

Engineering a Policy-Based System for Federated Healthcare Databases

Rafae Bhatti, *Member, IEEE*, Arjmand Samuel, *Student Member, IEEE*, Mohamed Y. Eltabakh, Haseeb Amjad, *Student Member, IEEE*, and Arif Ghafoor, *Fellow, IEEE*

Abstract—Policy-based management for federated healthcare systems has recently gained increasing attention due to strict privacy and disclosure rules. Although the work on privacy languages and enforcement mechanisms, such as Hippocratic databases, has advanced our understanding of designing privacy-preserving policies for healthcare databases, the need to integrate these policies in a practical healthcare framework is becoming acute. Additionally, although most work in this area has been organization oriented, dealing with the exchange of information between healthcare organizations (such as referrals), the requirements for the emerging area of personal healthcare information management have so far not been adequately addressed. These shortcomings arise from the lack of a sophisticated policy specification language and enforcement architecture that can capture the requirement for 1) the integration of privacy and disclosure policies with well-known healthcare standards used in the industry in order to specify the precise requirements of a practical healthcare system and 2) the provision of ubiquitous healthcare services to patients using the same infrastructure that enables federated healthcare management for organizations. In this paper, we have designed a policy-based system to mitigate these concerns. First, we have designed our disclosure and privacy policies by using a requirements specification based on a set of use cases for the Clinical Document Architecture (CDA) standard proposed by the community. Second, we present a context-aware policy specification language, which allows encoding of CDA-based requirements use cases into privacy and disclosure policy rules. We have shown that our policy specification language is effective in terms of handling a variety of expressive constraints on CDA-encoded document contents. Our language enables specification of privacy-aware access control for federated healthcare information across organizational boundaries, whereas the use of contextual constraints allows the incorporation of user and environment context in the access control mechanism for personal healthcare information management. Moreover, the declarative syntax of the policy rules makes the policy adaptable to changes in privacy regulations or patient preferences. We also present an enforcement architecture for the federated healthcare framework proposed in this paper.

Index Terms—Federated database security, healthcare engineering, policy-based management, role-based access control.

1 INTRODUCTION

PRIVACY and security of sensitive healthcare data is becoming a growing concern in the healthcare industry. With the recent legislations mandating the creation of electronic healthcare records (EHRs) as part of a nationwide initiative for providing electronic healthcare services, it becomes imperative to ensure that the availability of medical data in electronic form adheres to the same levels of privacy and disclosure regulations as applicable to present-day paper-based records accessible only from the physician's office. This introduces several challenges above and beyond data integration and involves the development of mechanisms to specify data disclosure and privacy policies that can be enforced across a network of interconnected healthcare service providers. This network of healthcare providers gives rise to a distributed clinical database, which resembles a federated system, and the

problem that we address in this paper is the design of a policy-based system for federated healthcare databases.

The research community has been engaged in addressing certain aspects of this problem. One notable technique called *hippocratic database* (HDB) [3] allows applications to enforce disclosure policies on databases on arbitrary data elements in an automated fashion. HDB includes mechanisms for active enforcement (data retrieval according to disclosure policy of the institution and privacy preferences of the user), compliance auditing, (ensuring compliance with data disclosure regulations, company policies, and customer preferences), and Sovereign Information Integration (allowing two or more autonomous entities to compute queries across their databases while only revealing the results of the queries), among others. Each of these dimensions encompasses a variety of issues that need to be considered at different levels. Our focus in this paper is the issues related to active enforcement, specifically, specification and enforcement of disclosure and privacy policies in a federated healthcare database system.

The various privacy and security approaches for database systems (including active enforcement in HDB) have relied on well-known privacy practices and languages (such as Platform for Privacy Preferences (P3P) [31] and Enterprise Privacy Authorization Language (EPAL) [16]). Although the work on privacy practices, languages, and enforcement mechanisms has advanced our understanding

• R. Bhatti is with IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120. E-mail: rafae@ieee.org.

• A. Samuel, M.Y. Eltabakh, H. Amjad, and A. Ghafoor are with the School of Electrical and Computer Engineering, Purdue University, Electrical Engineering Building, 465 Northwestern Ave., West Lafayette, IN 47907-2035. E-mail: {amsamuel, hamjad}@purdue.edu, ghafoor@ecn.purdue.edu.

Manuscript received 1 Feb. 2006; revised 30 Dec. 2006; accepted 9 Apr. 2007; published online 18 Apr. 2007.

For information on obtaining reprints of this article, please send e-mail to tkde@computer.org, and reference IEEECS Log Number TKDE-0049-0206. Digital Object Identifier no. 10.1109/TKDE.2007.1050.

of designing privacy-aware policies, the need to integrate these policies in a practical healthcare framework is becoming acute. No prior work has been reported on the specification of privacy and disclosure policies in relation to the clinical document (CD) storage and retrieval standards in place today. Additionally, we view the healthcare initiative involving EHR to have an impact beyond the hospital boundaries within a federated network and actually serve as an enabling technology for advancing ubiquitous and personal healthcare information management [22]. Although the EHR initiative, as currently envisioned, is organization oriented (that is, it is designed to manage the patient records between hospitals, such as referrals), personal healthcare information management would add a patient-oriented dimension to it; that is, patients equipped with mobile devices (such as PDAs and cell phones) will be able to obtain services from anywhere based on their EHRs located anywhere in the federated healthcare system. It may allow patients to obtain prescriptions without visiting a doctor's office or get remote diagnoses from their family physician who is currently out of town without having to visit another physician. It may also allow requests from a physician in the vicinity to access a patient's records in a remote emergency situation.

Present privacy-aware mechanisms have adequately captured the requirement for organization-oriented healthcare information management, but they do not provide adequate support for personal healthcare information management (see related work in Section 2). Satisfying this latter requirement requires the specification and enforcement of expressive context-aware disclosure and privacy policies. The very fact that the clinical data is now going to be available outside its traditional boundaries raises several issues regarding the establishment of the legitimate "context" used to enforce the disclosure and privacy policies. The context has a twofold interpretation—the context of the user and the environmental context—and both kinds of contexts have relevance for policy design. For instance, the federated healthcare system should be able to provide an affiliated physician (*user context*) access to a patient's information from a certain location (*environment context*), and the policies related to disclosure and privacy rules should have a context-specific restriction to incorporate this requirement in the security mechanism. In such a scenario, the user context for the physician would be used to satisfy the privacy requirements of the patient, whereas the environmental context would have implications in the disclosure rules controlling the release of sensitive information. The two contexts may even be used together in a policy: A patient may routinely permit only a particular physician to access his/her records, but if the context indicates emergency, then the patient allows any physician handling the emergency to access the records.

1.1 Contributions and Organization

Based on the preceding discussion, our contributions in this paper are twofold. One, we highlight a missing connection in prior work on policies related to disclosure and privacy rules vis-à-vis the integration of the policies with well-known EHR standards used in the industry. In the absence of such a connection, it is difficult to specify

the precise requirements of a practical healthcare system. In this paper, we have used Clinical Document Architecture (CDA) [11] as a representative EHR standard and designed our policies based on a set of use cases for the CDA standard proposed by the community [6], [19]. We then show that our policy specification language is effective in terms of handling a variety of expressive constraints on the CDA-encoded EHRs.

Our second contribution is the use of an expressive context-aware specification language, which not only allows encoding disclosure and privacy rules by using a declarative predicate-based syntax in the policy but also allows the use of temporal and nontemporal contextual constraints to be specified in the policy. The specification of disclosure and privacy rules allows privacy-aware access control for federated EHRs across organizational boundaries, whereas the use of contextual constraints allows the incorporation of user and environment context in the access control mechanism for personal healthcare information management. Moreover, the declarative syntax of the policy rules makes the policy highly reconfigurable and adaptable to changes in privacy regulations or patient preferences. The design of our policy specification language builds upon the well-known Role-Based Access Control (RBAC) model [25] and augments it with necessary extensions to support access management in a federated system. Our policy is supported by a well-defined administration model [9] and management framework [7] for scalable maintenance. We also present an enforcement architecture for the federated healthcare framework proposed in this paper.

The remainder of the paper is organized as follows: Section 2 provides a more detailed overview of related work and highlights the particular merits of our approach with respect to the outlined challenges. Section 3 discusses the design considerations for implementing the federated healthcare system proposed in this paper. In this regard, we provide an overview of the federated database architecture, the CDA standard used for representing EHRs in our framework, and the access control policy rules for EHRs. Section 4 presents the details of the policy specification language and discusses its salient features for access management in a federated system. Section 5 discusses the use of the policy specification toward providing support for federated healthcare access management. It describes the effectiveness of our language specification based on the requirement specification extracted from the CDA standard as a set of use cases proposed by the community. Some example use cases are also described in this section. Section 6 presents the enforcement architecture of our framework. Section 7 discusses the implementation of the example policy designed in the paper on our current prototype. Section 8 concludes the paper.

2 RELATED WORK

Our framework is aimed at designing a policy-based database system that integrates the knowledge of healthcare requirement specification and context-aware policy specification language to provide comprehensive support for federated healthcare information management. There are some aspects of this approach that have been reported in

prior work, but none addresses all these aspects of federated healthcare information management that we have outlined above.

Policy-based approaches have recently been proposed for access management in information systems [7], [8] and have also been applied to relational databases through notions like HDB [2], [3], [4], [5]. The focus of many of the policy-based approaches for database [2], [3], [4], [5], [10], [18] has been on privacy-aware access control. HDB includes, among others, the mechanism for active enforcement, which deals with data retrieval according to the disclosure policy of the institution and the privacy preferences of the user expressed in a well-known privacy language (such as P3P [31] or EPAL [16]). Although HDB focuses on privacy preserving in relational database management systems (DBMSs), the work described in [18] extends the data sources to include Lightweight Directory Access Protocol (LDAP) and file system directories. The work in [10] presents a purpose-based access control model to represent the privacy policies in terms of the purpose for which the data is requested. They also give a method to determine the access purpose of a user, but they do not have a high-level language that can be used to express purpose-based privacy preferences.

