World Academy of Science, Engineering and Technology
International Journal of Health and Medical Engineering
Vol:3, No:5, 2009

Expression of Security Policy in Medical Systems for Electronic Healthcare Records

Nathan C. Lea, Tony Austin, Stephen Hailes and Dipak Kalra

***Abstract*—**This paper introduces a tool that is being developed for the expression of information security policy controls that govern electronic healthcare records. By reference to published findings, the paper introduces the theory behind the use of knowledge management for automatic and consistent security policy assertion using the formalism called the Secutype; the development of the tool and functionality is discussed; some examples of Secutypes generated by the tool are provided; proposed integration with existing medical record systems is described. The paper is concluded with a section on further work and critique of the work achieved to date.

***Keywords*—**Information Security Policy, Electronic Healthcare Records, Knowledge Management, Archetypes, Secutypes.

I. Introduction

THIS paper introduces a new formalism and tool for the expression of information security controls that govern the use of medical information. The formalism, named the Secutype, and an authoring tool, named アルチ (aruchi), are designed to specify policy controls for the consistent protection of discrete, sensitive medical data items. These controls should be applied to sections of Electronic Healthcare Records (EHR), and protect the data stored by the EHR wherever it is used, according to the specifications of an information security policy. The information security policy is widely viewed as an important medium in which to specify the security controls as required by national and international legal stipulations, medical ethics and individual patient consent.

The issue of data protection has become increasingly pertinent in recent years as national initiatives have started to support greater storage and sharing of private medical information between care teams, researchers and legislators. The sharing is being achieved in part by the use of EHR Standards[1][2][3]. The commitment to sharing sensitive information is not without controversy or practical challenges: there are both legal and ethical anxieties about the impact of sharing personal data in the healthcare sector and beyond[4][5]; national and international data protection legislation surrounding the sharing of information can be ambiguous and often requires subjective interpretation[6]; hospitals and research projects are bound by institutional governance policies[7][8] which go some way to guiding how to manage security, but are also imprecise, ambiguous and require interpretation as well as understanding. Further guidance is available in the case of research projects where there is a need in many countries to apply to research ethics committees, but this process is complex, “bureaucratic and inflexible”[9], there are indications that decisions reached by different committees are not always consistent, agree only slightly[10], and have the same interpretation and application challenges.

In all cases, the writing of a policy that captures the nuances of these guidelines is recommended, but it is difficult to capture all the details for the policy to be effective, applied throughout the lifetime of the information, and unambiguously understood by all users[6]. A further significant complexity is the difference that exists in the sensitivity of different kinds of information that are being stored, and their protection requirements: for example, a blood pressure reading is unlikely to have the same concerns about privacy in the mind of the patient as their HIV status. The present generation of EHR standards help to identify what and how clinical data items are represented by defining a consistent and agreed structure for, as examples, a blood pressure reading or an HIV status within the EHR, but there is a need to specify appropriate security controls for those data items, and then to apply those controls whenever the EHR data is accessed.

In recognition of the issues surrounding information governance, a solution was proposed at the 21st International Congress of the European Federation for Medical Informatics (MIE 2008) in the form of a new formalism called the Secutype[11], which applies a knowledge management approach to the problem. The Secutype has been designed to provide a consistent format to store details about the security needs for individual data items. These data items, provided they are identifiable as a recognised semantic structure, such as an EHR Archetype[12], will be recognizable and/or mappable to a given data object within the EHR and enable the application of the relevant controls.

