An Empirical Study into Governance Requirements for Autonomic E-Health Clinical Care Path Systems

Mr. Philip Miseldine[1], Prof. Azzelarabe Taleb-Bendiab[1]

[1] Department of Computing and Mathematical Sciences, Liverpool John Moores University.

Abstract. Information technology has been widely recognized as a key building block to the Government modernization agenda for the NHS, and a vital component to assisting continuous improvement in clinical practice, patient safety and standard of care. Medicine is far from a static field, and this is especially true for research into the prevention and treatment of breast cancer, which is thankfully ever changing and advancing towards more comprehensive care and therapy for the condition. With such a fluid and fluctuating set of requirements, software that aids in the delivery and prognosis of therapy faces real challenges in its design so that it can adapt successfully as and when required to new requirements in the field. This paper will discuss the challenges that designers of such software must solve, and highlights the issues facing current state-of-the-art solutions in the domain of breast cancer prognosis. The paper then introduces the notion of system self-governance to produce a rigid yet highly dynamic system that is evaluated through a case study involving several leading UK cancer hospitals. The paper concludes with analysis of how the principals introduced in the paper can be applied to the wider domain of eHealthcare.

Keywords: breast cancer prognosis, requirements engineering, ehealthcare, self governance, adaptable system design

1 Introduction

Software engineering aims for the systematic, principled design and deployment of applications that fulfil software's original promise: applications that retain their full malleability throughout their lifetime and that are as easy to modify when deployed as they are on the drawing board [18].

Whilst significant advancement in the field has been made to meet these goals through the development of high-level languages, structured design using object orientated design, and software configuration abstraction through policy definition, software remains fragile to changes in its environment and its perceived requirement. Indeed, software that facilitates diagnosis in a field such as healthcare, where advances in research coupled with the wide-ranging and unpredictable requirements of

Miseldine M. and Taleb-Bendiab P. (2005).
An Empirical Study into Governance Requirements for Autonomic E-Health Clinical Care Path Systems.
In *Proceedings of the 1st International Workshop on Requirements Engineering for Information Systems in Digital Economy*, pages 89-100
DOI: 10.5220/0001423600890100

patients and clinicians, has to adapt and react to requirement change to remain useful. It is here where such fragility in design is all too apparent.

This paper discusses how the current design of information systems is ineffective in its delivery of informatics in healthcare systems, focusing on the domain of breast cancer care by evaluating current state-of-the-art prognosis systems against recognized requirements. An approach to ease the deployment of adaptive decision processes that encapsulate requirements using distributed, highly adaptable system governance is outlined and evaluated through a wide reaching case study involving several leading cancer treatment hospitals. The paper concludes with discussion of employing the proposed programming model within different areas of eHealthcare, including a description of a dental triage study.

2 Identification and Capture of Requirement

This section will explore the exact nature of the perceived requirements for modern medical information systems and judge the suitability and effectiveness of current state-of-the-art prognosis systems to represent these requirements.

2.1 Clinical Requirements in Medical Information Systems

In a report published by the National Coordination Office for Information Technology Research and Development (NITRD) [19] it was recommended that information systems play a critical role in assisting clinical decision as recommended by medical institutions and government regulations. An evidence-based approach to the delivery of clinical care has gained wide recognition within the healthcare community, advocating that decision-making should use current knowledge and clinical evidence from systematic research [23]. Medicine is far from a static field, and this is especially true for research into the prevention and treatment of breast cancer, which is thankfully ever changing and advancing towards more comprehensive care and therapy for the condition. For software to successfully aid in the clinical decision process these initial requirements imply that systems must adapt their decision process based on current, often changing recommended knowledge.