Although HDB is the most prominent of these works, and it is designed to meet some of the requirements that we have highlighted in this paper, it still lacks some features such as 1) providing a flexible and expressive specification language with a well-defined administration model for scalable maintenance and 2) integrating these rules and policies in the current practical healthcare frameworks, for example, the CD storage and retrieval standards in place today. With respect to item 1, HDB does not include a policy specification language by itself but only allows matching privacy preferences of organizations and users expressed in an existing language. The resulting policy is stored in a relational table, and all queries that access user data are rewritten according to the stored policy by the active enforcement engine. Therefore, the expressiveness of the policy specification is inherently tied to the existing languages used by HDB, and there is no opportunity for scalable maintenance, since no policy administration model or management framework exists to support the policy. As concerns item 2, no work has been reported that can demonstrate how the policies in HDB can be integrated with current healthcare standards. The main reason is that those policies (P3P and EPAL) are designed primarily for the online e-commerce space, and no dedicated provisions exist in them to support specific healthcare use cases such as those designed for the CDA standard, which we consider in this work.

Along another dimension, earlier efforts have been reported on policy-based access control in traditional database systems [23], [27]. Some recent work has focused on healthcare information management [26], [30], where the emphasis is to design policy-based trust frameworks for distributed healthcare applications. Standards such as Extensible Access Control Markup Language (XACML) have been used for expressing context-aware access control policies for Web-based documents [21]. The work in [15]

presents a context-aware access control system, where the idea is to incorporate user and environment context in the access control, much like the above-cited requirement for context-sensitive protection of clinical records for personal healthcare information management. These approaches are one step forward in meeting our stated requirements but are directed at traditional (as opposed to federated) database systems. Additionally, they do not capture the advanced context-based requirements for personal healthcare information management because they are based on very simple context abstractions (such as saying "physician d1 can access *all* records of patient p1 when *logged in from* ABC hospital network *on* Wednesday *between* 9 a.m. and 5 p.m.), but the federated healthcare system demands more fine-grained and expressive contextual constraints (such as saying "physician d1 can access a record of *type X* of patient p1 when *accessing from or near* ABC hospital *on the first* Wednesday of the *first* month of *every* quarter *between* 9 a.m. and 5 p.m.). The distinction here is that the former authorization is not fine grained enough to restrict the physician's access to a particular subset of records (based on the specialization of the physician and/or the privacy policy of the patient) and is not expressive enough to encode the "*accessing from or near*" location condition and "*on the first day of the first month of every quarter*" time condition. The location conditions are more relevant in the personal healthcare scenario, where either the disclosure policy of the hospital or the privacy policy of the patient may be location sensitive. The time-based conditions become more relevant in organizational data sharing, and they let hospitals allow external agencies such as insurance staff or auditors to access healthcare data under stricter schedule-driven temporal controls to mitigate the risk of unnecessary exposure of sensitive information and, at the same time, achieve regulatory compliance.

This discussion indicates the limitations of existing approaches with regard to the specification and enforcement of policies related to disclosure and privacy rules for 1) organizational and 2) personal healthcare information management. The former is focused more on relatively stable rules such as legislation and basic user preferences, whereas the latter concerns itself more with the provision of services against dynamic context-aware policy rules such as fine-grained and expressive constraints described above. Additionally, the need for integrating these policies in a practical healthcare framework and analyzing the coverage of the policy in terms of real-world requirements also remains unaddressed. To the best of our knowledge, ours is the first work that attempts to address these issues in federated healthcare information management.

3 DESIGN CONSIDERATIONS

3.1 Federated Database Architecture

The federated healthcare database system proposed in this paper is based on the well-established Federated Database System (FDBS) architecture described in literature [12], [13], [14], [24], [27], [29]. To summarize, such an architecture consists of a multilevel schema, consisting of a local schema, a component schema, an export schema, and the overall

federated schema. This schema architecture allows the resulting database system to support distribution, heterogeneity, and autonomy, which are the three cardinal requirements of an FDBS. A local schema is the conceptual schema of a component database system expressed in the native data model of the component DBMS and, hence, different local schemas may be expressed in different data models. A component schema is derived by translating local schemas into a data model called the canonical or common data model (CDM) of the FDBS. Two reasons for defining component schemas in a CDM are 1) they describe the divergent local schemas using a single representation and 2) semantics that are missing in a local schema can be added to its component schema. Thus, they facilitate negotiation and integration tasks performed when developing a tightly coupled FDBS. Similarly, they facilitate the negotiation and specification of views and multidatabase queries in a loosely coupled FDBS. The process of schema translation from a local schema to a component schema generates the mappings between component schema objects and local schema objects. An export schema represents a subset of a component schema that is available to the FDBS. The purpose of defining export schemas is to facilitate the control and management of association autonomy. A federated schema is an integration of multiple export schemas. A federated schema also includes the information on data distribution that is generated when integrating export schemas.

This architecture forms the baseline for the design and operation of an FDBS. In this paper, our design of a federated healthcare database system assumes that the distributed databases form a tightly coupled federation of multiple databases. Of particular relevance to us in this architecture are the concepts of CDM and the federated schema. We will present in Section 4 the policy definitions that act as the CDM throughout the healthcare federation and will be used by each participating site to encode their export schemas. The integration of these schemas will then constitute the federated schema. The policy framework that we have designed can be adapted to work with any implementation of FDBS that abides by the FDBS architecture. Notable among the implementations reported in the literature are the Mermaid by Templeton et al. [27], Interoperable Relational and Object-Oriented Databases (IRO-DB) [13], distributed search component (Disco) [29], and MIRO-Web [12].

3.2 EHR Format

For representing the EHR in our framework, we choose to adopt the CDA [11] standard, which is an American National Standards Institute (ANSI)-certified standard from the Health Level Seven (HL7) organization. HL7 is an international community of healthcare subject matter experts and information scientists collaborating to create standards for the exchange, management, and integration of electronic healthcare information. It is one of several ANSI-accredited Standards Developing Organizations (SDOs) operating in the healthcare arena. It explicitly deals with the clinical and administrative domain. HL7 standards are used in 90 percent of US healthcare facilities. The

organization has affiliates in 20 countries, and in some of those, HL7 messaging has a legal status.

CDA defines an XML architecture for exchange of CDs. CDA specifies the syntax and supplies a framework for specifying the full semantics of a CD. A CDA can contain any type of clinical content. Typical types of CDA documents would be a Discharge Summary (DS), Imaging Report, Admission and Physical, Pathology Report, and so on. CDA uses XML, although it allows for a non-XML body (PDF, Word, JPEG, and so on) for simple implementations. CDA documents can be displayed using XML-aware Web browsers or wireless applications such as cell phones. These features of CDA and its widespread use in the healthcare industry¹ makes our choice of using CDA as a reference standard for EHR a pragmatic one, which facilitates our outlined goals. It may be stated that many EHR vendors already have the capability to produce CDA-compliant documents [11].

CDA works by using the concept of “incremental” semantic interoperability to allow multiple organizations to exchange EHRs through a more or less automated process. What this means is that there is a range of complexity allowed within the specification, and users must set their own level of compliance. The minimal CDA is a small number of XML-encoded metadata fields (such as provider name, document type, document identifier, and so on) and a body, which can be any commonly used MIME type such as .pdf or .doc (Microsoft Word) or even a scanned image file. All documents encoded using even the minimal compliance level would be equally readable at the place of healthcare provision. In our framework, we assume CDA-encoded EHRs at the minimum compliance level or higher, and we design our disclosure and privacy policies based on a set of use cases for the CDA standard proposed by the community [6], [19].

3.3 Access Control Policy

Having discussed the format for the EHR, we now discuss the policies for the disclosure and privacy policies for them, but first, we would like to say a few words about our use of the terms disclosure and privacy policies as two components of access control policy for the healthcare database.

As the name suggests, we view a disclosure policy as the policy of an organization specifying the rules governing the disclosure of patient information to other entities. Our use of the term privacy policy refers to a policy where the privacy preferences of the patient are specified. We realize that the terms privacy policy and access control policy have often been used interchangeably, but we view privacy policy as a broader statement of preferences which contribute toward the design of the access control policy. Similarly, disclosure policy also has a broad goal and provides the guidelines for what access control needs to be enforced on the release of patient information. In our work, the access control policy is actually an enforceable subset of privacy and disclosure rules written in a policy specification language that can be translated into program code.

1. As an indication of its usage, the flagship implementation of CDA in the country is the Mayo Clinic, which is producing thousands of CDAs every week and is expected to reach 50,000 notes/week in full production [11].

Although the CDA format allows for a rich document markup useful for processing medical records, the key piece of metadata that we will utilize in the design of our policies will be the document type, which represents the type of EHR being requested. Keeping our goals for an organizational and personal standpoint for healthcare information management in view, we will now provide an overview of some context-aware requirements that need to be specified in the access control policy for the EHRs.

Consider a disclosure policy for an EHR of type DS, which may have a context-specific restriction such that the data is allowed to be viewed by any physician only in case of an emergency situation. A very straightforward definition of an emergency context can be based on location or the proximity of the device used to issue the request; that is, if a physician issues a request from near an emergency ward, then it is an emergency situation. More sophisticated techniques to capture context is an active research area [17]. In this situation, the location of the user provides a context, which has a bearing on the access control decision. Such a request is routinely met and satisfied in the case of paper-based records, where a physician physically retrieves a record while inside the emergency ward, but is much difficult to satisfy in the case of EHR because the proximity of a physician to the emergency ward needs to be 1) electronically determined and 2) incorporated in the access control decision. More significantly, the resource provisioning session needs to be disengaged when the proximity condition is no longer satisfied (which will be the equivalent of the physician returning the paper-based record in the cabinet before leaving the emergency ward). Similar context-aware restrictions may be placed on the privacy policy of patient, which may allow an individual (pharmacists, insurance examiners, and so forth) to access an EHR of a patient only within one month from the time of issue. An interesting point that arises here is the conflict between the privacy and disclosure policies in the event of an emergency access within one month of the issue of the discharge document. Situations like these are likely to arise in practical situations and should be handled by the access control policy.

Our policy specification language is designed to tie together the context-aware policy requirements with the industry standard CDA EHR format to specify and enforce disclosure and privacy policies over a broad range of scenarios, similar to the one described above, and also address the issue of policy conflicts. The scenarios are captured using the requirements use cases for the CDA standard. Our results indicate the effectiveness of the language in terms of being able to handle the most common use cases.

