# Behavior-Based Access Control for Distributed Healthcare Environment

Mohammad H. Yarmand, Kamran Sartipi and Douglas G. Down
Dept. Computing and Software, McMaster University Hamilton, ON, L8S 4K1, Canada
{*yarmanmh, sartipi, downd*}*@mcmaster.ca*

## Abstract

*Privacy and security are critical requirements for using patient profiles in distributed healthcare environments. The amalgamation of new information technology with traditional healthcare workflows for sharing patient profiles has made the entire system vulnerable to security and privacy breaches. In this paper we present a novel access control model based on a framework designed for data and service interoperability in the healthcare domain. The proposed model for customizable access control captures the dynamic behavior of the user and determines access rights accordingly. The model is generic and flexible in the sense that an access control engine dynamically receives security effective factors from the subject user, and identifies the privilege level in accessing clinical data using different specialized components within the engine. Standard data representation formats are used to make the model compatible with different healthcare environments. The access control engine uses a flow-based approach to follow the user's behavior. The proposed model is supported by a real world case study.*

KEYWORDS: Security; Context Aware Access Control; Healthcare; Behavior; HL7; Patient Data.

## 1 Introduction

The cost of healthcare in civilized countries is rising rapidly due to the expectations for a higher quality of health service including: broad accessibility, customizability, cost efficiency, and most importantly reliability and security. Also, evolutionary changes in concepts within the healthcare domain have caused health professionals to embrace quickly growing distributed information and communication technologies. The new proposals for national or international healthcare standardization (e.g., HL7 and Canada Health Infoway) meet most of these requirements by adopting new techniques such as service oriented architectures (SOA) which remove the need to consider the details of the particular web technology employed for each distributed system. While solving the problem of interoperability among heterogeneous systems, SOA introduces many security and privacy issues - the natural consequences of lifting the level of abstraction and provision of customizability and ease of use. Regarding confidentiality, integrity and availability requirements of patient data, a major concern is to avoid disclosure of these data to unqualified users and to protect them from different attacks. Authentication and authorization methods at inter-/intra-organizational levels should be employed to achieve these requirements.

Various access control methods exist in the literature, however few consider the problem in distributed environments [3, 15, 18, 24]. Most access control methods only deal with static systems. However dynamism and configurability are two requirements of models for distributed systems [15, 18, 24, 25]. The proposed approach is generic and is customizable for different healthcare environments based on dynamic characteristics. This allows the system to be used for a specific environment such as a hospital or a laboratory.

Two major characteristics of our model are flexibility and the capturing of user behavior. Flexibility is gained by following semantic interoperability requirements. Two layers of repositories, storing static configurations and dynamic events of the system entities, provide an interface for the model engine. The concept of behavior is defined as following special patterns on a sequence of recorded attributes for each user. The technical requirements for context aware systems, necessary for behavior extraction, are supported by the model. Access control decisions are based on user behavior and existing policies by following a flow-based approach.

## 2 Related work

In this section, we present an overview of existing access control methods and their applications.

Role Based Access Control (RBAC) is the most common method and acts as a basis for other methods. In RBAC access rights are defined for roles instead of individual users. Each user is associated with a specific role and role privi-

leges are transferred to the user. There are some extensions for RBAC such as Generalized RBAC, Generalized Spatio Temporal RBAC [21] and Dynamically Authorized RBAC [18].

Team Based Access Control [9] considers the privileges of a user when they join a team and then applies the context of the team to the user. Content Based Access Control [10] considers access restrictions of resources based on content. The idea of role templates and Hippocratic databases (which may or may not be dependent on the users), are used to embed privacy in the data access layer. In Attribute Based Access Control [7] the access decision is based on properties (attributes) of the requester and of the resource, providing essential flexibility and scalability in the context of large distributed open systems. Situation Aware Access Control [24] monitors situation changes through situation-aware middleware and enforces run time policies. The situation is defined as an expression on previous device-actions over a period of time and/or the variation of a set of contexts relevant to the application software running on the device.