As with much of scientific research, whilst there is general consensus on the identification of the problem domain, there is a wide range of variety in proposed solutions and best-practice recommendations. In the medical field this variety is encapsulated within published guidelines from different institutes, which clinicians adhere to based on internal preference, or through the recommendation of a governmental body. For example Christie Hospital, Manchester [2] and the Linda McCartney Cancer Research Centre, Liverpool [12], use their own internally developed set of guidelines based on the knowledge of their own respective clinicians and specialists. Any guidelines used by NHS centres (such as Christie) must also adhere to the governments own guidelines set by NICE [17].

With the wide range of requirements based on conformance to guidelines for care, there is no recognized standard method to support clinicians' decision-making processes as to how and when to include new evidence, or other models and practices from elsewhere. This means that there is no standard way to ensure that clinicians are following accepted guidelines or deviating from them.

The variety in the decision process, and the requirement for conformance to set guidelines makes coordination a central pillar in any solution to the NITRD proposals. Changes to guidelines must be reflected as soon as possible within a decision system so that conformance is maintained, and to keep the system as up to date with current knowledge as possible. As software diversity leads to data variety [9], to successfully represent variety within a guideline decision model, any system must also provide coordination in its formulation and representation so change is effectively propagated from requirement definition to actual inclusion within the system.

This is particularly relevant when individual clinician concerns are considered. Static modelling through a static system oftentimes will yield an updated decision process for an individual clinician, rather than for the organisation or institution as a whole. It is up to the responsibility of the clinician to distribute the change or new requirement, and then for this to be reintroduced into the institutions system. Some adaptation or requirement alteration might not be considered appropriate for general use within the organisation, however might be appropriate to a subset of clinicians within the institution. Again, within a static system the complexity of tracking which models have evolved for which clinicians makes this coordination impractical. This complexity is compounded by the fact that research shows uptake of guidelines by clinicians in practice is generally poor [4].

Indeed, these issues can be extended outside the scope of the medical domain. Any organisation that distribute decision processes through their systems to their members, and that require adaptation based on the position of a member within the organisation, or the personal requirement of the member, face the same issues of coordination and lack of support within statically defined information systems.

2.2 Patient Requirement and Requirement Priority

Clinical and clinician concerns are not the only dynamic requirements that a medical information system should consider, however. A strategic report for the Department of Health in particular notes that "*the involvement of patients in decisions about individual care and policies - including R&D priorities - is widely advocated but little information exists about its impact or value*" [8] and advocates that the involvement of the patient into the decision process is "*the most cost effective way to provide information to meet the needs of patients, their families, healthcare professionals and the public*". In addition, research [21] indicates that software implementations of guidelines (SIGs) are more likely to be used by clinicians if they provide patient specific advice during consultations.

Such requirements are particularly important in cancer care where proposed treatment is often invasive or can cause irreversible damage to the patient, both physically and mentally. Thus, patient consultation, and respecting their requirements for treatment are essential for successful treatment.

2.3 Computer Based Prognosis Systems and SIGs

Two complementary forms of system are used to provide computer-based prognosis in deployment. The first type of system is an all-encompassing solution that provides a service-based approach to prognosis, by assisting the clinician or patient during consultation. In general, these systems are web-based and distributed and maintained by an Application Service Provider (ASP). A notable example of an ASP delivered service in breast cancer prognosis is the AdjuvantOnline service [1] which is widely used by clinicians in the UK. Clinicians enter patient data into a predefined interface and retrieve a graphical report of prognosis and possible treatment plans. Clinicians enter the system "blind", that is, the system stores no information regarding their historical usage or preferences, and the same experience is repeated to any user of the system, without personalization. This is typical of ASP based prognosis, including the implementations Finprog [6], and OncoDoc [20]. As such, their effectiveness to deal with the issues highlighted in the previous discussion are highly limited.

The second system type is a multi-stage solution with a SIG implementation underlying a user interface that acts as an interface to query the SIG. The interface layer of these applications can fall into the first form of system, where an ASP provides the interface and the SIG is maintained externally. By controlling development of the underlying decision process encoded within a SIG, a better model for adaptation and coordination can hoped to be delivered.

