An Authorization Model for E-consent Requirement in a Health Care Application

Chun Ruan[1] and Vijay Varadharajan[1,2]

[1] School of Computing and Information Technology
University of Western Sydney, Penrith South DC, NSW 1797 Australia
{chun,vijay,yan}@cit.uws.edu.au
[2] Department of Computing
Macquarie University, North Ryde, NSW 2109 Australia
vijay@ics.mq.edu.au

Abstract. More and more coordination of health care relies on the electronic transmission of confidential information about patients between different health care services. Since the patient data is confidential, patients should be able to delegate, give or withhold e-consent to those who wish to access their electronic health information. Therefore the problem of how to represent and evaluate e-consent becomes quite important in secure health information processing. This paper presents an authorization model for e-consent requirement in a health care application. The model supports well controlled consent delegation, both explicit and implicit consent and denial, individual based or role based consent model, and consent inheritance and exception. A system architecture for e-consent is also presented.

Keywords: Authorization, e-consent, access control, security management

1 Introduction

Computer information processing and electronic communication technologies play an increasingly important role in the area of health care. More and more coordination of health care relies on the electronic transmission of confidential information about patients between different health care and community services. However, since the patient data is confidential, the need for electronic forms of patient consent, referred to as e-consent [3] has to be considered. Patients should be able to delegate, give or withhold 'e-consent' to those who want to access their electronic health information. That is, the secure health information technology needs to support confidential patient and service provider interactions.

The presence of an electronic distributed environment means that patient information will be able to be distributed over the network. More clinical workers in a greater diversity of locations can access it more often and more easily. Consequently, without the existence of some e-consent mechanism, such widespread information could be accessed by unauthorized individuals, or used for purposes not originally consented to by the patient, which can lead to substantial

J. Zhou, M. Yung, Y. Han (Eds.): ACNS 2003, LNCS 2846, pp. 191–205, 2003.

breaches of personal privacy. The main application areas that need e-consent are those that support coordinated health care. This is characterized by data-sharing among multiple teams of health care professionals and institutions, and uses and disclosures of patient data that are additional to those that have hitherto occurred. The aim is to ensure that appropriate patient authorization is obtained in any access to patient information, which would not normally be accessible to a practitioner. In this way, the patients are able to actively participate in the decision making and in the governance of the health services they need.

In an electronic environment, the patient consent is represented and stored in a computer, and the existence of it is determined by automatic processes. For example, a set of computer rules can be defined to determine whether clinical staff working in a hospital might have their right to access electronic patient records. This is actually the so-called access control in computer information security. A number of characteristics of consent are important, including: consent delegation and delegation control; consent inheritance and exception; the explicit and implicit consents; conflict resolution policy when consent and denial exist at the same time for an individual or a specific role; the specificity and boundedness of consent; consent update and revocation. Therefore, a flexible e-consent system is needed which is capable of representing and evaluating a complex set of consent instructions with inclusions and exclusions conditions to access information.

This paper is concerned with e-consent in relation to health data disclosure. While much is known about the access control technology, very little work exists to determine how a patient's consent to view private information is expressed and evaluated in an on-line electronic environment. Based on our previous work [6], we will present a logic based approach to representing and evaluating patient consent to the disclosure of their data, with particular emphasis on security. We will first present an architecture for e-consent system within health care and examine a range of e-consent requirements. We then show how our proposed authorization models can well support these various e-consent requirements.

The rest of this paper is organised as follows. Section 2 discusses access control models and and their application on e-consent. Section 3 presents electronic patient records. Section 4 proposes a system architecture for e-consent system, while Section 5 presents a logic approach to different e-consent requirements. Finally, Section 6 concludes the paper with some remarks.