In distributed environments, there are some transactions which use resources of different organizations. Scenario Based Access Control [13] uses steps to define tasks and work profiles and checks user access rights for each step. Delegation is another important factor for secure distributed computing environments. The basic idea behind role-based delegation is that users themselves may delegate role authorities to other users to carry out some functions authorized to the former [26].

Context Aware Access Control (CAAC), the focus of our work, authorizes users based on their contexts. Regarding the nature of healthcare environments and the benefits delivered by Context Aware Systems (CAS) to the healthcare domain, we focus on the application of CAAC. Existing CAAC models suggested for healthcare are mainly a configuration of CAS and RBAC where context is treated as an additional constraint to the policy engine of RBAC [3, 15, 25]. In Context Sensitive Access Control [16] context is used for both user authorization and authentication.

## 3 Healthcare standards background

The healthcare industry has several standards development organizations developing specifications and standards to support healthcare informatics, information exchange, systems integration, and a wide spectrum of healthcare applications. HL7 [2] is an international community of healthcare experts and information scientists collaborating to create standards for the exchange, management and integration of electronic healthcare information. HL7 Version 3 uses the Reference Information Model (RIM), an object model that is a large pictorial representation of the clinical data (domains) identifying the life cycle of events that a message will carry, and applies object-oriented development methodology on RIM and its extensions to create messages [2]. HL7 has suggested a scenario based access control method and has defined the RBAC tables for different healthcare roles [12, 14].

Canada Health Infoway [6] is an organization that provides specifications for a standard and nationwide healthcare infrastructure. Infoway's mission is to accelerate the development of an interoperable Electronic Health Record (EHR) system that is compatible with standards and communications technologies. The Privacy and Security Architecture (PSA) group is responsible for provisioning security and maintaining information privacy. PSA has not yet suggested an architecture for security but it has offered two useful documents: EHR Privacy and Security Requirements [4] which discusses the general security requirements in the healthcare domain and refers to data usage restrictions under privacy rules; EHRi Privacy and Security Conceptual Architecture [5] which explains specifications of the communication environment and required common services.

The Health Insurance Portability and Accountability Act (HIPAA) requires the establishment of national standards for electronic health care transactions and national identifiers for providers, health insurance plans, and employers. HIPAA provides a list of security and privacy suggestions and legal requirements. The access control requirement suggested by HIPAA [11] includes unique user identification, emergency access procedures, automatic log-off and encryption and decryption. These requirements are considered in our model and other parts of the Infoway infostructure.

## 4 Behavior-based access control framework

To develop an adaptable access control method, it is essential to identify and satisfy the requirements of a distributed systems access control mechanism. Some of these requirements are as follows: both organization specific privacy rules and generic domain policies should be preserved as much as possible; the access control mechanism should consider the user context; the effect of a sequence of events performed by a user for future access decisions should be considered; the access control model should be general enough to cover different configurations and requirements of different organizations; the access control mechanism should support distributed systems management to maintain the integrity of resources; there should be an interaction point for the administrator to modify the policy rules; it should be possible to define temporal relations between system entities; and finally, all events which occur in the system should be logged for additional analysis.