A well known model is the Arden Syntax [25] originally developed as an observer system to act as a hospital "watchdog" monitoring data values. It is a text based programming language for encoding medical logic modules (MLMs), though is difficult for non-computer literate medical experts to use. It has a simple grammar, rule-based formalism and has been adapted for representation of guidelines by using interacting MLMs. Similarly augmented decision tables (ADTs) [24] have also been used to extend the rule-based functionality of Arden by augmenting rules with additional data such as probability and utility. However neither the Arden Syntax nor ADTs provide support for producing a guideline that evolves over time and with evidence accrual. This makes individual modules weak in functionality and the more complicated multiple modules guideline model, increases complexity and decreases readability, making it unsuitable for rapid refactoring and adaptation.

2.4 Limitations in Current State-of-the-art

The review of requirements given in this section, when compared against the abilities of the current state-of-the-art systems used within breast cancer prognosis show a distinct lack of correlation. Indeed, the software used today falls well short of providing even basic coordination, governance and adaptability, all essential tenants to advance prognosis through informatics.

The rest of this paper will focus on a proposed solution by promoting the notion of self-governance to coordinate adaptation and usage of the decision process, based on user requirement, and conformance regulation.

3 The Promotion of Distributed Self Governance

With systems in the field of eHealthcare and breast cancer prognosis in particular requiring such fluid and adaptable requirements, yet with strict conformance to set practices and guidelines, a notion of governance is required to reign in and coordinate software requirements within a distributed organisation where individual users of the system require tailoring of the processes of the software system.

In the static, closed systems described in section 2.3, these requirements are often either hard-coded into the decision process, or variety is introduced through input by the clinician in the form of a patient profile. In addition, the context of the user's request is discounted. Thus any redevelopment or evolution of the decision process is applied to all users, no matter their requirement. To develop adaptive, self-governing systems however, this notion of context and its effect upon the system is critical.

3.1 Providing Governance Through Context

Context is often captured by software to provide personalization services to users, and to mould the experience of using the service to the particular requirement of the user. Oftentimes this is facilitated through an authentication model as typified by AAA [27], where a user provides credentials to identify themselves within the system, and their personal experience is reproduced and tailored based on stored criteria. Such authentication is widely used within the Internet community to provide forums for discussion, to access to personal information, and to digitally sign documentation.

In effect, by providing authentication criteria, a context of the request is formed as additional, meta-level descriptions can be attached to the profile of an authenticated user. For stricter control, by applying the techniques of role-based authentication [5], users can be placed into groups with specific responsibilities and requirements. Context can therefore provide the first stage of governance within a distributed system, where assessed conformance requirements, patient profiles, and access to evolved decision models can be planned according to individual profiling.

In the context of clinical care, different organisations and institutions have different conformance models, where any decision process needs ratification against, for example, NICE guidelines. Other organisations do not require such ratification. Through context, a request can be scrutinized accordingly, providing validation for the decision process. As such, there exists a hierarchical model for conformance based on context, as illustrated in Figure 1:

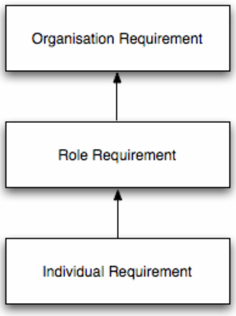

Fig. 1. Conformance hierarchy in a typical clinician scenario.

Accordingly, the individual concerns must conform to the requirements set out by the role in which the individual exists within the organisation. This is then supplanted by the requirements of the organisation as a whole. Thus, to govern adaptation of requirement, each conformance model defined above the requirement level in the hierarchy must ratify the change.