2 Access Control Models and E-consent

2.1 Discretionary Access Control and Logic Based Access Control

Access control is needed in any secure computer system that provides for controlled sharing of information and other resources among multiple users. It comprises all system mechanisms that are required to check whether an access request issued by a particular user is allowed or not, and moreover all mechanisms that are required to enforce the decision accordingly. Access control models, or authorization models, provide a formalism and framework for specifying, analyzing and evaluating security policies that determine how access is granted to and delegated among particular users. Various access control models have been published [2]. Some are defined in terms of well known abstraction of subjects, objects and access rights and some in terms of roles, permissions and users. Some models are formulated in matrix and rules, some in graphs, while other in logic.

Discretionary Access Control model has long been a widely accepted and used model in the real world secure systems. It govern the access of subjects to the objects on the basis of the subject's identity and of authorizations that specify, for each subject and object in the system, the access rights (e.g., read, write) the subject is allowed on the object. Objects are the passive entities storing information, such as patient health data files, records, fields in records, etc. Subjects are active entities that access the objects, such as doctors, nurses, general practitioners, etc. Each request of a user to access an object is checked against the specified authorizations. If there exists an authorization stating that the user can access the object in the specific access right, the access is granted, otherwise it is denied. Discretionary access control represents a flexible way to enforce different protection requirements, especially cooperative yet autonomous requirements. In [9], Varadharajan et al developed an access control model for mental health application.

On the other hand, *logic based* approaches have been developed by many researchers recently for the purpose of formalizing authorization specifications and evaluations [1,4,5]. The advantage of this methodology is to separate policies from implementation mechanisms, give policies precise semantics, and provide a unified framework that can support multiple policies. Logic based method usually has strong expressive and reasoning power which is particularly useful in supporting a complex set of e-consent instructions with nested inclusions and exclusions conditions to access health data.

2.2 Role-Based Access Control

Another paradigm of access control models is *Role Based Access Control*(RBAC) [7]. The central notion of RBAC is that accesses are associated with roles and users are assigned to appropriate roles thereby acquiring accesses.

Classic access control is based on the individual (subject) accessing a resource (object).

subjects → objects

In many situations, privileges are associated with posts other than individuals. Individuals get their privileges because their posts or positions in the organization. In other words, whoever get the post would get the privileges of the post. When people leave the organization or change the positions, their privileges will be revoked or changed, too. This happens in many organizations from the viewpoint of organization administration. For example, a doctor in a hospital can access the patients' information in the hospital. If the doctor leaves the hospital, he/she usually lose the capability to access the patients' information, too. If the number of subjects and objects is large, individual access control becomes difficult. Each individual needs to be assigned each access right when they get a position in the organization and revoked each access right if the person changes the post or leaves the organization. When privileges are indeed assigned to roles other than individual subjects, role-based access control can greatly simplify the administration work.

In role-based access control, roles are placed between the user and the resource and subjects get their access rights indirectly by assigning access rights to roles and roles to subjects. Roles describe rights, duties and tasks that people have to perform. When people leave or change roles, only the mapping from subjects to roles need to be revoked or changed. On the other hand, if the duties of the roles change, only the mapping from roles to objects need to be changed. Roles provide a more abstract viewpoint on access control.

subjects → roles → objects

The concept of role also applies to the provision of patient data in health care contexts. Some consents may be given by patients in relation to roles ("yes, I consent to have a pathology test done on those samples"). Multiple individuals may perform particular roles at different times, e.g. because of the need for shift-work in intensive-care.

Roles can be organized into hierarchies so that consents can be inherited, which could greatly reduce the amount of explicit consent specification. Roles can also be delegated. One entity may act on behalf of another. Attorney is an example to this relationship. Other examples include guardians of minors and of people who are not psychologically capable of managing their own affairs. Roles are created according to the job functions in an organization and users are assigned roles based on their qualifications, experience or responsibilities. In [8], Thomsen presented a role-based application design and enforcement in a health care environment.

