 (4). pp. 729-788. ISSN 0887378X.

[5] Patrick, J. A Critical Essay on the Deployment of an ED Clinical Information System – Systemic Failure or Bad Luck? Accessed 13-09-2010. http://www.it.usyd.edu.au/~hitru/index.php?option=com_content&task=view&id=91&Itemid=146, Item 6.

[6] McAlearney A S, Schweikhart S B, and. Medow, M.A. (2005): Organizational and Physician Perspectives about Facilitating Handheld Computer Use in Clinical Practice: Results of a Cross-Site Qualitative Study. *Journal of the American Medical Informatics Association* **12**(5):568-575.

[7] Beyer, M., Kuhn, K A, Meiler, C, Jablonski, S, Lenz, R (2004): Towards a flexible, process-oriented IT architecture for an integrated healthcare network. *Proceedings of the 2004 ACM symposium on Applied computing*: 264-271.

[8] Chute, C.G, Beck,S.A, Fisk, T.B, Mohr, D.N. The Enterprise Data Trust at Mayo Clinic: a semantically integrated warehouse of biomedical data. JAMIA 2010;17:131-135.

[9] Murphy, S.N, Weber, G, Mendis, M, Gainer, V, Chueh, H.C, Churchill, and S, Kohane, I. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). JAMIA 2010;17:124-130

[10] Shannon, C. (1948): A Mathematical Theory of Communication. *Bell Sys Tech. Jnl 27* pp 379–423, 623–56.

[11] Wallace, C.S. (2005): Statistical and Inductive Inference by Minimum Message Length, *Information Science and Statistics*, XVI, 432 pp, Springer

[12] Ockham's Razor. Wikipedia. accessed 23-08-2009, http://en.wikipedia.org/wiki/Occam's_razor.

[13] Patrick, J and Budd, P (2009): A Taxonomy of Terminology Server Desiderata*. Accessed 13-09-2010. http://www.it.usyd.edu.au/~hitru/index.php?option=com_content&task=view&id=91&Itemid=146, Item 7.*