3.2 Ensuring Safety and Influence Through Governance

Thus the hierarchical model of governance is essential when defining the structures required to represent and facilitate governance throughout the end system, and thereby adapt the system based on requirement safely and predictably. Safety in computer implemented guideline systems is a critical responsibility. Safety is very important both in terms of the outputted decision, provability of the correctness of the representation and completeness. These desirable characteristics are more likely in systems where there is "*a recognised software design paradigm used, knowledge and logic are separated, a simple tool kit is used, there is a minimal feature set expressive enough to encode guidelines in a simple manner, a decision making process can be followed and clinician knowledge can be used and the decision overridden*" [7].

By constricting the goal of a decision process to be the successful conformance of the decision process with the associated conformance models specified through the hierarchy, an adaptation or alteration based on a changing requirement can be governed in a programmatic fashion. As an example, acceptance of a treatment plan is itself a decision model, actuated by a clinician. The *process* of the decision model is to recommend a treatment plan for a patient. This is the intention or requirement of the clinician. However, for the process to be successful, it must ratify itself against the conformance models specified within the hierarchy.

Such conformance can be thought of as having direct, external influence upon the decision process of a model. After all, if a process does not successfully correspond to the conformance models set out within the organisation, it cannot be completed. This paradigm for governance is typified in the R^5 architectural model [11] which states that a decision will often be made because of some internally or externally imposed formality, and as such can be modelled in the form of rules, regulations and recommendations. Thus, by applying the principals of R^5 to the requirements of conformance, a rigorous approach can be produced.

It has been discussed in the preceding sections of the importance of equipping adaptive software with the notion of governance based on context. Coordinating and actuating this model within software however, produces many challenges. Any solution must distribute the currently held conformance models amongst the organisation, so that any adaptation requirement at any level of the system can be ratified. Adaptations to requirements must then be actuated and distributed so their effect is included at the required level within the organisation.

At a much lower level however, there needs to exist a methodology that supports adaptable decision models, or SIGs. As discussed in 2.3, SIG implementations used in the breast cancer prognosis are either static and hard-coded, or too complex to enable adaptation during the execution of the system. In addition, for information systems to support autonomic behaviour [26], that is, a systems ability to adapt itself to perceived change, the system must be able to reason upon its own construction, and have the necessary functionality to enable itself to adapt.

4 Implementation

4.1 Requirement Representation Using Neptune

In earlier work [16] the author's introduced a programming language, Neptune, that produces an object form of guidelines that can be inspected, modified and deliberated upon at runtime. Neptune objects encapsulate the logic and assignments expressed within a guideline in a transparent, introspective object notation, allowing fine-grain adaptation of the decision process to take place. Accordingly, requirements of guideline conformance and user preferences can be encoded in such a form that they can be both interpreted and adapted dynamically with or without direct human interaction.

As an example, Fig.2 shows a section of the NICE guidelines for operative breast cancer care, encoded within Neptune:

```
define rule calcNPI
        variables.sd = 0.2 * dataset.tumoursize + dataset.nodestatus + dataset.histological

        if (variables.sd <= 5.399)
            variables.npi = 2
    else
                variables.npi = 1
    end if
end define
```

Fig.2. A rule as defined within the NICE guidelines for operative breast cancer care, encoded within Neptune

A comprehensive guide to the language syntax is given in [15], and subsequent discussion of its structural design is outside the scope of this paper. Of importance however, is how this formulation of a rule in language is represented computationally as an object, allowing object orientated programming models to be applied to the decision process. Object serialisation techniques, allowing an object to be stored, transferred and retrieved from information repositories, can thus be applied to Neptune objects, allowing the safe and efficient distribution of decision models. In this way, Neptune allows the modification and enhancement to the decision process during any point of its lifetime whilst maintaining efficient transportation and storage.

4.2 Applying Self Governance Through CA-SPA Policy Modelling

Policies are a means of specifying and influencing governance behaviour within a distributed system, without directly including the behaviour within the management services themselves [13] unlike the approaches outlined in section 2.3. A policys' intent commonly takes one of two forms: authorisational and obligational. Authorisation policies specify what activities a service is permitted or forbidden to do to a set of target objects and are similar to security access-control policies. Obligation policies specify what activities a service must or must not do to a set of target objects and essentially define the duty of a service. As such, the obligation policy provides a suitable model for conformance enforcement.