3 Electronic Patient Records

The patient record is an account of a patient's health and disease after he or she has sought medical help. The record contains findings, considerations, test results and treatment information related to the disease process. The development of an electronic patient record comes from the increasing demand for well-structured and accessible patient data on one hand, and fast developments in computer technology on the other hand. Computer technology has the advantage to improve legibility, accessibility, and structured information management. Electronic patient records may be available to the clinician at any point where electronic access is provided to the records, and they allow simultaneous access by several clinicians at different sites, whereas paper records have to be physically present at the point of use. Data retrieval and update of electronic patient record is easier than from paper, since electronic records are physically more accessible to their users than paper records. It is also more convenient to integrate the content of electronic records for audit and analysis purposes. Efficient information management is critical to the practice of medicine. The existence of an up-to-date, complete database of the medical records and dependable patient data not only enhances the quality of medical care but also improves the teaching and continuing medical education and research.

The following is an example of electronic patient record format.

A . Essential Personal and Contact Details
  A1 . Name—surname, first name
  A2 . Address—street, suburb, state, postcode
  A3 . Date of birth
  A4 . Phone number—home and/or work and/or mobile
  A5 . Payment method—private fee, bulk billing, or 3rd party fee
B . Optional Personal and Contact Details
  B1 . Title (Miss, Ms, Mrs, Mr or Dr)
  B2 . Alias or preferred name
  B3 . Email address
  B4 . Fax number
  B5 . Occupation
  B6 . Gender (male, female or unknown)
  B7 . Marital status (divorced, married,...)
  B8 . Ethnicity (African, Aboriginal, etc)
  B9 . Country of birth
  B10 . Next of kin—name, address and phone number
  B11 . Employer—name, address and phone number
  B12 . Family members—name and address
C . Clinical Related Personal Details
  C1 . Status (regular, casual, transferred or visitor patient)
  C2 . Provider—name of doctor examining the patient
  C3 . Location of provider—(medical center, specialist center, ...)
  C4 . Precedures/treatment code (for billing purposes)
  C5 . Pathology and radiology results(in/out)—in or not in
  C6 . Visit history—all visits, to this location of provider
  C7 . Next appointment—name of preferred doctor, date and time
D . Health Details
  D1 . Consultation
    D11 . Date of consultation
    D12 . Current health problems and complains
    D13 . Examination and notes taking
    D14 . Management plan
    D15 . Pathology and radiology requests
    D16 . Referrals
    D17 . Follow up and recall arrangements
  D2 . Medical History (all consultation information)
  D3 . Sensitive Medical History
    D31 . STD(HIV/AIDS) (all consultation information)

D32 . Gynaecological Conditions (all consultation information)
D4 . Other History
D41 . Smoking and alcohol history
D42 . Employment details
D43 . Operation history
D44 . Family history
D5 . Allergies and Sensitivities
D6 . Immunization Record
D7 . Medication History
D8 . Pathology and Radiology Results

4 A System Architecture for E-consent

In this section, we present a system architecture that enables various requirements for e-consent to be developed.

Dimensions of Consent

Consents may involve subjects to whom the consents are given, information (data) to be protected, access rights allowed or prohibited on the information, and subjects who issue the consent.

Subjects: Roles and Individuals

In the context of e-consent for health care, the consent may be assigned on the basis of an individual's identity or a clinical role within an organization. Some consents may be given by patients in relation to identified individuals, e.g., "yes, I consent to send those details to Dr Smith". In other circumstances, the consent is for a role, such as, "yes, I consent to have a pathology test done on those samples". Roles can be organised into different hierarchies so that the consent can be inherited. The supervision role hierarchy, for example, is based on the organization management hierarchy; the isa role hierarchy is based on generalization; and the activity hierarchy is based on aggregation. For example, Figure 1 is a possible role hierarchy based on aggregation.

Data

To allow consent inheritance along the data dimension, data could be organised into hierarchies. For example, the electronic patient record given in Section 3 can be organised into the the hierarchy shown in Figure 2. A consent to read health details (D) means a consent to read all data associated with the episodes of care (D1,D2,D3 and D4), and all the data captured in various events of care.(D5, D6, D7 and D8).