4 XML-BASED GENERALIZED TEMPORAL ROLE-BASED ACCESS CONTROL (X-GTRBAC) POLICY SPECIFICATION

This section describes the key features of X-GTRBAC, the XML-based policy specification language in our federated healthcare information management framework. Our specification language is an extension of the RBAC model, which is recognized for its support for simplified administration in large-scale systems [25]. The key idea in RBAC is that permissions are assigned to roles (as opposed to

users directly), and users are assigned to roles to access the associated permissions. This simplifies the administration of privileges because the permissions are assigned to a user based on their job functions (that is, roles): A change in the job function only means reassigning the user to a different role, and the permissions are appropriately reconfigured for the user. The use of various constraints on the assignment of permissions to roles and of users to roles, together with the use of role-specific constraints using the notion of role attributes, role hierarchy, and role-based separation of duty (SoD), makes it possible to exercise fine-grained context-aware access control in RBAC. In the following sections, we describe our language that is designed to accomplish this task.

4.1 Policy Language

X-GTRBAC language specification is captured through a context-free grammar called X-Grammar, which follows the same notion of terminals and nonterminals as in Backus-Naur Form (BNF) but supports the tagging notation of XML, which also allows expressing attributes within element tags. The use of attributes helps maintain compatibility with the XML schema syntax, which serves as the type definition model for our language. The nonterminals are expressed as `< !-“non_terminal_name”>` XML tags, and terminals as standard XML tags. The data types of the values of elements or attributes are indicated inside parenthesis `“()”` symbols. The complete syntax of the X-GTRBAC language specification appears in Appendix A, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>.

4.2 Policy Components

Table 1 summarizes the key components of the policy language and their respective properties. The associated X-Grammar representation is provided in the respective figures, as indicated in the table entry. We have intentionally kept the exposition of the policy language simplified and concise. It outlines the features relevant for our present purposes and cleanly separates all the policy components and their properties. For a detailed discussion on the policy language and its components, we refer the reader to [7].

4.3 Salient Features

We now discuss the features of the policy specification language. The key concept used in the assignment rules in the policy is the notion of credential types in X-GTRBAC. A `CredType` is included in the definition of an XML User Sheet (XUS) or XML Role Sheet (XRS; see Figs. 1 and 2). We use these credential types for composing rules for the privacy and disclosure policies. In particular,

1. A privacy policy is represented using the XML User to Role Assignment Sheet (XURAS; see Fig. 7). It allows definition of user-to-role assignment rules, which can be used to encode the preferences of a patient to permit a physician to view his/her records. We introduce a role called `PermittedPhysicianPx` for the physician who is permitted by a patient `x`. Thus, to access the EHR of a patient `x`, the physician supplies a credential in an XUS to the system, with appropriate attributes needed to satisfy the assignment policy for the role `PermittedPhysicianPx` defined in XURAS. These attributes might

TABLE 1
The Components of the Policy Specification Language and Their Properties

Component	Sub-Category	X-Grammar Syntax	Properties	Example
Credentials	Authenticating credentials	They are defined in an <i>XML User Sheet (XUS)</i> (See Figure 1)	They encode the information needed to authenticate a user (assign it a role). It is encoded in the form of qualified <i>attributes</i> (as opposed to identities)	Examples of credential attributes for user are <i>DLN</i> , <i>SSN</i> , etc.
	Authorization credentials	They are defined in an <i>XML Role Sheet (XRS)</i> (See Figure 2)	They encode the information needed to authorize a role (assign it a permission). It is encoded in the form of role attributes	Examples of credential attributes for role are <i>time of day</i> , <i>system load</i> , etc.
Constraints	Semantic constraints	They are defined in an <i>XML Separation of Duty Definition Sheet (XSoDDef)</i> (See Figure 3)	Semantic constraints include integrity constraints on roles, such as conflict of interest. These can be included in the role definition in XRS (See Figure 2)	Examples of SoD constraints are static SoD (SSD) (assignment time) and dynamic SoD (DSD) (activation time) constraints
	Temporal Contextual Constraints	They are defined in an <i>XML Temporal Constraint Definition Sheet (XTempConstDef)</i> (See Figure 4)	An assignment rule (See Assignment Rules) may refer to a temporal constraint expression to associate the constraint with the rule; this association is modular and flexible	Temporal constraints include constraints on periodicity, interval, and/or duration of user-to-role or permission-to-role assignments
	Non-Temporal Contextual Constraints	See Assignment Rules	Non-temporal constraints are encoded as predicates on credential attributes, involving a function, an expected return value, and a logical comparison (See Figure 6)	A predicate may involve a call to <i>getCredAttrValue</i> function to evaluate the supplied value of a user (role) credential attribute against the expected value
Assignment Rules	User-to-Role Assignment	User-to-role assignment rules are defined in <i>XML User-to-Role Assignment Sheet (XURAS)</i> (See Figure 7)	User-to-role assignment constraints may be based on credential attributes of a user as defined in XUS	A non-temporal or temporal constraint predicate may be included or referred inside an assignment rule to enforce fine-grained evaluation criterion for user-to-role and permission-to-role assignments
	Permission-to-Role Assignment	Permission-to-role assignment rules are defined in <i>XML Permission-to-Role Assignment Sheet (XPRAS)</i> (Analogous to XURAS)	Permission-to-role assignment constraints may be based on credential attributes of a role or the resource type as defined in XRS or XRTS respectively (See Figures 2 & 5)	

include the qualification (such as MD or MPhil) or affiliation (such as FRCS or FCPS) of the physician which the patient deems satisfactory to have confidence in the physician. The CDA document type *IndividualHealthCarePractitioner* is a standard format to encode physician credentials, which we adopt to define credential attributes in our framework.

2. A disclosure policy is represented using the XML *Permission to Role Assignment Sheet (XPRAS)*; defined similarly as *XURAS*). It allows definition of permission-to-role assignment rules, which can be used to encode the organizational rules to assign a *PermittedPhysicianPx* role the permission to access a particular kind of EHR of the patient *x*, where this permission is defined using the XML *Permission Sheet (XPS)*; see Fig. 5). In order to access the EHR of the patient *x*, a physician role (that is, *PermittedPhysicianPx*) must have a credential defined in an XRS present in the system, with appropriate attributes needed to satisfy the assignment policy for the requested type of CD. As indicated in Table 1, these attributes of a role are used to define context-aware access constraints (such as conditioning access to patient record based on the location of the physician role).
3. The specification of permissions is based not on the individual resources but on their types as per the CDA standard (for example, CD, DS, and so forth) in line with our stated objectives. A permission defined in XPS then comprises a specified operation on a given resource type. The actual resource type instances (that is, the EHRs) belonging to a given resource type are defined in the XML *Resource Type Sheet (XRTS)*; see Fig. 5). Thus, a role that is assigned

```

<!-- XML User Sheet> ::=
<XUS xus_id = (xs:id) >
  <User user_id = (xs:id) >
    [ <!-- Cred Type> ]
  </User>
</XUS>
[ <!-- Cred Type> ] ::=
<CredType cred_type_id=(xs:idref)
  cred_type_name= ( xs:name) >
  [ <!--Header> ]
  <!-- Credential Expression>
</CredType>
<!-- Credential Type Definitions > ::=
<XCredTypeDef xctd_id = (xs:id) >
  <CredTypeDef cred_type_id = (xs:id)
    cred_type_name=(xs:name) >
    <!-- Attribute List>
  </CredTypeDef >
</XCredTypeDef>

```

Fig. 1. Top-level X-Grammar for XUS. This includes definition of authenticating credential.

a permission defined on a given resource type will be given access to all resource type instances belonging to that resource type. The use of standardized definitions of resource types (that is, EHRs) promotes interoperability between multiple organizations, and the granularity of access (resource type versus resource) makes the system scalable, since individual resource identifiers need not be known in advance to compose assignment policies.

4. Both privacy and disclosure policies (that is, the XURAS and XPRAS) can include the definition of contextual constraints. Thus, for example, an `AssignConstraint` can be defined within a user-to-role or permission-to-role assignment policy, as discussed in Table 1. An `AssignConstraint` (see Fig. 7) can refer to a `PeriodicTimeExpression` that defines a temporal expression (see Fig. 4). It can also directly include a `LogicalExpression` that defines a nontemporal contextual constraint (such as a constraint defined on the location attribute of a role). This mechanism of constraint specification allows modular and flexible construction of constraint definitions (see Fig. 3). This includes using location-based condition such as “accessing from or near a hospital” (as opposed to only

```

<!-- XML Role Sheet> ::=
<XRS xrs_id = (xs:id) >
  <Role role_id = (xs:id)
    role_name = (xs:name) >
    [ <!-- Cred Type> ]
    [ <!--(En|Dis)abling Constraint> ]
    [ <!--[De]Activation Constraint> ]
    [ <!--Delegation Constraint> ]
    [ <JuniorRoleId> (xs:idref) </JuniorRoleId> ]
    [ <SeniorRoleId> (xs:idref) </SeniorRoleId> ]
    { <SSDRoleSetId> (xs:idref) </SSDRoleSetId> } *
    { <DSDRoleSetId> (xs:idref) </DSDRoleSetId> } *
    { <!--LinkedRoleId> } *
  </Role>
</XRS>

```

Fig. 2. Top-level X-Grammar for XRS. This includes definition of authorization credential.

```

<!-- Separation of Duty Definitions> ::=
<XSoDDef xsod_id = (xs:id) >
  <!--SSDRoleSets>
  <!--DSDRoleSets>
</XSoDDef>

```

Fig. 3. Top-level syntax of SoD constraint definition.

```

<!-- Definitions of Temporal Constraints> ::=
<XTempConstDef xtcd_id = (xs:id) >
  { <!--Interval Expression> } *
  { <!-- Periodic Time Expression> } *
  { <!-- Duration Expression> } *
</XTempConstDef>

```

Fig. 4. Top-level syntax of temporal constraint definition.

“logged in from a hospital network”) and time-based conditions such as “on the first Wednesday of the first month of every quarter between 9 a.m. and 5 p.m. (as opposed to only “on Wednesday between 9 a.m. and 5 p.m.”) to allow provision of location-sensitive and schedule-driven access control to sensitive information, as we alluded to in Section 2. We will discuss the formal syntax of constraints in Section 6 and will illustrate their use in our outlined set of requirements use cases.