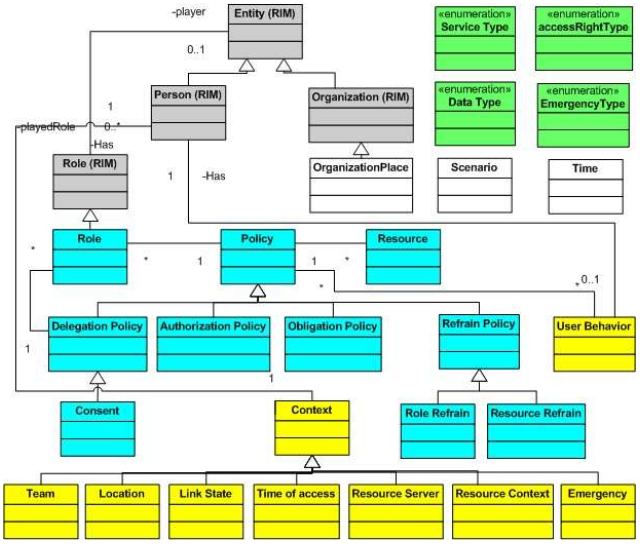


**Figure 1. Security factors class diagram**

### 4.1 Proposed model

Considering the goal of the proposed model as an add-on to the distributed healthcare architecture, the model will be connected to the communication layer, i.e., the service bus in service oriented architectures (SOA) and the Health Informatics Access Layer (HIAL) in Infoway's infostructure [6]. The communication layer sends a request for data access or service invocation to the model and the model returns a number representing the weighted user access rights to that specific resource or service. We now proceed to discuss the effective security factors to be considered for a decision. Then the basis of our access control decision, the behavior concept, is explained. Finally, the model's internal structure is described.

Figure 1 provides the UML class diagram of the security effective factors. Security effective factors must be dynamically identified and captured to ensure adaptation to different situations. A clear and accurate representation of security input factors and their inter-relationships are necessary for effective operation of other blocks of the model. We apply association rule mining among instances (objects) of different classes. This operation is intended to identify the effective factors affecting user behavior. Also, access control rules can be modified based on observed interactions between the objects.

In order to establish interoperability and reusability, the relations between these input factors and standard clinical data are defined, i.e., our class diagram is connected to the standard RIM classes. A few classes of HL7 RIM have been used in this class diagram. The Service type (top right) represents a list of services that a user invokes; this list is mapped to storyboards and transactions of different domains covering standard healthcare scenarios. The type of clinical data (i.e., Data Type, top right) is expressed using higher levels of standards clinical terminology hierarchies such as SNOMED and LOINC [22, 19].

There are four categories of classes in the proposed class diagram: i) HL7 classes which are labelled by *(RIM)* and located at the top; ii) context hierarchy classes (at the bottom) represent different contexts; iii) core security classes in the middle, and; iv) enumeration classes on the right side. We extend the policy classification offered by the Ponder project [8]. The rest of this subsection explains the major classes.

The class *Role(RIM)*, defined in HL7, represents the general role of a person, such as physician. However the class *Role* in our model refers to the security role - a subset of *Role(RIM)* and therefore the inheritance relation holds between the two classes. This class can represent either a functional or structural role.

The context class *Resource* (in the middle) represents the contexts over resources such as the "*pattern of access made to the resource*", "*type of data the resource contains together with their sensitivity level*", and "*users who had previously accessed the resource*". In a healthcare environment, communication is based on exchanging messages with the environment entities, standardized by the HL7 v3 community. Hence a patient profile is the collection of messages that are exchanged between the care givers. The content of a profile is determined by monitoring the flow of exchanged messages that correspond to the type of data for that profile.

The access control decision engine, the main decision making factor in our model, uses concepts from CAS methods to model user behavior. Context aware models define some logical constraints over context and restrict the set of possible context configurations. These constraints are placed in the class *Policy* to maintain model integrity. To meet the requirements of CAS as a central role player in the model, a major portion of the class diagram has been allocated to represent security related contexts. These contexts are inherited from the general class *Context* (bottom of diagram) where detailed attributes are used to express them. An additional context named *Emergency*, which determines a situation's emergency level based on parameters such as time, location, role and resource, is defined under class *Context* . The class *User Behavior*, composed of a set of contexts, represents the user behavior concept to make access control decisions. User behavior also contains additional information explained in the following subsections.

