Translating Role-Based Access Control Policy within Context

Jean Bacon, Michael Lloyd, and Ken Moody

University of Cambridge Computer Laboratory
New Museum Site, Pembroke Street
Cambridge CB2 3QG, UK
Tel: +44 1223 334600
email: {Firstname.Lastname}@cl.cam.ac.uk

Abstract. The motivation for this work derives from a study undertaken with a view to providing ubiquitous access to Electronic Health Records (EHRs) held within the National Health Service in England. Any implementation must guarantee confidentiality. In October 1999 the Cambridge Computer Laboratory's Opera group joined a consortium within the Eastern Regional Health Authority to propose an experimental architecture which included role-based access control (RBAC). Specifying a policy for role-based access has two aspects: first, the conditions for entering each role must be established; secondly, the access privileges associated with each role must be defined. Access control policy must implement public policy and its expression must be transparent to computer non-specialists. We have therefore designed and implemented a pseudo-natural language framework sufficient for both of these purposes. Policy statements are translated into first-order logic, with side conditions which are evaluated by consulting a context-dependent database, and subsequently into access control procedures.

1 Introduction

The research thrust of the Opera Group in the Computer Laboratory has been to develop a Middleware architecture in which individual services retain autonomy. Key components are the Cambridge Event Architecture (CEA) [1, 11], which offers support for generic registration and notification of events, and the role-based access control model OASIS (Open Architecture for Securely Interworking Services) [5]. The implementations of these components are interdependent; an overview can be found in [1]. A crucial practical advantage of making services autonomous in such matters as access control is that they can be managed independently; this is vital in any wide-area application that comprises a federation of independent partners.

Autonomy is not however the whole story. In such applications it is usual to find overall guidelines that each partner must respect. For example, health records contain personal data, and are subject to the provisions of the Data Protection Act. The Health Service is administered independently in England, Wales, Scotland and Northern Ireland, and each province has established its own Patient's Charter. Amongst other things, each charter establishes the patient's right to determine who may access their health records. In this paper we describe a strategy that will allow cooperating partners to express local role-based access control policy in pseudo-natural language, using the same process to specify the conditions for role entry and the access privileges associated with each role.

A prototype translator has already been built, and we describe the design principles underlying it [9]. The output of language translation is a specification in higher-order logic (HOL). This HOL specification may then be translated into first-order predicate calculus (FOPC), and also be represented in the Horn clause form required by OASIS. Java classes are then generated to enforce the policy specified; we refer to such an executable form as a (procedural) guard. In general, guards established at a local service must also take any overall policy guidelines into account.

In practice the OASIS model of RBAC is insufficient in itself to meet the needs of applications such as EHRs, in which individual patients may express additional conditions restricting access to their records. The experience in the UK is that most patients who override default policy wish to deny access to named individuals, either relatives or particular health care workers. In OASIS access *privileges* are associated with *roles*, and denial of access cannot be expressed directly; instead, RBAC policy must be supplemented by exception lists stored with individual health records. We expect that the default RBAC will be sufficient for most patients and that any exception lists will be short.

Provided that all the policies which determine local access are expressed in the OASIS model they are necessarily consistent, since the possession of additional roles can only increase a principal's privileges. More generally the component policies which contribute to the generation of a local guard must be expressed in a common logical formulation so that they can be checked for potential inconsistency. A general approach to resolving conflicts between policies in a distributed system is described in [10]. The guard generated should implement the synthesis of the policies expressed, and should carry a version number sufficient to identify the high-level policy sources from which it is derived. In an environment as complicated as the NHS it will only be possible to ensure that the policies enforced are current if guards can be generated automatically. We intend to store policies and their logical representations in active databases, and to use the CEA to ensure that changes are propagated throughout the system. At present we have prototypes for a number of components, but a great deal of work remains to be done.