One notable feature of the assignment rules deserves a mention. Our logical expression (LE) syntax allows multiple LEs to be combined together in an appropriate rule-combining mode using Boolean connectives. The modes supported by the specification language are AND (all rules must be true), OR (at least one rule must be true), and NOT (no rule must be true). Several levels of nesting are supported, each under a distinct mode, to allow a fine granularity of rule specification. An assignment condition is satisfied if all of its included LEs are satisfied according to the respective mode. The results of evaluating multiple

```

<!--XML Permission Sheet>::=
<XPS xps_id = (xs:id) >
  <Permission perm_id = (xs:id)
    [prop= (noprop|first_level|cascade)] >
    <Object res_type_id=(xs:idref) >
      { <!--Attribute > } *
    </Object>
    <Operation> (saml:Action) </Operation>
  </Permission>
</XPS>

<!-- XML Resource Type Definitions>::=
<XResTypeDef xrtcd_id = (xs:id) >
  <ResTypeDef res_type_id = (xs:id)
    res_type_name = (xs:name) >
    { <!--Attribute List> } *
  </ResTypeDef>
</XResTypeDef>

<!-- XML Resource Type Sheet>::=
<XRTS xrts_id = (xs:id) >
  <ResType res_type_id = (xs:idref)
    res_type_name = (xs:name) >
    { <!--Attribute> } *
  </ResType>
</XRTS>

```

Fig. 5. Top-level syntax of XPS, resource type definitions, and XRTS.

assignment conditions within an assignment constraint are combined similarly. Role assignment occurs as a consequence of an assignment constraint being satisfied.

We note that the “AND” mode essentially implements “deny overrides,” whereas the “OR” mode implements “permit overrides.” The NOT mode allows one to condition a role assignment based on negation. This is useful in instances when it is easier to express exclusion rather than inclusion as a criterion for membership in a role. Although negation is allowed in the body, it is not allowed in the consequence of a rule. This prevents contradictory rule sets from existing in the specification. This property is helpful when combining rules aimed at a given consequence, since one can always be sure that new rules will not clash with existing rules in the policy.

At the same time, we note that our policy supports two kinds of consequences for a rule: assignment and deassignment. Although the set of rules aimed at one kind of consequence (assignment or deassignment) may be conflict free, the fact that different sets of rules can result in different consequences (assignment or deassignment) implies that our access control language is nonmonotonic. This is a conscious design decision and is made, in particular, to support considerations like patient consent in privacy policies. For example, even though a patient may allow all certified physicians to access his/her EHRs, he/she may like to make an individual exception to the rule for a particular physician (say, a physician with the name X). In this case, a deassignment condition is necessary in the policy to revoke the permission assigned to physician X. Generally speaking, a deassignment rule is the same as revocation of existing permission, and no language can have both revocation and monotonicity at the same time. We consider revocation as an important means to fulfill privacy obligations and, therefore, we include it in the language.

Although we mentioned that the disclosure policies depend on the type (and not the individual identifier) of the resource, this is intended only as a convenience and not a limitation. The specification language allows resource access to be further based on attributes of individual resources within a resource type to provide support for individual resource level access, if required. The resource type instances defined in XRTS have a list of attributes that may be used in the assignment policy to restrict the assignment criterion to only those resource type instances that have the desired attribute values. Within the allowed set of resources, the finest granularity of access supported by the language is at the element level; that is, an access can be restricted to only the relevant elements of an XML-encoded EHR of the patient. This is equivalent to a tuple-level access in relational database systems.

A particularly relevant feature for federated systems is that the credential specification in our language allows credential federation through the use of interoperable protocols. The language currently allows one to specify the structure of the credentials by using security attributes defined as per the Security Assertions Markup Language (SAML) standard [20]. We employ appropriate translation mechanisms for SAML assertions to be used with the X-GTRBAC language syntax. Our framework also supports Single Sign On (SSO) through the use of digitally signed SAML statements that capture an authorization decision already issued by a domain corresponding to a user request.

4.4 Policy Composition

An overall X-GTRBAC policy is composed² from these individual policy components, as given in Fig. 8. The complete X-Grammar policy syntax is provided in Appendix A, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>.

5 POLICY-BASED HEALTHCARE INFORMATION MANAGEMENT

In this section, we discuss the use of our policy specification language for federated healthcare information management. The policy language comprises the definitions of the various policy components used to encode the disclosure and privacy policies of all federating organizations that are part of the FDBS architecture, as described in Section 3.1. The policy definitions therefore serve as the CDM for the federated healthcare database, and the policy documents are used to define an export schema, which collectively form the federated schema of the system.

5.1 Requirements Use Cases

As alluded to earlier in the paper, we focus on integrating real-world healthcare requirements within our policy specification language. For this purpose, we base the design of our policy on a set of use cases proposed by the community for the CDA standard [6], [19]. We specifically focus on the use cases involving the use of “layered constraints” within the CDA-encoded EHRs. The HL7 Template proposal [6], which is intended to provide a format for defining constraints against an HL7 specification, is a recent effort in this direction. The proposal allows for a template to be used to constrain the values of static assertions regarding an EHR. This includes constraints on allowable attribute values, comparison of attribute values, nesting of assertions, and logical evaluation of assertions. As the HL7 proposal suggests, the use of such a template will allow formulation of expressive constraints on the document contents, which we believe can be used in the design of disclosure and privacy policies.

Below, we present the set of use cases, based on which we design the policies used in our framework:

Actors: The creators and readers of EHRs (such as physicians and healthcare givers), patients associated with them who have access privileges, payers (insurance), and institutions (health maintenance organizations (HMOs), government bodies, and enforcers of legislations such as Health Insurance Portability and Accountability Act of 1996 (HIPAA)) who have been permitted to access EHRs.

Use Case 1: Access policy for an EHR must have granularity at the level of 1) the medical and administrative (such as address and phone number) data, 2) the category of medical record as defined per the `ClinicalDocument` type system in CDA, and 3) the type of requestor within the actor populations.

2. Policy composition as used here refers to combining multiple distinct components of a policy together into a single document. This should not be confused with composing a single policy from multiple potentially overlapping policies through formal analysis.


```

<!-- XML Predicate Function Definitions>::=
<XPredFuncDef xpfid_id = (xs:id) >
  <Function func_id = (xs:id)
    func_name=(xs:name) return_type=(xs:anyType)>
    <!--Parameter List>
  </Function>
</XPredFuncDef>

```

Fig. 6. Top-level syntax of predicate function definition.

Use Case 2: The portion of EHR retrieved by a permitted requestor depends upon the requestor and the privacy conditions defined on the EHR.

Use Case 3: The restrictions on accessing an EHR extend beyond the originating organization.

Use Case 4: A user needs to get through the protections in case of emergencies.

Use Case 5: A user may be denied access to certain sensitive and damaging diagnosis information.

5.2 Policy Design

We now outline the design of our policy with respect to the use cases described above. Toward this end, we provide a formal representation of the above-mentioned requirements use cases as predicates and their specification in the policy.

Based on the discussion in Section 4 about the features of the policy specification language, we know that the policy rules are expressed in the language as assignment constraints involving a set of temporal and nontemporal contextual predicates. Thus, our policy specification language is naturally amenable to express predicate-based requirements specification. Our analysis therefore begins by representation of requirements use cases as a set of predicates. Consequently, we show that our policy language is effective in terms of handling a variety of expressive constraints on the CDA-encoded EHRs that need to be defined in compliance with the HL7 specification.

In order to represent requirements use cases as a set of predicates, we introduce the formal notions of what we will call a *predicate expression* (PE; see Fig. 6). A PE in our framework may be used to capture the effect of evaluating a set of temporal or nontemporal contextual expressions. As discussed in Section 4, the X-GTRBAC framework uses two kinds of constraint expressions: a temporal constraint expression, which is represented as a *periodic time expression* (PTE; see Fig. 4), and a nontemporal constraint expression, which is represented as an LE (see Fig. 7). We first formally define the constraint expressions in X-GTRBAC as follows:

Definition 1 (Constraint Expression). A constraint expression $cr \in CR$ is defined to be one of the following types:

- A PTE represented by pairs $\langle [begin, end], P \rangle$. Here, P is a periodicity expression denoting an infinite set P of periodic time instants, and a $[begin, end]$ is a time interval I denoting the lower and upper bounds that are imposed on instants in P . Formally, P is defined as $P = \sum_{i=1}^n O_i.C_i \triangleright x.C_d$, where C_d, C_1, \dots, C_n are units of the calendar (such as year, month, and day), the C_i s represent the starting time of the event represented by P , C_d represents the duration of the event represented by P , and O_1, \dots, O_n are the time

```

<!-- XML User-Role Assignment Sheet::=
<XURAS xuras_id = (xs:id) >
  <URA ura_id=(xs:id) role_id=(xs:idref)>
  <AssignUsers>
  <AssignUser user_id=(xs:idref)>
  <AssignConstraint[op =(AND|OR|NOT|XOR)]>
  // opcode defaults to AND if none specified
  <AssignCondition cred_type_id=(xs:idref)
  [pt_expr_id=(xs:idref) | d_expr_id=(xs:idref)] >
  <LogicalExpr [op =(AND|OR|NOT)]>
  // opcode defaults to AND if none specified
  <!-- Predicate>+
  </LogicalExpr>
  </AssignCondition>
  </AssignConstraint>
  </AssignUser>
  </AssignUsers>
</URA>
</XURAS>

```

Fig. 7. Top-level syntax of user-to-role assignment policy.

occurrences such that $O_1 = \text{all}$, $O_i \in 2^N \cup \{\text{all}\}$, $C_i \sqsubseteq C_{i-1}$ for $i = 2, \dots, n$, $C_d = C_n$, and $x \in \text{IN}$.

- An LE using the usual \wedge and \vee operators on 3-tuples of the form $(y, \omega, \delta(p_1, \dots, p_n))$. Here, δ is a parameterized function, p_i and y are either constants (such as string or integers) or identifiers of users or roles, and $\omega \in \{=, \neq, \geq, \leq, \in\}$.