Policy intentions are complex statements that cannot easily be encoded computationally. Much work has been undertaken [22]; [27] to represent policies using language based methodologies, which has shown that whilst policies cannot be easily represented in terms of attributes or directly translated into method definitions, the traditional programming language constructs, they are often realised through a combination of attribute and method implementation. Such methodologies however generally lack the mechanism to represent and enable evolved adaptation. Neptune, as shown in [14] is well-suited to such representation, however.

By representing both the situation and required behaviour to enact upon the system entering the situation in a policy form, a defined boundary of behaviour can be produced. In other words, a model for conformance is achieved. Basing these representations within Neptune, each stage of the conformance can be adaptable as required, using the introspective nature of the language.

CA-SPA is a methodology developed by the authors to produce such a policy model, based on a situation, an action ontology, and a predicted situation that the system should find itself within after execution of the action ontology [3]. This final predicted situation is important to ensure that the intention of the policy has been completed. To demonstrate the CA-SPA model efficiently, the earlier example of ratifying a treatment plan against the conformance hierarch can be modeled as a CA-SPA policy. The only situation in which a treatment plan can be accepted, is when the treatment plan conforms to the organisation's own guidelines, and for the purposes of illustration, a set of government guidelines. These guideline models are themselves encapsulated within Neptune objects organisationGuidelines and governmentGuidelines respectively. As such, the situation for the CA-SPA is thus:

```
If (organisationGuidelines.Conform && governmentGuidelines.Conform)
```

When this situation is entered, we can accept the treatement plan, as specified within the ontology of the CA-SPA:

patientTreatmentPlan.Accept

Leaving our predicted situation, after the action ontology is executed, to represent the fact a treatment plan has been accepted for the patient:

Patient.TreatmentPlan != Nothing

5 Case Study

As part of an EPSRC sponsored project, the techniques and methodology described in section 4 have been implemented to produce a web based prognosis system using a set of tools that enable the representation of guidelines used by Christie Hospital, and the Linda McCartney Centre for Cancer Care of Royal Liverpool Hospital. This section details the benefits from a developmental perspective of the design, and how it addresses the problem domain as defined previously.

5.1 Venus: Medical Decision Support System

The medical system, known internally as Venus, produces a context of request by requiring users to log into the system. It is at this stage that users are assigned roles within the organisation based on predefined categorisation. All authentication data is stored on a database that provides a centralized repository of requirement and role data, meaning all levels of the system are granted access to this information.

At every stage of interaction within the system, the request is first ratified against a stored set of CA-SPA documents, which are themselves a federated set of Neptune objects (representing each stage of the CA-SPA). Thus, the widest set of situations can be captured, and requirement adapted as and when required.

5.2 Adaptation Based on New Requirement

Whilst the system can facilitate adaptation in its use of Neptune and CA-SPAs to represent the governance of the system, it is the ability of these methodologies to represent autonomic behaviour that is of particular interest to adaptation processes. By capturing evolving requirements during the lifetime of the application, all the constructs exist so that seamless evolution of the system occurs based on the requirements of its users. This evolution is monitored and controlled by the self-governance specified through the system, to provide validation and boundaries for the system. Hence patient, clinician, organizational, and external requirements can all be represented and enacted to produce a stable system that takes all requirement into account.

In Venus, an early example of these techniques has been produced. By analyzing the usage patterns of a system user, the interface can itself adapt to speed up the process of using the system, by tailoring its process to match that of its user. Thus, options rarely used by the user are hidden, and stages can be skipped based on usage. This is illustrated in figure 3, where to get from step A to step B within the process, a set of stages are required.