N. C. Lea is a Research Fellow at the Centre for Health Informatics and Multiprofessional Education, University College London, 4th Floor, Holborn Union Building, Highgate Hill, London N19 5LW (telephone: +44 (0) 207 288 3798)(email:n.lea@chime.ucl.ac.uk)

T. Austin is a Senior Research Fellow at the Centre for Health Informatics and Multiprofessional Education, University College London, 4th Floor, Holborn Union Building, Highgate Hill, London N19 (email:t.austin@chime.ucl.ac.uk)

S. Hailes is a Professor of Wireless Sytstems at the Department of Computer Science, University College London, Gower Street London WC1E 6BT United Kingdom (email:s.hailes@cs.ucl.ac.uk)

D. Kalra is a Clinical Senior Lecturer at the Centre for Health Informatics and Multiprofessional Education, University College London, 4th Floor, Holborn Union Building, Highgate Hill, London N19 (email:d.kalra@chime.ucl.ac.uk)

There are more practical challenges to information security management: available tooling to manage policy assertion is built to support specific parts of security management (like authorization, authentication or access control) but the presently available tools do not offer an holistic solution to manage the protection needs of the information at different points in its lifecycle[11], or map the use controls specified in an information security policy to the clinical data items themselves. These tools lack semantic interoperability for their configuration, which is normally managed by direct manipulation of XML configuration files, editing of other formalisms like Cassandra[6] or Ponder[13], or an administration screen that allows some capture of information. This is usually done by individuals with domain knowledge of the users, roles and assets to be protected. There is insufficient semantic power to process the detail required by the protection of medical data and no easy way to make policy controls that apply to particular kinds of data shareable across a community. It has been proposed that Secutypes are the formalism by which such tools could have a consistent configuration across all their areas of use.

The aruchi Secutype Editor has been developed to allow for the specification of Secutypes as a means to express detailed information contained in security policies. It allows for the authoring of Secutypes and for their binding with EHR data item components, to reflect the restrictions that apply to each class of clinical data. The rest of this paper discusses the development of aruchi, and how its use is planned. Some examples of policy specification are provided, followed by details on the planned integration with existing EHR systems. The paper concludes with details of further work, and a critique on the tool and approach so far.

II. Tool Development

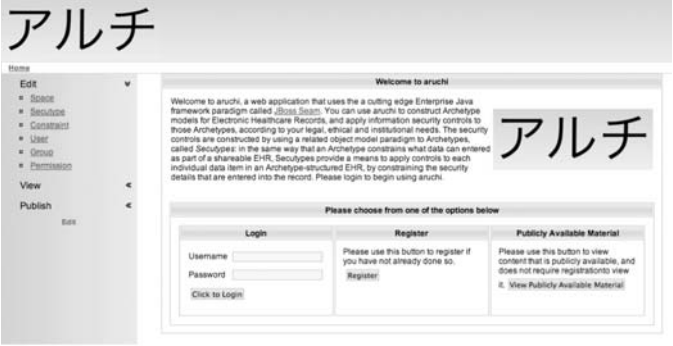
A. Secutypes

Secutypes are a set of Constraints that model how security policy controls should be applied when data is committed to a data repository. Individual Secutypes also hold unique values for Arguments, but these will only apply to the Arguments in the specific context of the Secutype that they belong to so that they can be reused in other Secutypes. Arguments, which are contained within Constraints, specify what kind of data types a Constrained Secutype can hold, be they numeric, string-based, date-time, multimedia or Boolean. These follow the basic data types within a typical engineering environment, but could incorporate openEHR[1], ISO 13606[2] and HL7[3] data types as well. They can also include ordinal lists. Labels and Permissions can be added to Secutypes, with Comments being planned for addition so that users may express opinions on individual Constraints as well as the Secutype as a whole.

In the newer domain of modeling security details, a Constraint could help a Secutype define how to express exclusion criteria for a specific area of the record (e.g. to represent an aspect of patient consent to disclosure), or the presentation of a date of birth as a Julian date under circumstances that do not warrant the release of a human understandable date of birth. For example, a research project could specify a Secutype for date of birth release, where a Constraint could be specified to capture the details about how to release dates of birth in different ways: PSEUDONIFY(String secutypeIdentifier, String method, OrdinalList context, String user) could be a Constraint with three Arguments, where the secutypeIdentifier Argument could be a string reference to a date of birth Secutype, the method Argument could be a String expression that refers to an executable instruction that converts to the date of birth to a Julian date or removes day and month (if that is what the policy instructed) and the user Argument could be used to name users of the data to whom the policy applied. Another example, this time a consent Secutype, would have an EXCLUSION(String secutypeIdentifier, OrdinalList context) Constraint, where the secutypeIdentifer Argument could refer to an HIV record, and the context OrdinalList argument would be a list of contexts - for example, "GP" or "Point of Care" or "Research".