Access Rights

Usual access rights such as read, write, and update apply to the patient data. Access rights can also be organised into hierarchies to allow inheritance along

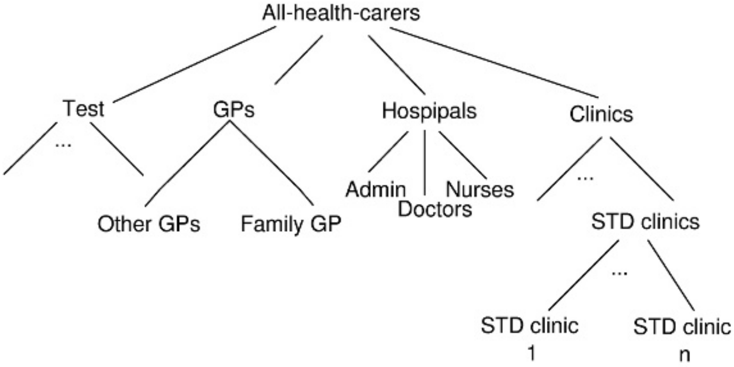

Fig. 1. A role hierachy.

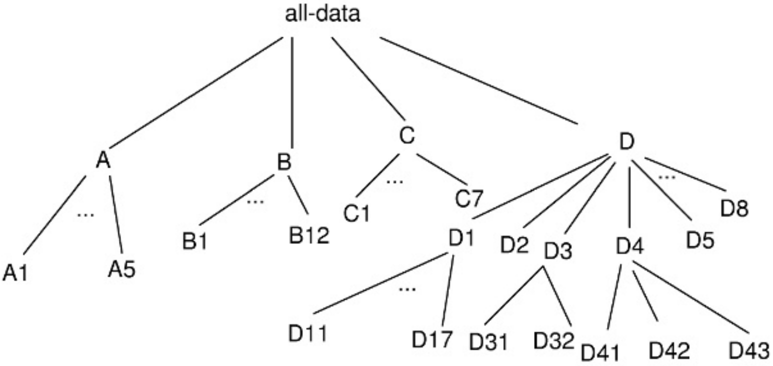

Fig. 2. An object hierachy.

this dimension. A consent to updating for example, may also imply a consent to reading and writing.

Consent and Denial

Both consents and denials are needed in a flexible e-consent system. Denials are useful when patients want to express explicitly that some disclosure is forbidden. In some e-consent models where the presumption is consent, the explicit denial will override the default consent. For example, some countries establish default consent in relation to organ donation that are overridden by an explicit denial. In other e-consent models where denials are established as default, the explicit denial could prevent further assigning of consent accidentally or intentionally.

Role Delegation and Consent Delegation

In some circumstances, a patient may wish to delegate the capability to grant consent to nominated representatives or medical practitioners, who may further wish to delegate the power to consent to other health professionals. This is usually done for flexibility, cooperation, and convenience of the carer. Note that the patient still keeps the capability to issue the consent after the delegation.

In role delegation, a patient delegates all of his/her privileges of consent to the nominated entity. However, in some situations, a patient may only wish to partially delegate his/her capability of creating consent to others, e.g., delegate the consent on the information regrading the treatment details but not the personal details.

Conflict Resolution

Because of role delegation and consent delegation, multiple grantors may exist for a specific consent and hence conflicts may arise. For example, a patient may wish to deny all information relating to HIV to be open to his/her immediate family, but his/her family GP, to whom he/she has delegated the privilege of granting consent, may want to disclose the information to the patient's immediate family. In this case, the patient's immediate family may receive two conflicting authorizations, consent and denial. A proper conflict resolution policy is then needed.

Control of Delegation

As we said before, to achieve high quality of treatment, a patient may wish to give the carer more flexibility by delegating them the required rights. On the other hand, to protect his/her privacy, a patient may not wish to lose control on his/her health data. One solution to this problem is to give a patient higher priority than his/her delegate, so that once conflict occurs, the patient's consent grant will override the other's. In addition, proper constraints on delegation are needed to avoid undesirable situations. For example, a doctor receiving the right to grant for a patient's health data should not be able to deny the patient to read his/her own health data by issuing the patient a negative authorization.