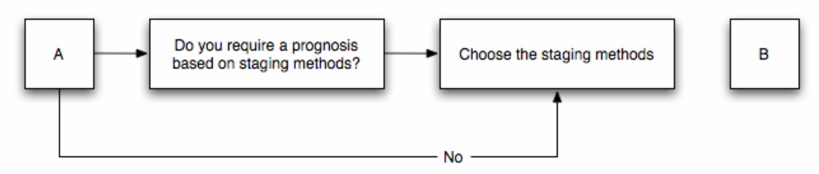

Fig. 3. Part of the Venus process

Based on a CA-SPA whose situation is entered when the user decides "No" in the above process for 5 subsequent times, these additional steps are removed from the process creating a direct path from A to B. Provision exists within the interface to restate the original process flow, if required at a later stage, as does the threshold in which steps are removed from the process (in this case, 5 subsequent uses) as this can adapt as required. In addition, patient concerns are expressed in Venus using the HADS anxiety scale [10], where documentation for a patient is produced based on the assessed anxiety (and therefore requirement) of the patient.

5.3 Application of Techniques to the Wider Field of e-Healthcare

Whilst the scope of this paper has focused on supporting breast cancer prognosis, the techniques described herein can equally be applied to other areas of e-Healthcare as shown by the production of a prototype decision support system for dental triage services, for the Royal Liverpool Hospital using the Neptune and CA-SPA methodology. The original system used by the Royal Liverpool Hospital combined a mixture of Microsoft Word and Microsoft Excel documents to provide flow through the process of triage for patients. In this way, information elicited from the patient formed the basis of decisions on their perceived health. The more critical the condition (such as persistent bleeding) the higher priority the patient was given to see a dentist.

The main requirement of the case study was to produce a system that could log patient details and their eventual classification by the system, with the end goal being that the system could adapt its classification based on historical evidence to improve the level of care given to patients. In addition, government targets facing the care of dental patient, including waiting times needs to be considered.

By encoding the process flow within a Neptune object, an instance of the object could be created for each patient as they passed through triage. Consequently, the decision process, or rather, why the patient received the classification they did, is encapsulated within the object instance itself. Storing this object instance, along with the patient's profile, comprehensive historical data was stored that could later form the basis of evidence given to machine learning and data mining processes. Another consequence of automating the storage of patient details and their path through the decision model, patients modelled in the prototype could return to the hospital at a later date, and clinicians would be presented with both the patient details and the exact reasoning behind the decisions of the information system, essential in ascertaining correct patient care.

6 Conclusions

This paper introduces the problem domain facing information systems in breast cancer prognosis, and offers a solution based on the notion of self-governance within a system to provide flexibility in representing evolving user and system requirements, whilst maintaining guideline conformance. It has been shown through the case study that the issues resulting from the problem domain can be effectively solved using the methodology introduced in this paper.

The issues facing breast cancer prognosis however, can be equally applied to a wide range of fields, in both e-Healthcare and organisations as a whole. The need for dynamic systems that adapt to perceived user and organisation requirement is felt in many sectors that use IT to automate decision processes, as typified in the second case study involving the automation of dental triage process. In the wider medical viewpoint we can see applications for our approach in other areas of medicine, such as the diagnosis and treatment of lymphoma which has a similar staging model to breast cancer.

The authors gratefully acknowledge EPSRC for funding this research paper.