B. aruchi

aruchi (see Fig. 1) has been developed as a web application, and is designed to run within an Enterprise Java compliant application server environment. The chosen environment is the JBoss Application Server (AS)[14], which is the environment used by an existing EHR Record Server developed at University College London (UCL). JBoss AS allows for several clinical applications to run within its environment, and integrates both the clinical applications and the record server within the same operational environment. aruchi has been developed using Java Server Faces, and can be fully integrated in this environment as well. It uses a pre-standard development framework that runs within the JBoss AS called Seam[15]. This framework has been developed by the JBoss community and helps to remove the configuration overhead of mapping Java Objects to relational databases: the framework allows for the automatic generation of tables, and full integration with Hibernate[16] so that objects can be stored and retrieved from the database without the need for complicated Structured Query Language (SQL) Queries or Hibernate configuration. This has eased the development overhead as the object model for Secutypes has been updated.



Welcome to aruchi

Welcome to aruchi, a web application that uses the a cutting edge Enterprise Java framework paradigm called JBoss Seam. You can use aruchi to construct Archetype models for Electronic Healthcare Records, and apply information security controls to those Archetypes, according to your legal, ethical and institutional needs. The security controls are constructed by using a related object model paradigm to Archetypes, called Secutypes: in the same way that an Archetype constrains what data can entered as part of a shareable EHR, Secutypes provide a means to apply controls to each individual data item in an Archetype-structured EHR, by constraining the security details that are entered into the record. Please login to begin using aruchi.

Please choose from one of the options below

Login

Register

Please use this button to register if you have not already done so.

Publicly Available Material

Please use this button to view content that is publicly available, and does not require registrationto view it.

Fig. 1: aruchi welcome screen

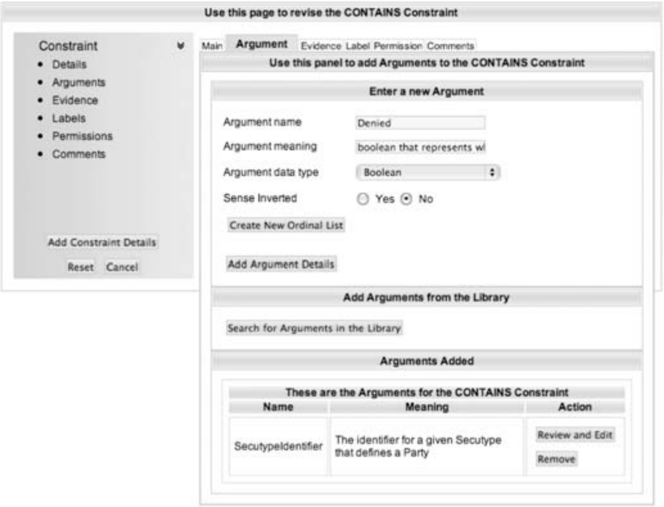
aruchi allows for the creation and editing of Secutypes and the constituent objects. Once created, other users who have appropriate permissions can view and add comments about the design of a particular Secutype or Constraint. Changes to design can be proposed, implemented and then published when contributors have had an opportunity to discuss their creation. These functions all happen within a Space, which defines the context of use for a set of Secutypes (for example, an HIV clinic, diabetes review clinic or a research project). Furthermore, they govern the details about how the models that it contains were reached, so it is a fully context driven representation of the current model, and the journey that was taken to reach it as a means to support collaborative efforts. Any given Space will also feature a snapshot view of all the Secutypes that it contains. Other classes include Permission and Comment. Permissions govern who has permission to view, edit and publish specific Secutypes and Constraints, and not permissions for access to data items that they constrain. A Permissions model to allow users accessing one Space to look at Secutypes and Constraints in another Space will be incorporated. Comments will allow for comments to be made about specific Secutypes and Constraints, and then reviewed and acted upon when a Secutype or Constraint is revised. The Space will allow you to view the comments, and therefore ntellectual journey, that has occurred to bring a specific nodel to the point of publication and use.