Consent Inheritance and Exceptions

As in many information systems, allowing consent inheritance would greatly reduce the amount of consents that need to be explicitly specified. When consents can be inherited, it is important to support exceptions to avoid undesire inheritance. For example, a consent to the health care professional means a consent to every health carer. A consent to the health care professional followed by a denial to a particular GP means that the GP could not be exposed to the patient's information.

Consent Capability

Usually a patient has the capability to grant consent on his/her own health data. However, in some circumstances a person is physically or legally incapable of giving consent; for instance children or minors, persons in coma, seriously incapacitated or frail people. These cases are usually subject to a variety of health-specific laws, e.g. laws relating to guardianship.

Purposes and Contexts

Sometimes, a consent is assigned on the basis of specific use of information. Common purposes include treatment, cooperation, training, teaching, notification (requests by persons closely associated with the person concerned, such as guardians, partners and immediate family), research, and getting advice from specialists.

Sometimes, consent is assigned based on the current context. A doctor may not be allowed to read the patient's health data in a normal situation, but may be allowed to do so in an emergency situation.

Implicit Consents and Inference Rules

The inherited consents belong to implicit consents. In general, rule based consent specification allows for implicit authorizations to be derived from the explicit authorization set through reasoning. Hence this can greatly reduce the size of explicit authorization set. A system architecture for e-consent is shown in Figure 3.

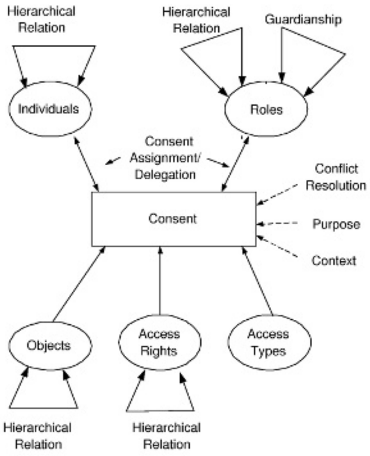

Fig. 3. A System Architecture for e-Consent.

5 Supporting E-consent Requirement

In this section, we show how to use the Delegatable Authorization Program (DAP) presented in [6] to support e-consent requirement.

5.1 DAP

Recall that our language $\mathcal{L}$ is a many-sorted first order language, with four disjoint *sorts* for subject, object, access right and authorization type respectively. In the constant set of authorization types $T = \{-, +, *\}$, $-$ means *negative*, $+$ means *positive*, and $*$ means *delegatable*. A negative authorization specifies the access that must be forbidden, while a positive authorization specifies the access that must be granted. A delegatable authorization specifies the access that must be delegated as well as granted. That is, $*$ means $+$ together with administrative privilege on the access. The partial orders $<_S, <_O, <_A$ represent inheritance hierarchies of subjects, objects and access rights respectively. We suppose $\sharp$ in S denotes the security administrator, and it is not comparable to any subjects in S w.r.t. $<_S$.

The predicate set consists of a set of ordinary predicates defined by users, and one built-in predicate symbol for delegatable authorization, *grant*. *grant* is a 5-term predicate symbol with type $S \times O \times T \times A \times S$. The first argument is the *grantee*, the second argument is the *object*, the third argument is the *authorization type*, the fourth argument is the *access right*, and the fifth argument is the *grantor* of this authorization. Intuitively, $grant(s, o, t, a, g)$ means s is granted by g the access right a on object o with authorization type t.

A *term* is either a variable or a constant. Note that we prohibit function symbols in our language. An *atom* is a construct of the form $p(t_1, ..., t_n)$, where p is a predicate of arity n in P and $t_1, ..., t_n$ are terms. A *literal* is either an atom p or the negation of the atom $\neg p$, where the negation sign $\neg$ represents classical negation. A *rule* r is a statement of the form:

$b_0 \leftarrow b_1, ..., b_k, not\, b_{k+1}, ..., not\, b_m, m >= 0$

where $b_0, b_1, ..., b_m$ are literals, and *not* is the negation as failure symbol. The b_0 is the *head* of r, while the conjunction of $b_1, ..., b_k, not\, b_{k+1}, ..., not\, b_m$ is the *body* of r. Obviously, the body of r could be empty. A *Delegatable Authorization Program*, DAP, consists of a finite set of rules.