Example 1.

$$\text{PTE}_{\text{EX1}} = \langle P, [2005.Years, 2005.Years] \rangle,$$

$$P = \text{all.Years} + \{1, 4, 7, 10\}.Months$$

$$+ \{1\}.Weeks \triangleright 1.Weeks$$

represents a PTE that defines a periodic time P , which is a set of intervals starting at the same time instance as the first week of every quarter of every year and having a duration of one week. Additionally, the PTE is valid within the interval bounded by the start and end of year 2005. An XML representation of PTE_{EX1} using the syntax of our policy specification language appears in Fig. 9a.

Example 2. $\text{LE}_{\text{EX2}} = (\text{NewYork}, =, \text{hasCredAttrValue} (\text{PermittedPhysicianPx}, \text{location}))$ represents an LE that includes the evaluation of a predicate function hasCredAttrValue (hCAV) that compares the value of the

```

<!--Policy Definition> ::=
<Policy policy_id = (xs:id)
  policy_name = (xs:name) >
  <!-- XML Credential Type Definitions>
  <!-- XML Separation of Duty Definitions>
  <!-- XML Temporal Constraint Definitions>
  <!-- XML Resource Type Definitions>
  <!-- XML Predicate Function Definitions>
  <!-- XML Resource Type Sheet>
  <!-- XML User Sheet>
  <!-- XML Role Sheet>
  <!-- XML Permission Sheet>
  <!-- XML User-Role Assignment Sheet>
  <!-- XML Permission-Role Assignment Sheet>
</Policy>

```

Fig. 8. The overall X-GTRBAC policy.

<pre> <?xml version="1.0" encoding="UTF-8"?> <XTempConstDef xtcd_id="HCF_XTCD"> <IntervalExpr i_expr_id="Year2005"> <begin>1/1/2005</begin> <end>12/31/2005</end> </IntervalExpr> <DurationExpr d_expr_id="OneWeek"> <cal>Weeks</cal> <len>1</len> </DurationExpr> <PeriodicTimeExpr pt_expr_id="PTQuarterWeekOne" i_expr_id="Year2005" d_expr_id="OneWeek"> <StartTimeExpr> <Year>all</Year> <MonthSet> <Month>1</Month> <Month>4</Month> <Month>7</Month> <Month>10</Month> </MonthSet> <WeekSet> <Week>1</Week> </WeekSet> </StartTimeExpr> </PeriodicTimeExpr> </XTempConstDef> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <XPRAS xuras_id="HCF_XPRAS"> <PRA pra_id="ura PermittedPhysicianPBob"> role_id="PermittedPhysicianPBob"> <AssignPermissions> <AssignPermission perm_id="PBobCD"> <AssignConstraint> <AssignCondition cred_type_id= "CDAIndividualHealthCarePractitioner" pt_expr_id="PTQuarterWeekOne"> <LogicalExpr> <Predicate> <Operator>eq</Operator> <FuncID>fhCAV</FuncID> <ParamName>location</ParamName> <RetVal>NewYork</RetVal> </Predicate> </LogicalExpr> </AssignCondition> </AssignConstraint> </AssignPermission> </AssignPermissions> </PRA> </XPRAS> </pre>
(a)	(b)

Fig. 9. (a) This temporal constraint definition includes a periodic time expression (PTE), which states that the access is allowed beginning the first week of every quarter of year 2005. Note that the duration expression and/or the interval expression are referenced inside a PTE. (b) This is a permission assignment policy for a `PermittedPhysicianPBob` role, which includes an LE defined on the credential attribute of the role and also references the PTE defined in (a). It states that any user (any is a keyword) can be assigned the permission to access a `ClinicalDocument` belonging to patient Bob if he/she supplies `CDAIndividualHealthCarePractitioner` credential and satisfies the associated PTE and LE.

location attribute of the role `PermittedPhysicianPx` using the equality operator with the expected value of `NewYork`. An XML representation of LE_{EX2} using the syntax of our policy specification language appears in Fig. 9b.

Based on these concepts, a PE and a policy rule are formally defined as follows:

Definition 2 (Predicate Expression). A PE is a Boolean expression involving a set P of predicates such that every $p \in P$ is defined to evaluate a constraint expression, that is, $p: CR \rightarrow \{\text{true}, \text{false}\}$. A constraint associated with a predicate evaluates to true in one of the following ways:

1. It is a PTE, and the associated interval and periodicity conditions are satisfied.
2. It is an LE with clauses of the form $(y, \omega, \delta(p_1, \dots, p_m))$, and the expression is satisfied over the set of clauses. A clause evaluates to true if y compares with the return value of the function δ according to the comparison operator ω .

Example 3. $PE_{EX3} = p_1 \wedge p_2$ represents a PE, which evaluates to true if the constraint expression associated with p_1 is true, and the one associated with p_2 is true. If p_1 is defined on PTE_{EX1} , and p_2 is defined on LE_{EX2} , then PE_{EX3} is true if PTE_{EX1} evaluates to true, and LE_{EX2} evaluates to true.

Definition 3 (Policy Rule). A policy rule is a statement involving an administrative function $f \in F$, the execution of which is subject to a predicate $p \in P$. The set F includes the following functions:

1. $ur(de)assign(u, r, ucred)$, which (de)assigns a user u to (from) a role r subject to the user supplying a user credential $ucred$ and
2. $pr(de)assign(pm, r, rcred)$, which (de)assigns a permission pm to (from) a role r subject to the role supplying a role credential $rcred$.

Example 4. Let $pm = PBobCD$, $r = PermittedPhysicianPBob$, $rcred = PermittedPhysicianPBob$ and let p be defined as the periodic expression PE_{EX3} in Example 3. Then, the permission-to-role assignment policy in Fig. 9b is represented as $prassign(pm, r, rcred)$ iff p .

5.3 Example Policy

We shall now discuss an example policy involving the use of the constraint expressions in our framework to capture the requirements use cases outlined in Section 5.1. We consider a HealthCareFederation (HCF) clinical system. It is assumed in the example that all role names, document types, and patients are part of the HCF database. We do not concern ourselves with setting up the federated database, and only describe the design and enforcement of policies. The rules in the example policy, along with the use cases to which they relate, are given below:

Granularity of Access [UseCase1]

- R1. A US-board-certified physician can access medical data in any `ClinicalDocument`.
- R2. Locally certified physicians can only access clinical documents of type `Discharge Summary`.
- R3. A billing clerk can access administrative data in any `ClinicalDocument` of any patient.

Disclosure Rules [UseCase2]

R4. The disclosure policy for a billing clerk accessing an EHR of category "ClinicalDocument" requires the access to be restricted during the first week of any quarter in year 2005 and for a duration of one week.

R5. For a ClinicalDocument of type Discharge Summary, the resource access is restricted to only occur from within the state of NewYork.

Privacy Rules [UseCase3]

R6. The privacy policy of patient Bob allows a physician to access its records only if they have attributes board_certified_id with value NY and fellowship_field_cd with value GeneralMedicine in their CDA-encoded credential.

Emergency Defaults [UseCase4]

R7. The applicable privacy or disclosure policy for an EHR may be overridden if the access has to occur from near or inside the EmergencyRoom.

Information Hiding [UseCase5]

R8. The patient may not be able to view a ClinicalDocument of type Psychiatry Report.

We will use the following categories of resources as defined by CDA:

1. Clinical Document (CD): all EHRs belong to this category,
2. Discharge Summary (DS): an EHR belonging to this specialized category of CD,
3. Psychiatry Report (PR): an EHR belonging to this specialized category of CD, and
4. CDAD: the administrative portion of the CD (all types except CDAD mean medical portion).

Although we have already indicated the assignment rules that allow physicians to access the EHRs of patients, it is also necessary to encode rules that allow patients to view their EHRs. For this purpose, we use the convention that a user x is assigned to a role Px in order to access his/her own EHRs. A role Px , by default, has the permissions to access all its records, but manual overrides may be encoded by the system administrator in special circumstances. We focus only on the specification of policy rules for encoding privacy and disclosure policies. The details of policy administration in X-GTRBAC, including the rights and functions of system administrators, can be found in [9].

We now encode the policy rules and use, where applicable, the PE formalism.

Rule 1. Let $pm_{rule1} = PxCD$, $r_{rule1} = PermittedPhysicianPx$, and $rcred_{rule1} = PermittedPhysicianPx$, and let p_{rule1} be defined on

$$LE_{rule1} = (US, =, hasCredAttrValue \\ (PermittedPhysicianPx, board.certified.id)).$$

Then, the policy rule is represented as

$$prassign_{rule1}(pm_{rule1}, r_{rule1}, rcred_{rule1}) \text{ iff } p_{rule1}.$$

Rule 2. Let $pm_{rule2} = PxDS$, $r_{rule2} = PermittedPhysicianPx$, and $rcred_{rule2} = PermittedPhysicianPx$, and let p_{rule2} be defined on

$$LE_{rule2} = (NY, =, hasCredAttrValue \\ (PermittedPhysicianPx, board.certified.id)).$$

Then, the policy rule is represented as

$$prassign_{rule2}(pm_{rule2}, r_{rule2}, rcred_{rule2}) \text{ iff } (p_{rule2} \vee p_{rule1}).$$

Rule 3. Let $pm_{rule3} = PxCDAD$, $r_{rule3} = BillingClerk$, and $rcred_{rule3} = BillingClerk$. Then, the policy rule is represented as $prassign_{rule3}(pm_{rule3}, r_{rule3}, rcred_{rule3})$. Note that this policy rule has no associated constraint.