III. EXAMPLES OF SECUTYPES PRODUCED WITH ARUCHI

This section illustrates some examples of Secutypes created oy aruchi. Here are the security policy controls that are being epresented, followed by examples of the corresponding Secutypes. Secutypes, Constraints and Arguments needed are isted, and screenshots of the application have been taken to llustrate how the application prepares them:

Policy item: *All requests for … data from any party should be denied and logged.* This stipulation was taken from a nedical research project's information security policy.

Secutype: A proposed Secutype could be to have a Secutype called Data Request Log. It could contain two Constraints. There would be Constraint called CONTAINS, and this Constraint could hold two Arguments (see Fig. 2): one of type String called SecutypeIdentifier (where this would be the identifier of a Party Secutype, defined as a Secutype itself); another of type Boolean called Denied. Within the context of the Data Request Log Secutype, this Boolean would have an argument value initially set to true, so that the request for the data would be denied. The Constraint can then be added to the Secutype (see Fig. 3). The second Constraint could be called CONTROLS, and it would have one String Argument, called SecutypeIdentifier (or ArchetypeIdentifier if the data is held in an EHR, and Archetypes are being used to model the clinical model for the research project). Further Constraints could be defined to capture more details about the request. As it stands, an EHR system would be directed to log the fact that a request to access data had been received and denied as the policy control now forms part of the record structure. A system administrator could then follow the request up, and if it is reasonable, they could grant access to the data by the approved Party by setting the Denied argument value to false in a given instance of the data item that is being protected in this fashion.



Use this page to revise the CONTAINS Constraint

Constraint
- Details
- Arguments
- Evidence
- Labels
- Permissions
- Comments

Add Constraint Details

Reset Cancel

Main Argument Evidence Label Permission Comments

Use this panel to add Arguments to the CONTAINS Constraint

Enter a new Argument

Argument name: Denied
Argument meaning: boolean that represents wl
Argument data type: Boolean
Sense Inverted: ○ Yes ● No

Create New Ordinal List

Add Argument Details

Add Arguments from the Library

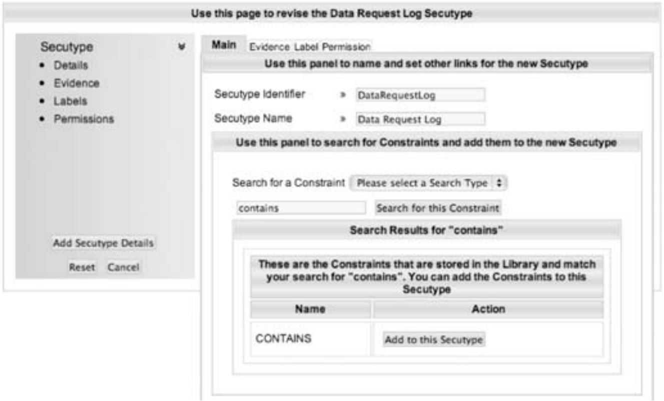
Search for Arguments in the Library

Arguments Added

These are the Arguments for the CONTAINS Constraint

Name	Meaning	Action
SecutypeIdentifier	The identifier for a given Secutype that defines a Party	Review and Edit Remove

Fig. 2: Expression of Arguments in the CONTAINS Constraint



Use this page to revise the Data Request Log Secutype

Secutype
- Details
- Evidence
- Labels
- Permissions

Add Secutype Details

Reset Cancel

Main Evidence Label Permission

Use this panel to name and set other links for the new Secutype

Secutype Identifier » DataRequestLog
Secutype Name » Data Request Log

Use this panel to search for Constraints and add them to the new Secutype

Search for a Constraint: Please select a Search Type
contains
Search for this Constraint