2 Electronic Health Records: a Federated Management Problem

The UK National Health Service (NHS) has been underfunded over a long period, and is recognized as being in crisis. The new Labour government that took office in 1997 has made reviving the NHS one of its prime goals [14]. [15] outlined an implementation strategy intended to lead progressively to the integrated storage of health data, with access from all health care points. The strategy was based on bottom-up deployment, and there was no clear explanation of the mechanisms that would ensure compatibility across the country as a whole. The Opera group joined a consortium (EREHRC) formed by health care providers within the Eastern Region, coordinated by the Clinical and Biomedical Computing Unit based at Addenbrooke's Hospital in Cambridge. The EREHRC submitted a proposal to the NHS for a "pan-community demonstrator", focussing on what we see as the main obstacles to the introduction of EHRs: heterogeneity, local autonomy, and above all continuing local evolution - of hardware and software, management structures, and medical practice and taxonomy. The proposal was not funded, but it was well received, and there is some evidence that the NHS IT Authority is now more aware of the difficulties of a once-off deployment. In July 2000 a long White Paper [16] appeared just as Parliament was going into recess, presenting a new plan for reviving the ailing Service, but without emphasis on ubiquitous access to EHRs.

2.1 Browser-Based Access to Virtual Health Records

The EREHRC proposal contained a separate technical appendix developed by the Opera group, together with a sabbatical visitor, John Hine, from the Victoria University of Wellington, New Zealand. Specific proposals for a certificate authority suitable for supporting OASIS in a health care environment are described in [6]. An overview of the architecture is given in Figure 1.

A patient's Electronic Health Record (EHR) comprises many patient records, each relating to some aspect of treatment and stored by the provider of the unit of treatment. A crucial feature of the design is the use of a **virtual health record**, essentially an index structured according to a medical ontology, containing references to all the patient records relating to a given individual. The Index service of Figure 1 is responsible for storing and maintaining these virtual health records and for ensuring they are highly available. By law every access to an individual's EHR should be recorded, and we provide for an asynchronous audit, noting the identity of the data item, the principal accessing it, and the context. This context must include information sufficient to identify the policy that was current at the time of access.

2.2 Privacy Requirements for Virtual Health Records

The architecture developed by the EREHRC requires that each elementary patient record held at a provider service be given a unique identifier. This is essential for subsequent location of the record via the index service (a Universal Resource Name (URN) might be included) and for the records of the Audit service. Provider services report new data items to the Index service as they are generated; each item is incorporated in the index structure corresponding to the

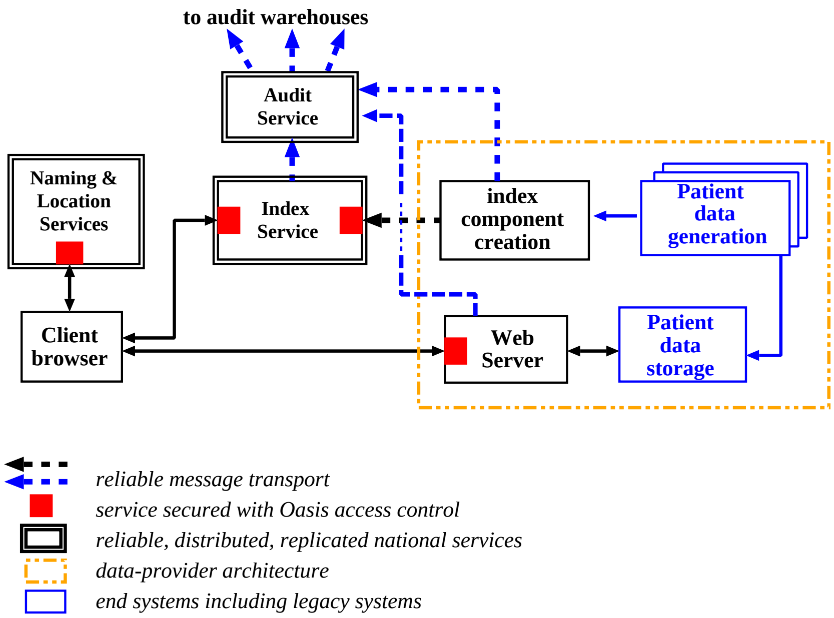

Fig. 1. An Architecture for an Electronic Health Record Service

patient in question. The Index service is essentially a replicated and highly available directory service; it must be reliable, but there is no need for immediate update as new records are created.