References

1. AdjuvantOnline. (2005). Http://www.Adjuvantonline.Com/.
2. Christie Hospital. (2005). Http://www.Christie.Nhs.Uk.
3. D. England, A. Taleb-Bendiab, P. Miseldine, M. Randles, & Murphy, K. (2005, February 2005). *Integrating best practice and evidence based decision support for breast cancer care.* Paper presented at the Multi-disciplinary Workshop on Healthcare Information Services and Future ICTs, UCL.
4. Feder, G., Eccles, et. al. (1999). Clinical guidelines: Using clinical guidelines. *BMJ*(318), 728-730.
5. Ferraiolo, D., Cugini, J., & Kuhn, D. (1995, December 1995). *Role-based access control (rbac): Features and motivation.* Paper presented at the 11th Annual Computer Security Application Conference, New Orleans, LA.
6. Finproject. (2005). Http://www.Finprog.Org/.
7. Fox, J., & Das, S. (2000). Safe and sound: Artificial intelligence in hazardous applications. *American Association for Artificial Intelligence*.
8. Health, D. o. (1999). Strategic priorities in cancer research and development. In C. W. Group (Ed.) (pp. 58): Department of Health.
9. Hiller, M. (1998). Software fault tolerance techniques from a real time systems point of view: An overview: Chalmers University of Technology, Sweden.
10. Lampic, C., Essen, L. v., Peterson, V., Larsson, G., & Sjoden, P. (1996). Anxiety and depression in hospitalized patients with cancer: Agreement in patient-staff dyads. *Cancer Nursing, 19*(6), 419-428.

11. Laws, A., Allen, M., & Taleb-Bendiab, A. (2002). *Normative services for self-adaptive software to support dependable enterprise systems.* Paper presented at the 4th International Conference On Enterprise Information Systems, Cludad Real, Spain.
12. Linda McCartney Cancer Research Centre. (2005). Http://www.Linda-mccartney-centre.Org.Uk/.
13. Lupu, E. C., & Sloman, M. (1999). Conflicts in policy-based distributed systems management. *Software Engineering, 25*(6), 852-869.
14. M. Randles, A. T.-B., P. Miseldine, A. Laws. (2005, April 2005). *Adjustable deliberation of self-managing systems.* Paper presented at the 12th IEEE International Conference on the Engineering of Computer-Based Systems (ECBS), Greenbelt, MD, USA.
15. Miseldine, P. (http://www.cms.livjm.ac.uk/2nrich/deliverables.asp). Neptune application development guide.
16. Miseldine, P., & Taleb-Bendiab, A. (2005, June 2005). *Rainbow: An approach to facilitate restorative functionality within distributed autonomic systems.* Paper presented at the PGNET 05, Liverpool.
17. National Institute for Clincial Excellence. (2005). Http://www.Nice.Nhs.Uk/.
18. Nauer, P., & Randell, B. (1969). Software engineering - report on a conference sponsored by the nato scientific committee. *Garmisch*.
19. NCO. (2004). Revolutionizing health care through information technology. In N. C. Office (Ed.): President's Information Technology Advisory Committee.
20. OpenClinician. (2005). Http://www.Openclinical.Org/prj_oncodoc.Html.
21. Overhage, J. M., Zhou, X. H., & McDonald, C. J. (1997). A randomized trial of corollary orders to prevent errors of omission. *JAMIA, 4*(5), 354-375.
22. Ryutov, T., & Neuman, C. (2000, January 2000). *Representation and evaluation of security policies for distributed system services.* Paper presented at the DARPA Informatioon Survivability Conference and Exposition, Hilton Head, South Carolina.
23. Sackett, D. L., Rosenberg, W. M. C., Gray, J. A. M., Haynes, R. B., & Richardson, W. S. (1996). Evidence based medicine: What it is and what it isn't. *BMJ, 312*(7023), 71-72.
24. Shiffman, R. N. (1997). Representation of clinical practice guidelines in conventional decision tables. *JAMIA, 4*(5), 382-393.
25. Starren, J., Hripcsak, G., Jordan, D., & Allen, B. (1994). Encoding a post-operative coronary artery by-pass surgery care plan in the arden syntax. *Comput. Biol. Med., 24*(5), 411-417.
26. Sterritt, R. (2002). *Towards autonomic computing: Effective event management*: IEEE Computer Society.
27. Woo, T., & Lam, S. (1993, March 1998). *Designing a distributed authorisation service.* Paper presented at the INFOCOM '89, San Francisco.