Rule 4. Let $pm_{rule4} = PxCDAD$, $r_{rule4} = BillingClerk$, and $rcred_{rule4} = BillingClerk$ and let p_{rule4} be defined on

$$PTE_{rule4} = LE_{EX1} = \langle P, [2005.Years, 2005.Years] \rangle, \\ P = all.Years + \{1, 4, 7, 10\}.Months \\ + \{1\}.Weeks \triangleright 1.Weeks.$$

Then, the policy rule is represented as

$$prassign_{rule4}(pm_{rule4}, r_{rule4}, rcred_{rule4}) \text{ iff } p_{rule4}.$$

Rule 5. Let $pm_{rule5} = PxDS$, $r_{rule5} = PermittedPhysicianPx$, and $rcred_{rule5} = PermittedPhysicianPx$, and let p_{rule5} be defined on

$$LE_{rule5} = PTE_{EX2} = (NewYork, =, hasCredAttrValue \\ (PermittedPhysicianPx, location)).$$

Then, the policy rule is represented as

$$prassign_{rule5}(pm_{rule5}, r_{rule5}, rcred_{rule5}) \text{ iff } p_{rule5}.$$

Rule 6. Let $u_{rule6} = any$, $r_{rule6} = PermittedPhysicianPx$, and $ucred_{rule6} = Px$. Let p_{rule6a} be defined on

$$LE_{rule6a} = (NY, =, hasCredAttrValue \\ (Px, board.certified.id))$$

and let p_{rule6b} be defined on $LE_{rule6b} = (GeneralMedicine, =, hasCredAttrValue (Px, fellowship_field.cd))$. Then, the policy rule is represented as

$$urassign_{rule6}(u_{rule6}, r_{rule6}, ucred_{rule6}) \text{ iff } (p_{rule6a} \wedge p_{rule6b}).$$

Rule 7. Let $pm_{rule7} = PxCD$, $r_{rule7} = PermittedPhysicianPx$, and $rcred_{rule7} = PermittedPhysicianPx$, and let p_{rule7} be defined on

$$LE_{rule7} = (EmergencyRoom, =, hasCredAttrValue \\ (PermittedPhysicianPx, location)).$$

Then, the policy rule is represented as

$$prassign_{rule7}(pm_{rule7}, r_{rule7}, rcred_{rule7}) \text{ iff } p_{rule7}.$$

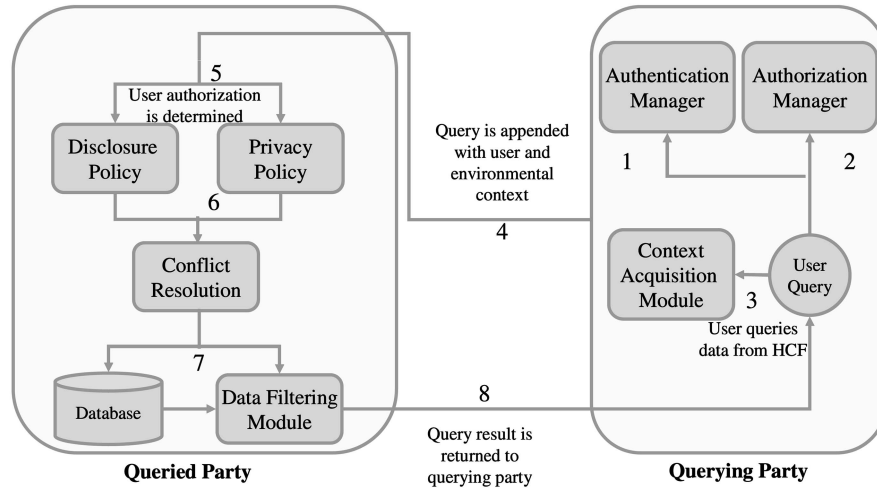


Fig. 10. The overall system architecture.

Rule 8. Let $pm_{rule8} = PxPR$, $r_{rule8} = Px$, and $rcred_{rule8} = \text{null}$. Then, the policy rule is represented as

$$prdeassign_{rule8}(pm_{rule8}, r_{rule8}, rcred_{rule8}).$$

We note that in practice, a combination of these rules may be needed in an assignment policy. We recall that our rule specification supports combining rules from multiple sources, which allows combining multiple PEs in a policy rule, as shown in Example 4. For instance, Rule 3 and Rule 4 may be combined in a permission-to-role assignment policy by using the AND rule-combining mode to ensure that both rules are satisfied before the policy returns true, whereas Rule 7 used for emergency defaults may be combined with any existing rule by using the OR rule-combining mode so that the policy always returns true when an emergency context has been detected (see Section 4.3 for the discussion on rule-combining modes). Also, with particular reference to Rule 8, it is an example of manual overriding of default privileges of a patient to access his/her own records. We note that the semantics of *deassign* is opposite to that of *assign*, and it removes the assignment of a permission from a role. To resolve conflicts, we associate a higher priority with the *deassign* operation.

Rule combining is an important feature of our specification language. The rules are based on a set of constraints that are represented in the language using the syntax described in Section 5.2. Rule combining essentially entails combining multiple constraints. The knowledge about which constraints are to be combined in a policy rule is typically gathered based on either design-time analysis or evolving system requirements. In any case, the actual changes to the policy are made manually by the system administrator by associating the appropriate constraints with the appropriate user-to-role or permission-to-role assignment policy rule. Though the rule specification can get complicated when dealing with expressive constraints, we recall that our policy is supported by a well-defined administration model [9] and management framework [7] for scalable maintenance to make these tasks easier.

6 SYSTEM ARCHITECTURE

In this section, we present the system architecture of our HCF clinical database. The overall system architecture is shown in Fig. 10. This architecture is employed at all parties in the HCF, and each party uses the definitions of the various policy components described in Section 4.2 to encode the disclosure and privacy policies. As indicated earlier, the policy documents constructed using these policy definitions form the federated schema of the system. We now provide an overview of the key components of the architecture.

6.1 Authentication and Authorization

A user (any actor in our use cases) wishing to request clinical data from the HCF needs to provide credentials defined as per the federated schema. To provide a scalable identity and authorization management infrastructure, the architecture employs an Authentication Manager and an Authorization Manager. The Authentication Manager is not directly a component of our authorization infrastructure but is used to issue an authenticating credential to the user (encoded as an XUS in our framework). Subsequently, this authenticating credential is presented to the Authorization Manager. The Authorization Manager is then responsible for the role assignment of the user request based on the attributes encoded in the user credential. Following a successful role assignment, the Authorization Manager issues an authorization credential to the user (encoded as an XRS in our framework). Since the Authorization Manager issues the credential defined as per the federated schema, the authorization credential issued by it is accepted at all federating sites within the HCF. The fact that the role assignment is done based on the attributes (and not the identity) of the user and that the users and roles are defined using credentials as per the federated schema makes this a scalable mechanism, since any user can be assigned to any role within a federating site based on its local access control policy.

6.2 Context Acquisition

We have determined that access to sensitive clinical records can be based on contextual conditions. Various parameters

can be used for the context-aware access management. These parameters may include identity, activity, location, and time, as defined in [1]. In our current discussion, we include the user context (which is included in the credential attributes) and the environmental context (such as location information). Both the user context and environmental context need to be appended to the access request before they are submitted to the queried party. Based on our requirements for location-sensitive access control in personal healthcare information management, we provide a location capture mechanism, which is part of the more general Context Acquisition Module (CAM). We envision the following function of the location capture service: It will capture location information through the IP address of the nearest registered access point (AP), through which the device of the requesting user communicates (we do not rely on using the IP address of the device itself because the devices in a pervasive computing environment will typically be mobile devices having temporary IP addresses not sufficient to track location). This mechanism will be used to satisfy the proximity requirement in our system and also to detect the loss of proximity to appropriately disengage the resource provisioning session. The design of a CAM providing such a service is outside the scope of our work and is the subject of ongoing research [17]. We, however, for the purposes of our prototype system, assume the existence of a software routine that simulates its effect. The location data is received by the system through a GUI, which allows the location of a device to be entered for the purposes of simulated testing. This is not an impediment for our purposes, since the functioning of the access control mechanism is orthogonal to the location capture mechanism. We plan in the future to integrate our current prototype with a CAM providing location service, as envisioned in our proposed architecture.

6.3 Disclosure and Privacy Policies

The query embedded with the context information is evaluated by the access control module of the queried party. The query is checked against the policy documents for the federating site defined as per the federated schema. The evaluation consists of two phases: 1) the privacy policy of the patient is checked for the requisite authorizations of the requesting user based on the user's supplied credentials and 2) the disclosure policy of the queried party is checked for any restrictions on the release of the requested content based on the user authorization. As we indicated earlier, it is possible that the rules may be conflicting (such as Rule 8 in the example policy in Section 5.3 conflicts with the default access of patients to their own records), and a conflict resolution mechanism is provided. The current strategy employed in our prototype is simply denial takes precedence, which has been implemented by assigning higher priority to deassignment rules.

We would like to say a few words here on patient consent. The patient consent is relevant here, since no such access or disclosure should be allowed to which the patient has not explicitly consented. It should be straightforward to see that the lack of consent can therefore be represented as a denial constraint in the privacy policy of the patient. For example, if patient Bob does not want New-York-certified

physicians to access his/her records as per Rule 2 (disclosure policy) in Section 5.3, his privacy policy can contain an explicit deassignment rule of the form

$$\text{prdeassign}_{\text{rule2}}(\text{pm}_{\text{rule2}}, \text{r}_{\text{rule2}}, \text{rcred}_{\text{rule2}}),$$

where pm_{rule2} , r_{rule2} , and $\text{rcred}_{\text{rule2}}$ are as defined in the original rule.

6.4 EHR Retrieval

Based on the privacy and disclosure policy, information violating user preferences or organizational rules can be omitted from the returned view of the data. The requested EHR content is retrieved from the HCF database and sent to the Data Filtering Module, where an appropriate view of the content is generated. Finally, the resulting view of the requested EHR content is returned to the querying party.

7 IMPLEMENTATION

In this section, we discuss the implementation of the example policy in Section 5. The policy has been implemented in our prototype system that has been designed based on the architecture described in Section 6.

7.1 Policy and Records Database

The policy documents for the HCF clinical database are stored in an XML Policy Base (XPB) at each federating site. The XPB contains all policy-related XML documents that collectively form the federated schema. These include XUS, XRS, XPS, XRTS, XURAS, and XPRAS. Also stored in the XPB are the policy definitions, including XCredTypeDef, XResTypeDef, XSoDDef, XTempConstDef, and XPredFuncDef. The set of XML policy documents for the example policy implemented in our prototype is provided in Appendix B, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>.