How suitable is the OASIS model of RBAC for expressing access control policy in this context? We believe that the model is suitable for many of the services that store patient records, and the language design experiments reported in later sections address this need. Preferences expressed by individual patients relate to their own health records, and it is natural to store exception lists with the index structures, which serve as a complete description of the electronic health data held for each individual. RBAC may be sufficient for expressing the default policy at the Health Record service, but other mechanisms will be needed for explicit denial of access.

2.3 Legal and Ethical Constraints Governing Access

Health records are personal, and therefore subject to the provisions of the Data Protection Act. The Patient's Charter [13] guarantees a patient's general right to restrict access to their health records. Ideally an access control guard should respect this regulatory framework as well as explicit policy, but it is hard to express law within logic [7]. Recent advances in biology are undermining the assumptions lying behind the patient's right to determine access; knowledge of one's genotype can affect one's kin as well as oneself. Genetic counsellors report that cases in which family members disagree are becoming more frequent, and a committee has been set up to look at the ethics relating to such shared medical data. When it becomes commonplace for genome data to be part of every health record, access control policy will need to take note of the ethical criteria that are established.

3 Role-Based Access Control: Horn Clause Logic

In OASIS each service is responsible for classifying its clients into named roles. A role definition language (RDL) is provided so that services may specify precise conditions for clients to enter each role. Figure 2 shows a service secured by OASIS access control. Policy is checked on role entry and on service use.

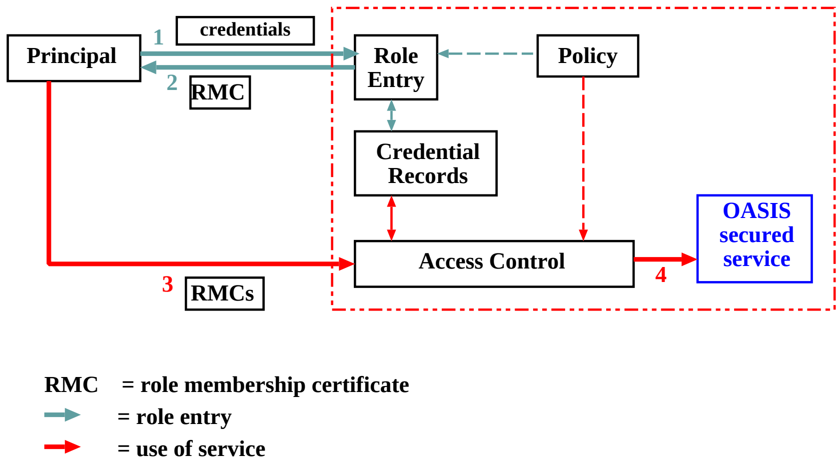

Fig. 2. A service secured by Oasis access control

RDL is a formal logic based on Horn clauses. A client becomes authenticated by presenting credentials to a service that enable the service to **prove** that the client conforms to its policy for entry to a particular role, and the client is then issued with a role membership certificate (RMC). The RMC may include parameters identifying aspects of the client, for example a local user identifier on a log-in certificate; a digital signature is generated to protect the parameter values and to ensure that the certificate is useless if stolen.

In OASIS services authorise clients by specifying the methods that they may invoke. A formal language, also based on Horn clauses, expresses the privileges associated with each role; the policy may include constraints which must be checked when access is attempted.

RBAC has a number of advantages. Permissions are expressed in terms of roles that may be adopted by principals. The policy governing role entry is decoupled from the rights associated with each role, which may be modified at a generic level. This leads to essentially scalable policy management, and incidentally enables secure access by anonymous principals, should this be desired.

3.1 Checking Constraints on RMC Parameters

The OASIS model includes provision to check side conditions on the parameters supplied with RMCs, typically by database look-up. For example, in a primary care context a group of doctors, called General Practitioners (GPs), share premises but have individual registered patients. Access to some information may be restricted to the patient's own GP rather than anyone who has the role GP in that practice. Side conditions will be evaluated within each environment, and the procedural guards at each service will consult a local database.