Search Results for "contains"

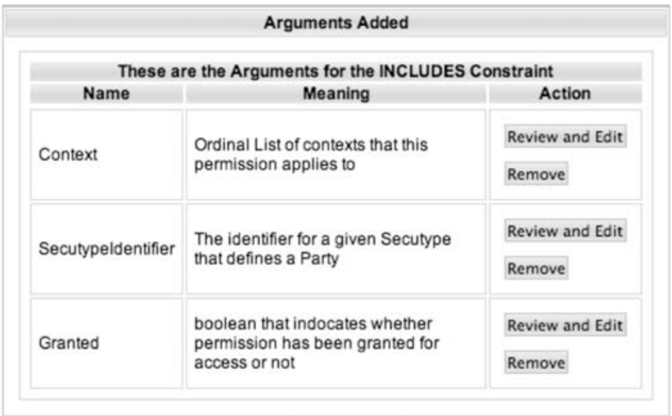
These are the Constraints that are stored in the Library and match your search for "contains". You can add the Constraints to this Secutype

Name	Action
CONTAINS	Add to this Secutype

Fig. 3: Addition of the CONTAINS Constraint to the Data Request Log Secutype

Policy item: A patient has given consent for their long term glucose control (Hb1ac) readings to be shared with other diabetes clinics to help with analyzing glucose control for a population, but does not give consent for their HIV record to be shared with anyone else at all, other than the treating clinician. This is a fictitious and credible scenario for how consent may be granted by an individual.

Secutype: In this case, a Secutype could be called Consent. It could contain a Constraint called INCLUDES. It could hold three Arguments (see Fig. 4), one called sectypeIdentifer (or again, archetypeIdentifier), which defines the record identifier that the consent relates to, in this case Hb1ac or HIV; the second Argument would be a Boolean, called Granted, and defines whether a patient has given consent for release of a particular piece of information defined by secutypeIdentifier. The third Argument could be an ordinal list of Strings called Context, which could define the contexts within which consent has been granted or denied (point of care, research project and so forth) for blanket consent. If the patient specifies contexts more specifically, the Context argument could be a String and specified further.



Arguments Added

These are the Arguments for the INCLUDES Constraint

Name	Meaning	Action
Context	Ordinal List of contexts that this permission applies to	Review and Edit Remove
SecutypeIdentifier	The identifier for a given Secutype that defines a Party	Review and Edit Remove
Granted	boolean that indocates whether permission has been granted for access or not	Review and Edit Remove

Fig. 4: Addition of three Arguments to the INCLUDES Constraint

IV. Planned Integration With Existing Systems

aruchi allows for the editing of the Secutypes themselves, whilst the Secutypes are a specification, or blueprint for how information about security policies should be constrained and stored for further use. In this section, those further uses as part of an existing system will be discussed.

Secutypes have so far been designed with a target deployment system that adheres to EHR Standards in mind. It was recognized that the EHR approach to storing information helped to share the information with greater ease and accuracy across different care teams, research users and clinical providers, but would also form an important part of a solution to assert the required security controls. The Secutypes are in effect another beneficiary of the EHR Standards as they can be applied to discrete items of data as specified by the Archetype model.

There are examples of medical records systems that the authors have worked on, and are referred to in section II B. The design of these systems include a module that asserts the EHR structure on data that is committed and retrieved, and is illustrated in the MIE Paper[11]. A module that will assert the structure of Secutypes in a similar fashion is being planned, and will allow for the control of EHR data items governed by the EHR structure module. This will include linkage between Secutypes and their target Archetypes or other Secutypes. The current method of instantiating runtime Archetypes is to generate Java classes that impose the structure and constraints specified by the Archetype, so that the data fields are populated with appropriate data, leading to a representation of an EHR for a given individual. It is anticipated that a similar approach will be used for Secutypes to facilitate integration with the current system. It is of course possible to use XML or another formalism to represent the Secutypes, but for the purposes of evaluation in a live system, the most expedient means of integration is ideal.

aruchi will have the functionality to publish and export Secutype models in a given Space as and when they are ready for use. The exported model will be integrated with the existing system as outlined. The possibility of being able to view and export to target systems is also being considered as another function.