Our examples in this section are mainly based on the electronic patient record given in Section 3, its hierarchical structure in Figure 2 and subject hierarchical structure in Figure 1.

5.2 Specifying and Evaluating E-consent Requirement

1. Consent and Denial

We use the authorization predicate *grant* to define consent and denial in the context of health care.

Definition 1. *A subject(individual/role) s consents (denies) another subject s' to exercise access right a on patient data o, which is defined by* $grant(s', o, +, a, p, s)$ $(grant(s', o, -, a, p, s))$.

For example, $grant(Bob, health\text{-}data, +/-, read, Alice)$ means that Alice consents/denies Bob to read her health-data.

By using DAP, it is easy to express the presumed consent requirement. For example, if a patient Bob wants to express that the access to his health data is presumed for anybody unless a explicit denial exists, he can use the following rule.

$grant(\_s, all\text{-}data, +, \_a, Bob) \leftarrow not\, grant(\_s, all\text{-}data, -, \_a, Bob)$

Intuitively, the rule says, $grant(\_s, all\text{-}data, +, \_a, Bob)$ is true if $grant(\_s, all\text{-}data, -, \_a, Bob)$ is not known to be true. Obviously, this consent rule is in favor of the convenience of accessing the health data.

It is also easy to express the general consent requirement. Suppose Bob wants to give a general consent to all the health carers for reading his health data, which can be represented by the following rule.

$grant(all\text{-}health\text{-}carer, all\text{-}data, +, read, Bob) \leftarrow$

If Bob only wish to disclose his health details (D) to all health carers for reading, then the following rule can express this.

$grant(all\text{-}health\text{-}carer, D, +, read, Bob) \leftarrow$

If Bob wants to give a general denial to all the health carers for all the access rights on his data, he can use the following rule to express.

$grant(all\text{-}health\text{-}carer, all\text{-}data, -, \_a, Bob) \leftarrow$

The following rule expresses that Bob give a general denial to a doctor John for all the access rights on his data.

$grant(John, all\text{-}data, -, \_a, Bob) \leftarrow$

2. Consent Delegation and Role Delegation

We first use authorization predicate *grant* to define consent delegation.

Definition 2. *A subject(individual/role) s delegates another subject s' the privilege to grant consent for access right a on patient data o, which is defined by* $grant(s', o, *, a, s)$.

For example, $grant(familyGP, health\text{-}data, *, read, Bob)$ means that Bob delegates his family GP the right to further disclose his health data for reading. Please note that, in our formulation, $*$ means $+$ plus the right to grant. This means, in this example, the family GP can read the data as well as disclose the data.

The role delegation from $\_s$ to $\_s'$ can be expressed by the following rule, which means that $\_s$ will delegate any of his/her rights on any object to $\_s'$.

$$grant(\_s', \_o, *, \_a, \_s) \leftarrow grant(\_s, \_o, *, \_a, \_g)$$

It is also easy to express the presumed delegation requirement. For example, if Bob wishes to delegate the administrative privilege on his health data to any body unless the explicit denial or consent exists, then the following rule can express this.

$$\begin{aligned} grant(\_s, \mathit{all\text{-}data}, *, \_a, Bob) \leftarrow\ & not\, grant(\_s, \mathit{all\text{-}data}, -, \_a, Bob), \\ & not\, grant(\_s, \mathit{all\text{-}data}, +, \_a, Bob) \end{aligned}$$

Intuitively, the rule says, $grant(\_s, \mathit{all\text{-}data}, *, \_a, Bob)$ is true if $grant(\_s, \mathit{all\text{-}data}, -, \_a, Bob)$ and $grant(\_s, \mathit{all\text{-}data}, +, \_a, Bob)$ are not known to be true (negation as failure). Please note that $grant(\_s, \mathit{all\text{-}data}, +, \_a, Bob)$ means a subject can exercise $\_a$ on all-data, but cannot further grant $\_a$ to other subjects.