Two services which administer equivalent resources may recognize the same roles, and be made subject to the same policy; thus, the haematology departments of two hospitals within the same health authority may be managed according to a single policy. On the other hand different health care workers will be employed at the two hospitals, and side conditions will be evaluated in separate environments.

3.2 Failure of Expressive Power in OASIS RBAC

Individual access can be authorised within RBAC provided that the identity of the principal is available as well as the roles that the principal has entered. For example, in OASIS an RMC is issued to a principal on role entry. The encryption-protected fields of the certificate must contain the role name and issuing service but may also include additional parameters. It is possible to include a role that establishes a global name for the principal, and to require that this RMC is presented at every access. A possible name is the NHS-identifier that each patient and employee may soon carry on a smart card.

Within the strict OASIS model it is not possible to deny access in accordance with a patient's expressed wishes even if the identity of the client is known, since the model is permission based. In addition, there is no mechanism for expressing role conflicts [12], though it should be fairly straightforward to handle dynamic separation of duties by extending the semantics of the side conditions in RDL.

4 Managing Policy for Role-Based Access Control

4.1 Expressing Policy for Role Entry (Authentication)

Role Entry is expressed in the Role Definition Language. Formally, Role Entry statements are Horn clauses taking the form:

$$RoleName(arg, ..) \leftarrow RoleRef(arg, ..)[\wedge RoleRef(arg, ..)]$$
$$[: SideCondition(arg, ..) \wedge ..]$$

For example:

$$SeniorHaematologist(x) \leftarrow Haematologist(x) \wedge SeniorDoctor(x)$$

which states that a principal in the roles of Haematologist and Senior Doctor can take on the role of Senior Haematologist. The arguments may be either constants or variables when the policy is stated in this form. The values of variables will be instantiated at the time of role entry from information about the user established at log on, or from auxiliary information that is retrieved by database look-up.

4.2 Expressing Policy for Method Invocation (Authorisation)

Authorisation of clients is also expressed by Horn clauses in an identical form:

$$MethodRef(field, owner, invoker) \leftarrow RoleRef(arg, ..)[\wedge RoleRef(arg, ..)]$$
$$[: SideCondition(arg, ..) \wedge ..]$$

This establishes a right for clients who hold the nominated RMCs to invoke the stated method, subject to the side condition expressed. This side condition may link the parameters of the method invocation to parameters of the RMCs. Note that any service interface may be authorised in this way, and data access is not restricted to read and write. For example

$$append(haematology\text{-}field, y, x) \leftarrow SeniorHaematologist(x)$$

4.3 Side Conditions and Database Lookup

Restrictions can be imposed by having constants, or matching variables, in the argument for the role or method references. This forces matching at the time of a principal's role entry or data access request. Further restrictions in the form of side conditions are also required to provide enough expressive power. These side conditions will take the form of relations or methods that will be looked up in a local database. For example:

$$Invoke\text{-}read(contact\text{-}details, y, x) \leftarrow GP(x)[gp\text{-}of(x, y)]$$

establishes a right for clients who hold the role GP to read the contact details of their patients.

5 Expressing Policy in Pseudo-Natural Language

Access control policy must implement public policy and its expression must be transparent to computer non-specialists. By defining a subset of full English that can be deterministically translated into logic, policy statements intelligible to non-specialists can be defined. Such a subset of English defined in this way has been named Controlled English. Controlled English has been described in [3] and its use as a front-end to a model generation method is described in [4].

5.1 Language Design using Controlled English

It is impossible to restrict the grammar of Controlled English to eliminate ambiguity without reducing the expressive power to an unusable degree. Where ambiguous sentences can be expressed there is a choice of approaches. Either two or more readings can be returned to the user who then makes a selection, or a fixed interpretation can be forced on readings. The role-entry and access policy statements must be formal and unambiguous so we favour a hard definition, allowing a deterministic interpretation from the natural language statements as originally written. The designer of the grammar should however attempt to ensure that ambiguity is kept to a minimum, and that where an interpretation must be forced it is the more common one.