The actual resource type instances (that is, the XML-based EHRs) at a participating site to which the users of the HCF will be requesting access are stored in an XML database. Our current implementation stores them as XML-type tables in Oracle DBMS. In Fig. 11, we give the layout of the EHRs. Based on the requirement for fine-grained access to certain portions (administrative versus medical) and type (DS, PR, and so forth) of the CD, we have structured the EHR in a manner that allows retrieval of only the authorized content. The *PersonalInformation* tag contains the patient's personal information such as name, address, date of birth, identification numbers, and phones. This tag is of type "administrative" and, hence, an element having this tag name can be accessed only by a user who has been authorized to access a resource of type CDAD (recall that this type has been defined in Section 5.3 to refer to the administrative portion of the CD). All the other tags in the document are of type "medical" and, hence, an element having any of those tag names can be accessed only by a user who has been authorized to access a resource of type CD or any subcategory of it (recall that all types other than CDAD have been defined in Section 5.3 to refer to the medical portions of a CD). These tags include the *MedicalServiceProvider* tag, the *FamilyMedicalHistory* tag, and the *PatientMedicalRecords* tag.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<Patient id= "xxx">
  <PersonalInformation>
    ....
  </PersonalInformation>
  <MedicalServiceProvider>
    ....
  </MedicalServiceProvider>
  <FamilyMedicalHistory>
    ....
  </FamilyMedicalHistory>
  <PatientMedicalRecords>
    ....
  </PatientMedicalRecords>
</Patient>

```

Fig. 11. XML-based EHR layout.

Each individual record (EHR) presents an instance of interaction between the patient and one of the service providers. As already motivated, each EHR belongs to a category (resource type) as defined per the CDA standard, which is captured through the use of a type tag. Examples for the possible types are CD, DS, PR, Diet, Disability, and Observation. Note that binary encoding allows objects to be embedded within XML-based EHRs. Such objects may include multimedia data such as medical images, which can be accessed by users who have authorization to view the containing document.

7.2 Policy Enforcement

We now illustrate the enforcement of the example policy implemented in our current prototype. The idea is explained as follows: Access requests to retrieve the patient's EHR are composed using credentials obtained from the Authentication and Authorization Manager. The CAM appends the contextual information to the request, and it is then submitted to the queried party. At the latter end, they are first checked against the disclosure and privacy rules maintained in the XPB of the queried party. Based on these rules, queries are reformulated such that users can only access what they have authorization to. The reformulated query stores information about the patient and the authorized EHR types in a well-known format. The reformulated query is then passed to a procedure that parses the query and retrieves the required content from the EHR database. The answer to the query is stored in an XML document that contains only the required text and multimedia data. This result document is composed by the Data Filtering Module and is then returned to the querying party.

We now illustrate some scenarios of the evaluation of access requests using our example policy, where the requests are representative of the requirements use cases outlined earlier. In particular, we cover Rules 1, 2, 3, 4, 5, and 6 mentioned in Section 5.3. The scenarios involving the remaining rules will be similarly handled by making simple

extensions to the example policy to handle exceptions and denial, as already described

Scenario 1. Physician Smith wishes to access the CD of patient Bob. Smith has a US board certification.

Smith presents to the system a CDAIndividualHealthCarePractitioner credential encoded as an XUS (see Fig. B.3, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>). The credential contains the attributes `board_certified_id` having a value US and `fellowship_field_cd` having a value GeneralMedicine for Smith. The privacy policy of patient Bob is encoded as an XURAS (see Fig. B.9, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>), which requires that any physician having a board certification of NY is permitted to view the EHRs of Bob. Since Smith has a US board certification, and assuming that the US certification subsumes local certification, he is assigned the `PermittedPhysicianPBob` role (which is defined in XRS in Fig. B.5, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>). Following this role assignment, Smith is then evaluated for permission assignments against the constraints defined on the credential attributes to enforce the disclosure policy. The assignment of `CP_PBob_CPrCD_GET` permission (which is defined in XPS in Fig. B.8, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>) defined to access any Clinical Document of patient Bob requires the user to have a `board_certified_id` having a value US (see Fig. B.10, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>) and, hence, *Smith is eligible to be assigned the requisite permission.*

Scenario 2. Physician Carla wishes to access the Clinical Document of patient Bob. Carla has an NY board certification. Carla is accessing from New York.

Carla presents to the system a CDAIndividualHealthCarePractitioner credential encoded as an XUS (see Fig. B.3, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>). The credential contains the attributes `board_certified_id` having a value NY and `fellowship_field_cd` having a value GeneralMedicine for Carla. The privacy policy of patient Bob is encoded as an XURAS (see Fig. B.9, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>), which requires that any physician having a board certification of NY is permitted to view the EHRs of Bob. Since Carla has an NY board certification, she is assigned the `PermittedPhysicianPBob` role (which is defined in XRS in Fig. B.5, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>). Following this role assignment, Carla is then evaluated for permission assignments against the constraints defined on the credential attributes to enforce the disclosure policy. The assignment of `CP_PBob_CPrCD_GET` permission (which is defined in XPS in Fig. B.8, which can be found

on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>) defined to access any Clinical Document of patient Bob requires the user to have a `board_certified_id` having a value US (see Fig. B.10, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>) and, hence, *Carla is not eligible to be assigned the requisite permission.*

Scenario 3. Physician Carla wishes to access the Discharge Summary of patient Bob. Carla has an NY board certification. Carla is accessing from New York.

This is similar to the previous scenario, except that now the permission requested is `CP_PBob_CPrDS_GET`. The assignment of the `CP_PBob_CPrDS_GET` permission (which is defined in XPS in Fig. B.8, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>) defined to access the Discharge Summary of patient Bob requires the user to have a `board_certified_id` having a value of either US or NY and the role to have a `location` having a value of `NewYork` (see Fig. B.10, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>). Since these conditions are satisfied in this instance, *Carla is eligible to be assigned the requisite permission.*

Scenario 4. Billing clerk John wishes to access the Clinical Document of patient Bob. John is accessing in the second week of February in 2005.

John presents to the system a `ClinicPurdueBillingClerk` credential encoded as an XUS (see Fig. B.3, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>). The assignment policy for the `BillingClerk` role (which is defined in XRS in Fig. B.5, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>) is encoded as an XURAS (see Fig. B.9, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>), which does not require any user credential to be presented, but has an associated temporal constraint `PTQuarterWeekOne` (which is defined in XTempConstDef in Fig. B.4, which can be found on the Computer Society Digital Library at <http://computer.org/tkde/archives.htm>). This constraint states that the assignment is only allowed beginning the first week of every quarter of year 2005. Since John is accessing in the second week of February, which is not the first week of any quarter in 2005, *John is not eligible to be assigned the requisite role.*

Scenario 5. Billing clerk John wishes to access the Clinical Document of patient Bob. John is accessing in the first week of April in 2005.

This is similar as above, except that now the time of access is within the range of the temporal constraint. Since John is accessing in the first week of April, which is the first week of the second quarter of year 2005, *John is eligible to be assigned the requisite role.*

The implementation of the above scenarios, along with the related policy documents, can be accessed at <http://cobweb.ecn.purdue.edu/~iisrl/project/hcf>. Our current prototype uses SAML to encode the access requests as

SAML decision queries and to encode the authorization decision as SAML decision statements.

8 CONCLUSIONS

In this paper, we have presented a policy-based system for federated healthcare databases. One of our contributions is that we have highlighted the requirement for the integration of privacy and disclosure policies with well-known EHR standards used in the industry in order to specify the precise requirements of a practical healthcare system. To address this concern, we have used CDA as a representative EHR standard and designed our disclosure and privacy policies based on a set of use cases for the CDA standard proposed by the community. We have shown that our policy specification language is effective in terms of handling a variety of expressive constraints on the CDA-encoded EHRs as per the HL7 specification. Our second contribution is designing a system that not only meets organization-centric requirements for healthcare organizations (such as referrals) but also meets the requirements of the emerging personal healthcare information management, where patients can obtain ubiquitous healthcare services by using the same infrastructure that enables federated healthcare management for organizations. We have presented a context-aware policy specification language, which not only allows encoding disclosure and privacy rules using a declarative predicate-based syntax in the policy but also allows the use of temporal and nontemporal contextual constraints to be specified in the policy. The specification of disclosure and privacy rules allows privacy-aware access control for federated EHRs across organizational boundaries, whereas the use of contextual constraints allows the incorporation of user and environment context in the access control mechanism for personal healthcare information management. Moreover, the declarative syntax of the policy rules makes the policy adaptable to changes in privacy regulations or patient preferences. Our policy is supported by a well-defined administration model and management framework for scalable maintenance. We have also presented an enforcement architecture for the federated healthcare framework proposed in this paper.