If Bob wishes to deny a doctor John to access his data unless a explicit consent or delegation is granted, the following rule can express this.

$$\begin{aligned} grant(John, \mathit{all\text{-}data}, -, \_a, Bob) \leftarrow\ & not\, grant(John, \mathit{all\text{-}data}, +, \_a, Bob), \\ & not\, grant(John, \mathit{all\text{-}data}, *, \_a, Bob) \end{aligned}$$

3. Capability to Give Consent

To denote the capability to give consent, we define a guardian relation first. We introduce a new predicate, $guardian(s, s')$ with type $S \times S$, which means that s is a guardian of s'. For usual patients, they are their own guardians. For patients who are physically or legally incapable of giving consent, their guardians are different persons (subject to law). Let $own(s, o)$, with type $S \times O$, represent that s is the owner of the data o. Patients are considered to be the owners of their own health data.

Next, let the system administrator $\sharp$ delegate all the access rights on health data to the guardians of the patients.

$$grant(\_s, \_o, *, \_a, \sharp) \leftarrow own(\_s', \_o), guardian(\_s, \_s')$$

Hence, at the beginning, only guardians of patients can give consent to patients' data. However, through consent delegation or role delegation, other persons may receive capability to give consent, too.

4. Conflict Resolution

Let us have a look again at the conflict resolution policy proposed in [6] and see how it works here. First we solve the conflicts in terms of the delegation relations and give higher priorities to predecessors. In the e-consent context, a patient certainly wish to hold higher priorities than his/her delegates, so that whenever his/her consent instructions are conflicting with other's, his/her instructions will win. This is true for other delegators, too. In fact, to achieve high quality of treatment, a patient may wish to give the carer as much flexibility as possible. On the other hand, to protect his/her privacy, a patient may not wish to lose control on his/her health data. Giving a patient or, in general, a delegator higher priority than his/her delegate would best suit this situation. For example, a patient Bob may wish to deny all information relating to HIV to be open to his immediate family, but his family GP, to whom he has delegated the privilege of granting consent, may want to disclose the information to his immediate family. In this case, the patient's authorization will override his family GP's and therefore his immediate family could not access his HIV related information.

When two conflicting authorizations have the same grantors, the hierarchies of subjects, objects, and access rights are considered and the more-specific take precedence principle will be used. This can support the exceptions in consent inheritance. For example, a patient provides a general consent to a health care professional on the health data, but specifically precludes the disclosure of information about an STD condition, gynaecological procedure; or disclosure to their family GP. By using the more specific-take- precedence policy, the general consent will be overridden by the more specific denials.

If all the above policies don't apply, a patient can simply select denial-take-precedence based method, which is in favor of privacy; or consent-take-precedence based method, which in favor of treatment convenience.

Example 1. For more complex model where qualifications are nested, see the following example based on [3]. A patient Alice delegates to all information to all health carers; but within that denies all information relating to HIV (D3) and consents to information relating to HIV to STD clinics; and finally denies all information to a specific STD clinic-1 (where her mother works). The following rules plus our conflict resolution policy can achieve these requirements.

(1) $grant(all\text{-}health\text{-}carer, all\text{-}data, *, \_a, Alice) \leftarrow$

(2) $grant(all\text{-}health\text{-}carer, D3, -, \_a, Alice) \leftarrow$

(3) $grant(STD\text{-}clinics, D3, +, \_a, Alice) \leftarrow$

(4) $grant(STD\text{-}clinic\text{-}1, D3, -, \_a, Alice) \leftarrow$

Let us have a closer look on this. The first rule is a general rule which will propagate downward the subject and object hierarchies defined by Figure 1 and Figure 2, since consent inheritance in both dimensions is supported. therefore all health carers can access D3 which conflicts with the second rule. As Alice is the grantor of all the rules, the more specific-take-precedence principle will be used here; hence the inherited consent instructions from the general rules will be overridden by the more specific consent rules. This means that the second rule will override the first rule(on D3). For the same reason, the fourth rule overrides the third which again overrides the second.