Role-based access control by its nature restricts the type of English statements that need to be expressed. One of the most important restrictions is in the universal nature of statements; they must say something about **all** principals in a role. Furthermore, in access policy, the statements must say something about all fields of a certain type that may be accessed. The statement:
"A GP can read all his patients' details"
should not be used in such a policy expression. In addition to the statement's contextual ambiguity, in the conventional logic of natural language it only implies that there is a single GP who can read all his patients' details. All statements must therefore start with the words "every" or "all", and not "a".

5.2 Representing Policy in Higher Order Logic

The form of higher order logic that has been chosen for the initial conversion from the Controlled English statements is Discourse Representation Theory [8]. This form of logic has distinct advantages over more traditional Webber style logics. Clearer, more compact representations result from pronouns, and "donkey sentences" (e.g. "every farmer who owns a donkey beats it") can be translated.

Discourse Representation Structures (DRSs) are built up compositionally through a unification based grammar. We use a subset of Controlled English; in particular, only universal quantification can occur. Both the grammar and the lexicon are therefore restricted, which in turn limits the DRS structures that can arise. This simplifies the later conversion to FOPC.

Initially the DRSs may contain unresolved pronoun constructs. During the first pass, when constructing the logic, potential antecedents for pronouns are threaded through the syntax tree, and the possible alternatives are stored for each pronoun appearing in the statement. The pronoun references are then resolved according to set rules.

5.3 Translation to First-Order Predicate Calculus

DRSs are essentially equivalent in expressive power to (function free) first order predicate calculus. Translation from the DRSs to FOPC is simple. Take for example the statement:
"Every GP can read all his patients' contact details"
The resolved DRS for this statement is shown in Figure 3.

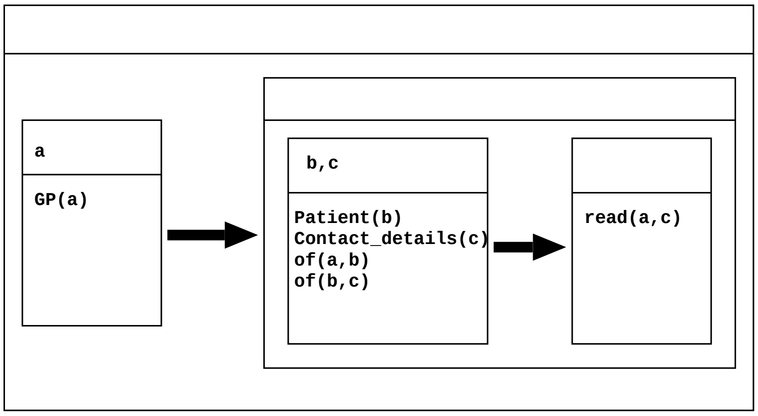

Fig. 3. An example Discourse Representation Structure (DRS)

A set of simple rules translates this into the FOPC formula:

$$\forall a.(GP(a) \Rightarrow \forall bc.(Patient(b) \wedge Contact\text{-}details(c) \wedge of(a,b) \wedge of(b,c) \Rightarrow read(a,c)))$$

This formula is equivalent to:

$$\forall abc.(GP(a) \wedge Patient(b) \wedge Contact\text{-}details(c) \wedge of(a,b) \wedge of(b,c) \Rightarrow read(a,c))$$

Converting the original FOPC formula to Horn clauses is made simpler by the limited nature of the HOL structures that can be formed from the grammar. In fact these structures can be enumerated, and it is possible to convert to the authorisation language directly by simply recognising the DRS pattern. The natural language makes explicit that b references some patient, and that the contact details of that patient may be read. The natural language side conditions that establish these relationships are subsumed by the field selector in the method invocation, and they must be discarded when generating the side conditions to be checked by the procedural guard.