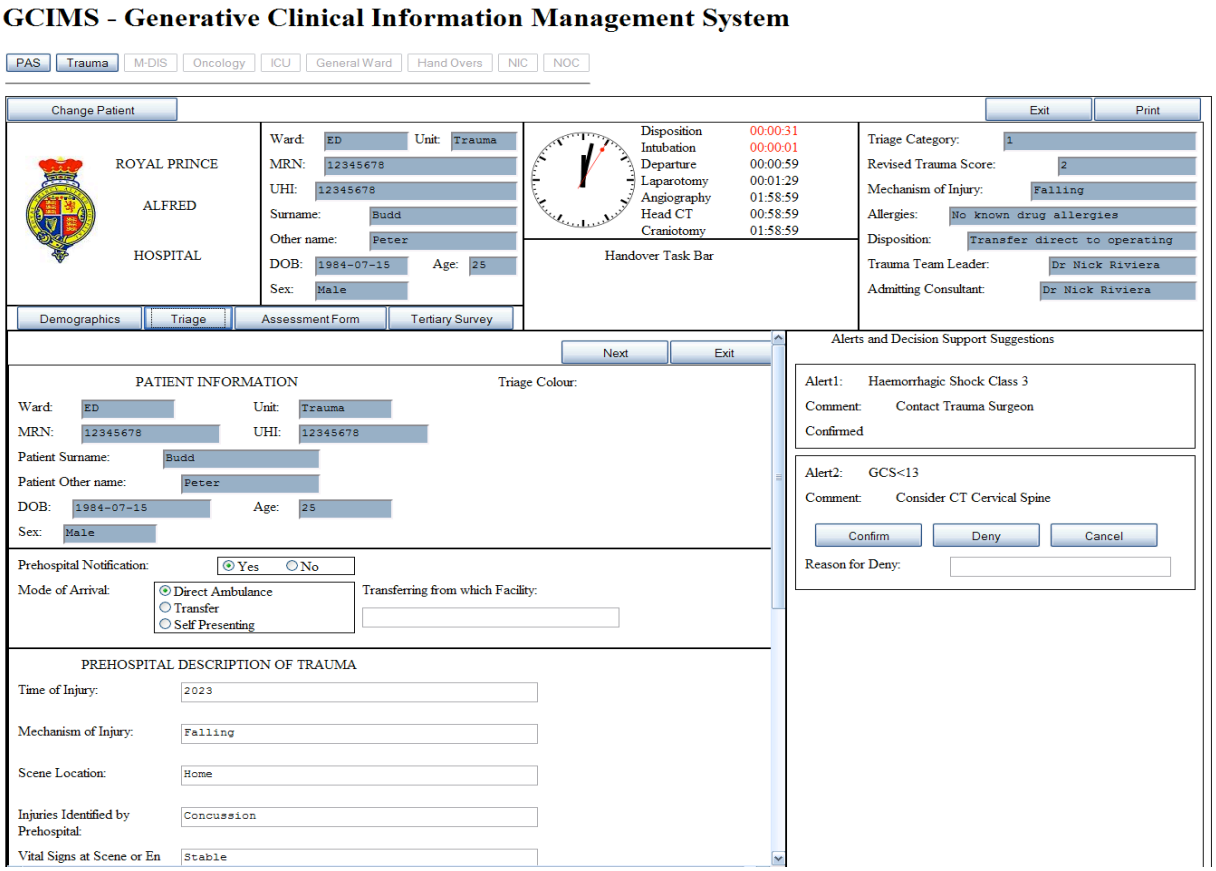
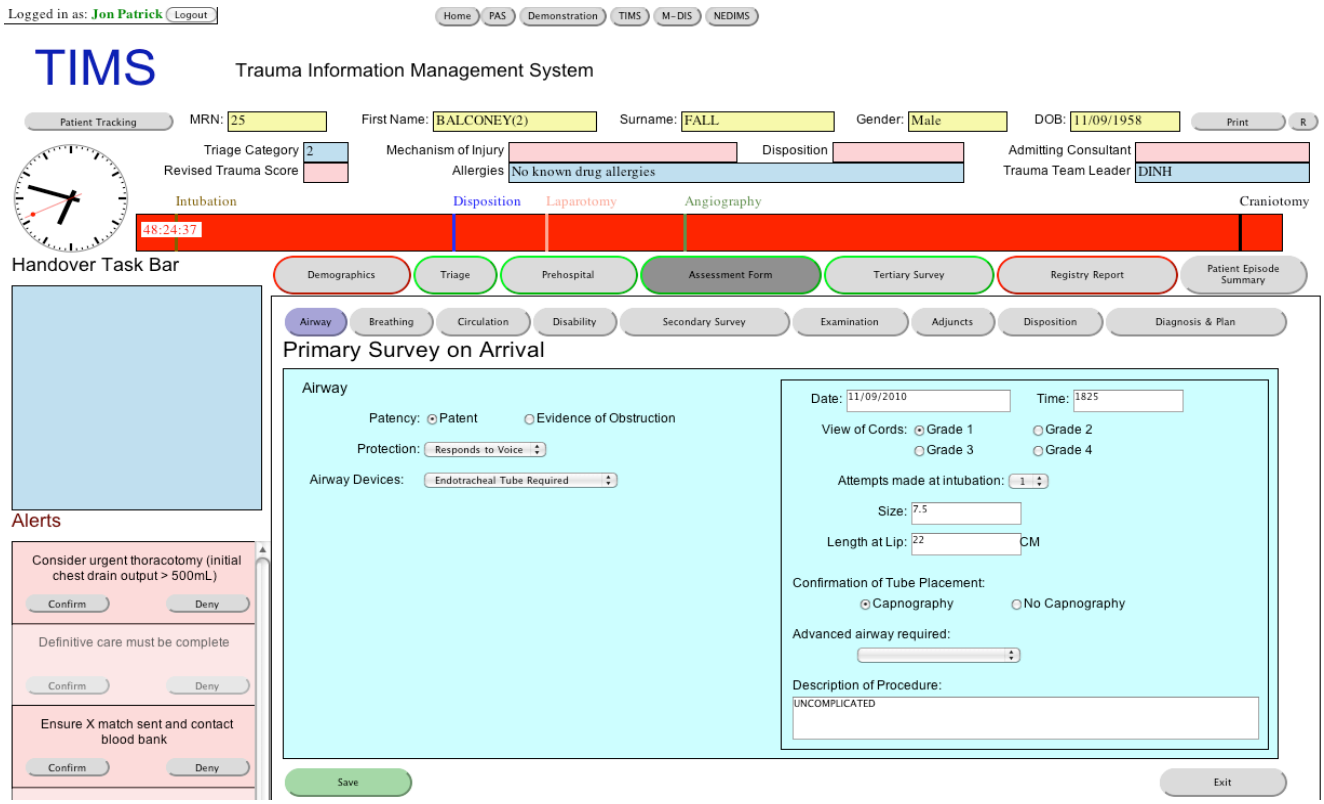
Figure 4a and 4b. Screen design for TIMS Version 1 (above) and Version 2 (below).