V. Further Work

A. *Evaluation of Secutypes*

Once integrated, the Secutype approach will be evaluated. The evaluations will form three phases. The first phase will be to evaluate whether or not Secutypes assert the controls as anticipated. To achieve this, a set of scenarios that include the events that occur in the deployment and use of an EHR Server will be constructed: the scenarios will map the processes and flow of information that occurs in a research and live clinical setting when clinical information is committed to and retrieved from an EHR Server; information security policies that have been constructed for their use will inform the Secutype construction. From this, expected outcomes of data use information will be established and used to verify what data is stored or released when the Secutype controls are applied: for example, in some research cases, dates of birth are needed to establish ages at different points throughout a patient's care[17]. Researchers do not always need the whole date of birth; sometimes just a year of birth is sufficient. A Secutype could be constructed to assert the control that only years of birth should be released in a given context of use (like a research query). Use of forthcoming research projects and perhaps live clinical deployments as exemplars is planned.

The second phase will be to compare the effectiveness of the Secutype approach with a system that does not use the Secutype mechanism. Traditional methods for asserting policy controls have required a lot of manual intervention on a day-to-day basis. Under this evaluation, the controls that can be asserted using Secutypes will be attempted in the existing system; a comparison of what can and cannot be done using both approaches will be made, as well as a list of manual controls that are needed. This phase will also compare performance between the two approaches: it is anticipated that the use of Secutypes will introduce a delay in storing and retrieving data, thus introducing a negative effect on performance. Should this be proven to be the case, phase three of the evaluation would apply load-balancing techniques in terms of application server clustering, database management and possible high performance computing techniques to ease the burden.

B. *Further Functions for Secutypes and aruchi*

Once the evaluation of Secutypes has been completed, areas of interoperability, audit and security assurance will be investigated. Using aruchi as a tool to facilitate this, interoperability between the different EHR Standards will be investigated as a means of trying to encourage data provenance and accuracy when an *open*EHR model is required to work with an HL7 model. Secutypes could, in theory, be a means to control how certain concepts can be mapped between the different standards.

The institutional governance and information security standards that have been looked at encourage the logging of commitment of users to information security. Other assurance-based information and techniques exists, such as being able to access system logs from EHR Servers, operating system logs and network traffic. In terms of providing assurance for patients and research councils, presentation of the facts and basing certain assertions about the safety and security of a given system on those facts is essential. Currently, this is a slow and difficult process, but it is anticipated that Secutypes will offer the means to store those sorts of details, and aruchi will provide a user-friendly tool to review and store that information, beyond specifying how it should be stored and retrieved.

Secutypes could be used to model clinical domain models as well. Whilst the primary function is for them to model security domain concepts, the Secutype model has been designed to allow for some clinical modeling. This is an area for future work, as the primary goal is to test the effectiveness of Secutypes in managing security details, and the ability to integrate them with existing clinical modeling formalisms and standards.

VI. Conclusions

In conclusion, a critique of the work reveals some issues. At this point, there are no formal guidelines about how to produce a Secutype, given that it is a relatively new formalism. There is a lot of freedom about how to create one, and how to model the security concepts. This is advantageous insofar as it will allow users a range of options to explore ideas, but can be detrimental in terms of it being difficult to gather consensus or enable interoperability between collaborators on specific concepts. As feedback is collected from trial users in the clinical and research domains, efforts to offer guidelines or impose some restrictions will be considered.