The example above would be translated to:

$$Invoke\text{-}read(contact\text{-}details, b, a) \leftarrow GP(a)[Patient(b) \wedge of(a, b)]$$

It is useful to do some preprocessing on the statement in this form. There are two common problems that arise. First, the "of" side condition that results from the possessive needs to be expanded. This can be done simply by examining the referent of the "of" condition and altering it accordingly. This would result in:

$$Invoke\text{-}read(contact\text{-}details, b, a) \leftarrow GP(a)[Patient(b) \wedge gp\text{-}of(a, b)]$$

The second problem is conditions that are implicit in the context, for example $Patient(b)$ in the statement above. If we generate a check to look up (a, b) in the *gp-of* relation, and this check is satisfied, then we can infer from the database schema that b **must** be a patient. The final form of the translation into the authorisation language becomes:

$$Invoke\text{-}read(contact\text{-}details, b, a) \leftarrow GP(a)[gp\text{-}of(a, b)]$$

6 Monitoring System Behaviour and Managing Policy Change

The policy language family described above is particularly suitable for expressing OASIS authentication and authorisation, and we shall investigate its use in specific contexts. The guards generated at services which administer patient records will check that each client has presented certificates which together satisfy a set of preconditions for the access requested. By law the access made and the principal requesting it - here represented by the credentials supplied - must be written to the audit trail. Guards must be kept up to date, and the audit record should be sufficient to identify the policy context under which access was authorised.

6.1 Design and Use of Audit Records

One strength of the OASIS model is that whenever access takes place the procedural guard proves that it is authorised. The context in which the proof takes place includes the arguments to the method invocation, the certificates presented and a representation of the current policy. When a patient record is read, the audit record should identify the patient, the specific data item, the certificates on which the proof depends and a code sufficient to identify the local and higher-level policies from which the procedural guard was derived.

The audit trail so established contains a lot of information about the way the service operates. In the EHR example, it is easy to imagine a number of audit warehouses, each serving a different function. There is a statutory requirement to ensure that patients can know who have seen their health records, and accesses will be indexed by patient identity. It is equally important for health workers to be aware that every access will be recorded, both to inhibit casual curiosity and to restrain them from being subverted by agents of interested parties such as insurance companies and journalists.

In [2] a system is described which allows staff throughout a hospital to access EHRs. In an emergency any health care worker can enter a role which overrides the normal privacy checks. This has worked well, because staff really believe in the effectiveness of the audit trail, and there has been no abuse of the emergency privileges that are available.

6.2 Responding to Changes of Policy

Access to an individual's EHRs is regulated in a variety of ways. In particular, records are personal, and therefore subject to Data Protection legislation, as enacted in both the UK and European parliaments. In each UK province there is also a Patient's Charter, which among other things makes explicit the right of each patient to determine who may access their health records [13]. Health authorities will express policies in terms of the role and seniority of their staff, and the nature of the data that is to be accessed or service that is to be managed. Specialist departments will recognize professional skills in addition to seniority.

All of these sources of policy must be respected when generating procedures (Java classes, in our case) to implement the access control guards on databases which contain patient records. For each high-level source non-specialists should be able to express policy intuitively; the large scale of an application such as the NHS means that guards on the individual databases must be generated automatically. Audit records must identify the policy regime under which each access has been authorized.

7 Evaluation and Future Work

Although this work is in its early stages it has shown that the use of roles in policy specification leads to economical expression of generic policies. We have focussed on access control policy, using a National Electronic Health Record application as an example. We have demonstrated that, starting from the expression of policy in a pseudo-natural language, it is feasible to generate code which carries out access control checks automatically.

We already favour RBAC as the most promising approach for meeting the access control requirements of large scale, continually evolving applications comprising independently developed, heterogeneous components. We have developed a RBAC system OASIS, an open architecture for securely interworking services, and have used the OASIS role entry policy, enforced through its role definition language (RDL), in the work reported here.

We have explored the limits to the expressive power of RBAC; in some situations individuals need to be named explicitly. OASIS easily meets this requirement since its encryption protected certificates may include any number of protected fields, including the holder's name, if identification rather than anonymity is required. At the level of policy expression, such requirements are expressed as side conditions on the parameter values. Alternatively, for an application such as the NHS where identification is a universal requirement, it may be preferable to create an explicit role for the purpose. Principals then hold a principal-specific role membership certificate which proves their authenticated identity for the duration of a session.