Several improvements to our current work can be foreseen, both with respect to design and implementation. From the design perspective, we realize that the list of use cases covered in this work is certainly not exhaustive. For example, we currently do not cover use cases eliciting requirements not directly related to the information dissemination but rather its transmission. For example, Use Case 3 in Section 5.1 may require data encryption or perturbation in order to restrict the data access outside its originating institution. We also do not cover policy requirements as related to interdependency of CDs and history-based policy rules as may be relevant in the case of collaborative clinical work. The management of EHRs for such purposes opens up a whole new set of challenges. Such a scenario may include requirements based on the access history of the EHRs accessed by a heterogeneous pool of users and/or the semantic dependence between EHRs created by heterogeneous policy administrators. It may also include requirements for modifying or updating

the document or the associated constraints. The specification and compliance of these requirements is still an ongoing research challenge. We will attempt to pursue it as part of the future work on our policy specification framework.

On the technology side, it can be observed in the demonstration that the request processing incurs a measurable overhead, which is because we work with XML-encoded and richly populated EHRs (some include medical images as well). A (potentially) different EHR is retrieved from the database and appropriately pruned for every new request. The overhead of processing an XML-encoded EHR includes parsing and creation of XML objects, serialization, and manipulation of XML data. The focus of our current work is limited to policy design and engineering, and we have not particularly addressed the issues of processing XML-encoded documents. However, due to the increasing significance of XML as an interoperable data format, including multimedia data, XML parsers and XML-enabled browsers are increasingly getting more efficient at XML processing, and it is expected that future versions of these parsers and browsers will be equipped with various optimizations to improve the performance of dealing with richly populated documents and multimedia data. We therefore expect to mitigate this concern in the future by introducing optimizations in our parsing and browser-side code that makes use of improved technology.

REFERENCES

- [1] G.D. Abowd, A.K. Dey, P.J. Brown, N. Davies, M. Smith, and P. Steggle, "Towards a Better Understanding of Context and Context-Awareness," *Proc. First Int'l Symp. Handheld and Ubiquitous Computing (HUC '99)*, pp. 304-307, 1999.
- [2] R. Agrawal, P. Bird, T. Grandison, J. Kiernan, S. Logan, and W. Rjaibi, "Extending Relational Database Systems to Automatically Enforce Privacy Policies," *Proc. 21st Int'l Conf. Data Eng. (ICDE '05)*, pp. 1013-1022, 2005.
- [3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Hippocratic Databases," *Proc. 28th Int'l Conf. Very Large Databases (VLDB '02)*, 2002.
- [4] R. Agrawal, D. Asonov, R. Bayardo, T. Grandison, C. Johnson, and J. Kiernan, "Managing Disclosure of Private Healthcare Data with Hippocratic Databases," white paper, IBM, Jan. 2005.
- [5] R. Agrawal and C. Johnson, "Securing Electronic Health Records without Impeding the Flow of Information," *Proc. Int'l Medical Informatics Assoc. Working Conf. Security in Health Information Systems*, Apr. 2006.
- [6] L. Alschuler, "Layered Constraints: The Proposal for HL7 Healthcare Templates," *XML*, 2002.
- [7] R. Bhatti, E. Bertino, and A. Ghafoor, "X-FEDERATE: A Policy Engineering Framework for Federated Access Management," *IEEE Trans. Software Eng.*, vol. 32, no. 5, pp. 330-346, May 2006.
- [8] R. Bhatti, J.B.D. Joshi, E. Bertino, and A. Ghafoor, "X-GTRBAC: An XML-Based Policy Specification Framework and Architecture for Enterprise-Wide Access Control," *ACM Trans. Information and System Security*, vol. 8, no. 2, pp. 187-227, 2005.
- [9] R. Bhatti, B. Shafiq, E. Bertino, A. Ghafoor, and J. Joshi, "X-GTRBAC Admin: A Decentralized Administration Model for Enterprise-Wide Access Control," *ACM Trans. Information and System Security*, vol. 8, no. 4, pp. 388-423, 2005.
- [10] J. Byun, E. Bertino, and N. Li, "Purpose Based Access Control of Complex Data for Privacy Protection," *Proc. 10th ACM Symp. Access Control Models and Technologies (SACMAT '05)*, June 2005.
- [11] *HL7 Clinical Document Architecture, Release 2.0*, R. Dolin, L. Alschuler, S. Boyer, C. Beebe, F. Behlen, P. Biron, eds., Aug. 2004.
- [12] P. Fankhauser, G. Gardarin, M. Lopez, J. Munoz, and A. Tomasic, "Experiences in Federated Databases: From IRO-DB to MIRO-Web," *Proc. 24th Int'l Conf. Very Large Data Bases (VLDB '98)*, 1998.
- [13] G. Gardarin, S. Gannouni, and B. Finance, "A Distributed System Federating Object and Relational Databases," *Object-Oriented Multi-Database System: A Solution for Advanced Applications*, Prentice Hall, 1995.
- [14] D. Heimbigner and D. McLeod, "A Federated Architecture for Information Management," *ACM Trans. Information Systems*, vol. 3, no. 3, July 1985.
- [15] J. Hu and A.C. Weaver, "Dynamic Context-Aware Access Control for Distributed Healthcare Applications," *Proc. First Workshop Pervasive Security, Privacy and Trust (PSPT '04)*, 2004.
- [16] IBM, "The Enterprise Privacy Authorization Language (EPAL)," www.zurich.ibm.com/security/enterprise-privacy/epal, 2007.
- [17] V. Kumar and S. Zidonik, "Workshop Report," *Proc. NSF Workshop Context Aware Mobile and Sensor Information Management*, Jan. 2002.
- [18] M. Mont, R. Thyne, K. Chan, and P. Bramhall, "Extending HP Identity Management Solutions to Enforce Privacy Policies and Obligations for Regulatory Compliance by Enterprises," *HP Laboratories Technical Report 2005-110*, 2005.
- [19] F. Moss, "Clinical Record Use Cases," OASIS XACML Technical Committee, 2001.
- [20] OASIS SAML, <http://xml.coverpages.org/saml.html>, 2006.
- [21] OASIS XACML, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacm, 2006.
- [22] W. Pratt, K. Unruh, A. Civan, and M. Skeels, "Personal Health Information Management," *Comm. ACM*, vol. 49, no. 1, Jan. 2006.
- [23] X. Qian and T.F. Lunt, "A MAC Policy Framework for Multilevel Relational Databases," *IEEE Trans. Knowledge and Data Eng.*, vol. 8, no. 1, Feb. 1996.
- [24] M.P. Reddy, B.E. Prasad, P.G. Reddy, and A. Gupta, "A Methodology for Integration of Heterogeneous Databases," *IEEE Trans. Knowledge and Data Eng.*, vol. 6, no. 6, Dec. 1994.
- [25] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, "Role-Based Access Control Models," *Computer*, vol. 29, no. 2, pp. 38, 47, Feb. 1996.
- [26] A.M. Snyder and A.C. Weaver, "The Elogistics of Securing Distributed Medical Data," *Proc. First IEEE Int'l Conf. Industrial Informatics (INDIN '03)*, Aug. 2003.
- [27] M. Tempelton, D. Brill, A. Chen, S. Dao, and E. Lund, "Mermaid: Experiences with Network Operation," *Proc. Second Int'l Conf. Data Eng. (ICDE '86)*, 1986.
- [28] B. Thuraisingham and W. Ford, "Security Constraint Processing in a Multilevel Secure Distributed Database Management System," *IEEE Trans. Knowledge and Data Eng.*, vol. 7, no. 2, Apr. 1995.
- [29] A. Tomasic and L. Raschid, "Scaling Access to Heterogeneous Data Sources with Disco," *IEEE Trans. Knowledge and Data Eng.*, vol. 10, no. 5, Sept./Oct. 1998.
- [30] M. Wilikens, S. Feriti, A. Sanna, and M. Masera, "A Context-Related Authorization and Access Control Method Based on RBAC: A Case Study from the Health Care Domain," *Proc. Seventh ACM Symp. Access Control Models and Technologies (SACMAT '02)*, June 2002.
- [31] *Platform for Privacy Preferences (P3P)*, World Wide Web Consortium (W3C), www.w3.org/P3P, 2006.



Rafee Bhatti received the BS degree in electronics engineering from Ghulam Ishaq Khan Institute of Engineering Sciences and Technology (GIKI), Pakistan, in 1999 and the MS degree in computer engineering from Purdue University in 2003. He is currently a postdoctoral researcher at IBM Almaden Research Center in San Jose, California. This work was done when he was a PhD candidate in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, where he was also affiliated with the Center for Education and Research in Information Assurance and Security (CERIAS). His research interests include information systems security, with emphasis on design and administration of access management policies in distributed systems. His PhD research focuses on the access management problems posed by the emerging federated paradigm of information sharing and collaboration and on specification of XML-based security protocols for Web-based information systems. His work on XML-based access control framework for the Role-Based Access Control (RBAC) model has recently been cited by the OASIS consortium in their official announcement of the RBAC standard. He is a member of the IEEE and the IEEE Computer Society.



Haseeb Amjad received the BS degree in electrical engineering from the University of Engineering and Technology (UET), Lahore, in 2004. He is currently working toward the master's degree at Purdue University. His research interests include access management and information security in multimedia databases. He is a student member of the IEEE.



Arif Ghafoor received the BS degree in electrical engineering from the University of Engineering and Technology (UET), Lahore, Pakistan, in 1977 and the MS, MPhil, and PhD degrees in electrical engineering from Columbia University in 1977, 1980, and 1984, respectively. He is currently a professor in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, and is the

director of the Distributed Multimedia Systems Laboratory. He has served on the editorial boards of various journals, including the *ACM/Springer Multimedia Systems Journal*, the *Journal of Parallel and Distributed Databases*, and the *International Journal on Computer Networks*. He has served as a guest/coguest editor for various special issues of numerous journals including the *ACM/Springer Multimedia Systems Journal*, the *Journal of Parallel and Distributed Computing*, the *International Journal on Multimedia Tools and Applications*, the *IEEE Journal on the Selected Areas in Communications*, and the *IEEE Transactions on Knowledge and Data Engineering*. He has coedited the book *Multimedia Document Systems in Perspectives* and has coauthored the book *Semantic Models for Multimedia Database Searching and Browsing* (Kluwer Academic Publishers, 2000). He has been actively engaged in research areas related to database security, parallel and distributed computing, and multimedia information systems. He has published more than 175 technical papers in leading journals and conferences and has supervised 21 PhDs. He is a fellow of the IEEE. He received the IEEE Computer Society 2000 Technical Achievement Award for his research contributions in the area of multimedia systems.



Arjmand Samuel received the BS degree in aeronautical engineering from the College of Aeronautical Engineering, NED Engineering University, Pakistan, in 1989 and the MS degree in electrical engineering from the Beijing University of Aeronautics and Astronautics, Beijing, in 1997. He is currently working toward the PhD degree in the School of Electrical and Computer Engineering, Purdue University. From 1989 to 1994, he worked for the Pakistan Air Force. He

was the manager of software engineering at Pakistan Aeronautical Complex, Kamra, Pakistan, from 1997 to 2003. He worked as a visiting faculty at the Ghulam Ishaq Khan Institute of Engineering Sciences and Technology (GIKI), Pakistan, from 1999 to 2004 and taught in the Department of Electrical Engineering, COMSATS Institute of Information Technology, from 2003 to 2004. His research interests include context-aware access control models, access control policy verification, validation and conflict resolution, and requirement specification of access control policy. He is a student member of the IEEE.



Mohamed Y. Eltabakh received the BSc degree in computer science from Alexandria University, Egypt, in 2001 and the MSc degree in computer science from Purdue University, West Lafayette, Indiana, in 2005. He is a PhD graduate student and a research assistant in the Department of Computer Science, Purdue University. His current research interests include analysis of biological data, indexing, and data mining.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.