On the other hand, since Alice is the owner, her instructions can not be overridden by any other person's instructions. Suppose a doctor John issues the following rule:

$$(5)\ grant(STD\text{-}clinic\text{-}1, D3, +, \_a, John) \leftarrow$$

This rule conflicts with the fourth rule and will be overridden by the fourth rule since John receives his delegatble privilege on Alice's health data from Alice (through the propagation of rule (1) along the subject hierarchy). Therefore the above four rules meet Alice's requirements.

5. Delegation Control

Cyclic authorizations are prohibited in DAP, as they usually do not make much sense and may cause undesirable situation. Consider for example a patient delegating the "read" right on his/her health data to his/her family GP. It is thus meaningless that the GP grants back to the patient to read his/her data. Moreover, it is undesirable if the GP could deny the patient to read his/her own data. On the other hand, as we mentioned before, giving predecessors higher priorities than successors provides users further control on consent delegation.

6. Purposes and Contexts

We extend the language a little by adding another sort *other* to the language, which contains constants and variables other than subjects, objects, access rights and authorization types. For example, we can put different purposes and contexts in this sort.

We further introduce two predicates, *purpose* and *context*; they both with one argument in sort *other*. The following rule means that a patient Alice consents to all health professionals to exercise any access on her health data if their purposes are for *treatment*.

$$grant(all\text{-}health\text{-}carer, all\text{-}data, *, \_a, Alice) \leftarrow purpose(treatment)$$

Similarly, the following rule states that a patient Bob gives all health professionals the consent for any access to his health data if it is in an *emergency* situation.

$$grant(all\text{-}health\text{-}carer, all\text{-}data, *, \_a, Bob) \leftarrow context(emergency)$$

6 Conclusion

In this paper, we presented an authorization model for representing and evaluating e-consent requirement in a health care application. This model supports well controlled consent delegations, both explicit and implicit consents and denials, individual based or role based consent instructions, and consent inheritance and exception. It is shown that the Delegatable Authorization Program and its conflict resolution policy provide users a good framework to express complex e-consent requirements. In addition, an electronic patient record is discussed and a flexible system architecture for e-consent is also presented.

References

1. E. Bertino, F.buccafurri, E.Ferrari, and P.Rullo, A logical framework for reasoning on data access control policies. In *Proc. of the 12th IEEE Computer Society Foundations Workshop*, pp 175–189, IEEE Computer Society Press, 1999.
2. S. Castano, M. Fugini, G. Martella, and P. Samarati, *Database Security.* Addison-Wesley Publishing Company, 1995.
3. E. Coiera, Consumer consent in electronic health data exchange. *Report*, the University of New South Wales, Australia, 2001.
4. J. Crampton, G. Loizou, and G. O'Shea, A logic of access control. *The Computer Journal*, 44:54–66, 2001.
5. S. Jajodia, P. Samarati, and V.S. Subrahmanian, A logical language for expressing authorizations. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, pp 31–42, 1997.
6. C. Ruan,V. Varadharajan, and Y.Zhang, Logic-based reasoning on delegatable authorizations. In *Proc. of the 13th International Symposium on Methodologies for Intelligent Systems* pp 185–193, 2002.
7. R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and Charles E. Youman. Role based access control models. *IEEE Computer*, 29(2):38–47, February 1996.
8. D.J. Thomsen, Role-Based Application Design and Enforcement.*Database Security,IV: Status and Prospects*, Elsevier Science Publisher B.V., pp 151–169 1991.
9. V. Varadharajan and C. Calvelli, An access control model and its use in representing mental health application access policy. *IEEE Transaction on Knowledge and Data Engineering*, 8(1):81–95, 1996.