The online editing and collaboration will offer good sharing of ideas and expertise. This will introduce security issues in terms of revealing sensitive details about the structure of both clinical and security information. Whilst a permissions based solution is being developed, it remains to be seen whether this will be sufficient, and whether Secutypes can be made recursive so that they not only protect sensitive personal information, but also the structure details themselves as well, and warrants further investigation.

The Secutypes will store the details of how systems and users should behave with the data, but there may be some extra features that should be applied to EHR Servers to enact the controls that are needed. The extent to which this will be required is not yet clear, but it is clear that a lot more work needs to be done to assert policy controls in the healthcare sector, internationally. This work aims to inform that process through continued evaluation and publication.

Given the strategies for sharing of personal information for a wide range of purposes, some to do with direct point of care and care provision improvement established by research, greater effort will need to be made to achieve the security assurance that is expected and required by law and good practice. Secutypes are presented as a foundation for these efforts, and aruchi a tool to facilitate their use and incorporation into working practice.

References

[1] ISO 13606 Health informatics – Electronic Health Record Communication Parts 1, 2 and 3, International Organization for Standardization, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40784 (last accessed 30th January 2009)

[2] openEHR Clinical Models, The *open*EHR Foundation, http://www.openehr.org/clinicalmodels/project.html (last accessed 30th January 2009).

[3] Health Level 7 Record Information Model, www.hl7.org (last accessed 30th January 2009)

[4] Consultation on the Data Sharing Review, The Foundation for Information Privacy Research http://www.fipr.org/080215datasharing.pdf (last accessed 30th January 2008)

[5] R. Thomas and M. Walport, "The Data Sharing Review, " in http://www.justice.gov.uk/docs/data-sharing-review-report.pdf (last accessed 30th January 2009)

[6] M.Y.Becker, "Information Governance in NHS's NPfIT: A Case for Policy Specification," in International Journal of Medical Informatics vol. 76 (5-6), 2006, pp. 432-437.

[7] The United Kingdom National Health Service Confidentiality Code of Practice,http://www.dh.gov.uk/en/Managingyourorganisation/Informationpolicy/PatientConfidentialityAndCaldicottGuardians/DH_4100550 (last accessed 30th January 2009)

[8] University College London Research Governance http://www.ucl.ac.uk/joint-rd-unit/ResGov (last accessed 30th January 2009)

[9] A. Slowther, P. Boynton and S. Shaw, "Research Governance: Ethical Issues," in Journal of the Royal Society of Medicine, vol. 99 (2), 2006, pp. 65-72

[10] E. Angell, A. J. Sutton, K. Windridge, M. Dixon-Woods, "Consistency in Decision Making by Research Ethics Committees: a Controlled Comparison" in Journal of Medical Ethics, BMJ Publishing Group Ltd, vol. 32 (11), 2006, pp. 662-664

[11] N. Lea, S. Hailes, T. Austin, D. Kalra, "Knowledge Management for the Protection of Information in Electronic Medical Records," in eHealth Beyond the Horizon – Get IT There, Proceedings of MIE2008. IOS Press, 2008, pp. 685-90

[12] T. Beale, "Archetypes: Constraint-Based Domain Models for Future-Proof Information Systems," in Eleventh OOPSLA Workshop on Behavioral Semantics: Serving the Customer (Seattle, Washington, USA, November 4, 2002). Edited by Kenneth Baclawski and Haim Kilov. Northeastern University, Boston, 2002, pp. 16-32

[13] M. Sloman and E. Lupu, "Security and Management Policy Specification," IEEE Network vol. 16, 2002, pp. 10–19

[14] The JBoss Community and Application Server, http://jboss.org/ (last accessed 30th January 2008)

[15] JBoss Seam Framework, http://seamframework.org/ (last accessed 30th January 2009)

[16] Hibernate, http://www.hibernate.org/ (last accessed 30th January 2009)

[17] T. Austin, D. Kalra, A. Tapuria, N. Lea, D. Ingram, "Implementation of a Query Interface for a Generic Record Server," International Journal of Medical Informatics, Elsevier, vol. 77 (11), 2008, pp. 754-764