This is work in progress. A three-year EPSRC grant starting in October 2000 will develop active policy stores, so that local resource monitors can be updated automatically when a primary policy source is modified. The initial validation of the work will be for access control, but we have other applications in mind, including Service Level Agreements with major communications vendors. We intend to store both primary policies and their intermediate representations in active object-orientated databases, and will make access control to the policy store an essential part of the project. The high bandwidth and reliability of modern communications make it inevitable that applications will be federated across a wide area, with individual management domains interacting subject to a high-level regulatory framework. Many problems remain to be solved before automatic enforcement of expressed policy can become a reality.

8 Acknowledgements

We acknowledge EPSRC's support for the continuation of this work under GR /N35786 "Access Control Policy Management". We are grateful to Steve Pulman for his guidance on Natural Language Processing theory and practice and to Walt Yao for sharing his experience of OASIS implementation. We thank the anonymous referees for generally constructive criticism and for pointing out several errors of detail.

References

1. Bacon, J., Moody, K., Bates, J.,Hayton, R., Ma, C., McNeil, A., Seidel, O., and Spiteri, M.: Generic Support for Asynchronous, Secure Distributed Applications. IEEE Computer Vol. 33(3) (March 2000) 68–76
2. Denley, I., and Weston Smith, S.: Privacy in clinical information systems in secondary care. British Medical Journal 318 (May 1999) 1328–1331
3. Fuchs, N.E., Schwertel, U., and Schwitter, R.: Attempto Controlled English - Not Just Another Logic Specification Language. Lecture Notes in Computer Science, Vol. 1559. Springer Verlag, Berlin, Heidelberg and New York.(1999) 1–20
4. Fuchs, N.E., Schwertel, U., and Torge, S.: Controlled Natural Language Can Replace First-Order Logic. Proceedings 14th IEEE International Conference on Automated Software Engineering, IEEE Computer Society Press, (1999) 295–298
5. Hayton, R., Bacon, J. and Moody, K.: OASIS: Access Control in an Open, Distributed Environment. Proceedings IEEE Symposium on Security and Privacy. IEEE CS Press, Los Alamitos, Calif. (1998) 3–14

6. Hine, J.H., Yao, W., Bacon, J. and Moody, K.: An Architecture for Distributed OASIS Services Proceedings Middleware 2000, Lecture Notes in Computer Science, Vol. 1795. Springer-Verlag, Berlin, Heidelberg and New York. (2000) 107–123.
7. Jones, A.J.I., and Sergot, M.J.: On the Characterisation of Law and Computer Systems:The Normative Systems Perspective In Deontic Logic in Computer Science: Normative System Specification Meyer,J.-J.Ch., and Wieringa, R.J.(eds), John Wiley and Sons (1993)
8. Kamp, H., and Reyle, U.: From Discourse to Logic: Introduction to Modeltheoretic Semantics. In Natural Language, Formal Logic and Discourse Representation Theory, Vol.1 and 2, Kluwer (1993)
9. Lloyd, M.: Conversion of NHS Access Control Policy to Formal Logic. MPhil in Computer Speech and Language Processing, University of Cambridge, (2000)
10. Lupu, E. and Sloman, M.: Conflicts in Policy-Based Distributed Systems Management. IEEE Transactions on Software Engineering Vol. 25(6) - Special Issue on Inconsistency Management, (Nov/Dec 1999) 852–869
11. Ma, C., and Bacon, J.: COBEA: A CORBA-based Event Architecture. In Proceedings of the 4th Conference on Object-Oriented Technologies and Systems (COOTS-98), USENIX Association, Berkeley, (April 1998) 117–132
12. Simon, R. and Zurko, M.: Separation of duty in role-based environments. In Proceedings of the 10th IEEE Computer Security Foundations Workshop, Rockport, Mass., (June 1997) IEEE CS Press, Los Alamitos, Calif. 183–194.
13. The Patients's Charter (for England), January 1997
see http://www.doh.gov.uk/pcharter/patientc.htm
14. UK Government White Paper, "The New NHS: Modern, Dependable", December 1997
see http://www.doh.gov.uk/nnhsind.htm
15. UK Government White Paper, "Information for Health", September 1998,
see http://www.doh.gov.uk/nhsexipu/strategy/index.htm
16. UK Government White Paper, "The NHS Plan - A Plan for Investment, A Plan for Reform", July 2000,
see http://www.nhs.uk/nationalplan/