### 4.2 User behavior

In this section we explain the concept of *user behavior* and describe how it can be used for making access control decisions. We define an *Action* that is performed by a user, as a tuple composed of attributes:

$Action$ = <*Person, Role, User Location, Server Location, Time of Day, Team, Delegation, Requested Profile Status, Service Invocation Type, Requested Data Type, Login/Logout Event*>

where *Person* identifies the user; *Role* is the user security role; *Server Location* is where the requested resource server is located; *Team* refers to those rights and actions which are permitted only when the user is a team member; *Delegation* explains the access rights given or taken by delegation rules or consents; *Requested Profile Status* refers to properties of the requested profile explained in the class *Resource Context* of the input factor class diagram; *Requested Data Type* refers to the clinical data type (mapped to higher layers of clinical terminologies); *Service Invocation Type* is the type of service requested (mapped to Infoway transactions). *Login/Logout Event* identifies usage of an ongoing session or a login/logout event.

A *Behavior* is defined as a sequence of actions that can be manifested in two forms:

- *Time-span behavior.* A record of a sequence of actions performed during a specified period, e.g., during the last five hours, a day, a month, etc.

- *Snapshot behavior.* A record of particular attribute(s) of the "same action" in consecutive days to extract specific behavior over a long period of time.

Whenever an attribute of the Action of a user changes, a new tuple is recorded. Since we are modeling the privileges of the user, any changes in the set of user access rights should be monitored. A new tuple may be recorded even if the user has not requested access to a resource. For example when a user joins a team, privileges change and therefore a new tuple should be recorded even if the user does not request access to a resource.

**Behavior based access control**

**Single action** represents a single action tuple. Given a single action tuple we choose one of the action attributes as a key attribute and use it to constrain the domain of other attributes. If *Role* is the key attribute, the domain of other attributes would be limited based on *Role*. In order to determine how the domains are filtered according to the *Role* value, general clinical guidelines or hospital policies defined for that specific role can be used. If *Person* is the key attribute, the domain of other attributes would be limited based on that specific user. This makes our model very dynamic and flexible. In order to determine how the domains are filtered according to a specific user, the history of action tuples recorded for that user is analyzed to extract associated domain values.

**Daily behavior** consists of a sequence of action tuples recorded in one day for a given person. Some access control processes require more than a single tuple to be able to make an access decision. Examples are: log in-out pattern; duration and correctness of attempts; sequence of service invocations; spatial proximity of consecutive actions; and policy rules explicitly defined over time such as tasks or restrictions of a person on special days of a week.

**Snapshot** represents the historical aspect of our system. It considers the same attributes of actions of a person in consecutive days (called snapshot behavior). The results of this analysis are also used in the Single Action section.

### 4.3 Model's internal structure

In this subsection we describe different blocks of the proposed access control model, illustrated in Figure 2.

**Input**. The values of effective security factors previously considered.

**Representation**. In order to make the system interoperable and usable in different environments, input factors must be mapped to a standard format. In this way, when a workflow spans multiple organizations with different security architectures, change to the internal security architectures of each organization is not required. In the healthcare domain, HL7 RIM provides a hierarchy for clinical roles which we adopt as our standard ontology for roles [14].

**Configuration storage**. Repositories reside between the input layer and engine block as an interface for the engine. The purpose of using the repositories is to avoid losing model generality by making the engine independent to any special data format. The input data and additional configurations are stored in the repositories.

**Cross input storage**. This layer has the same purpose as configuration storage, but it considers relations between the inputs and also the dynamic attributes of system entities such as contexts of users and resources.

**Decision making engine**. The decision is based on information gained about a user, distributed among four decision blocks. *Critical Access Control* enforces the privacy and policy rules including relations between users, roles, resources and permissions. This block is responsible for reasoning over different rules to discover the policy that should be applied for a user. *Action Access Control* checks for domain membership and CAAC constraints introduced as "single action" in *time-span* behavior. *Behavior Access Control* checks for daily behavior in the *time-span* behavior

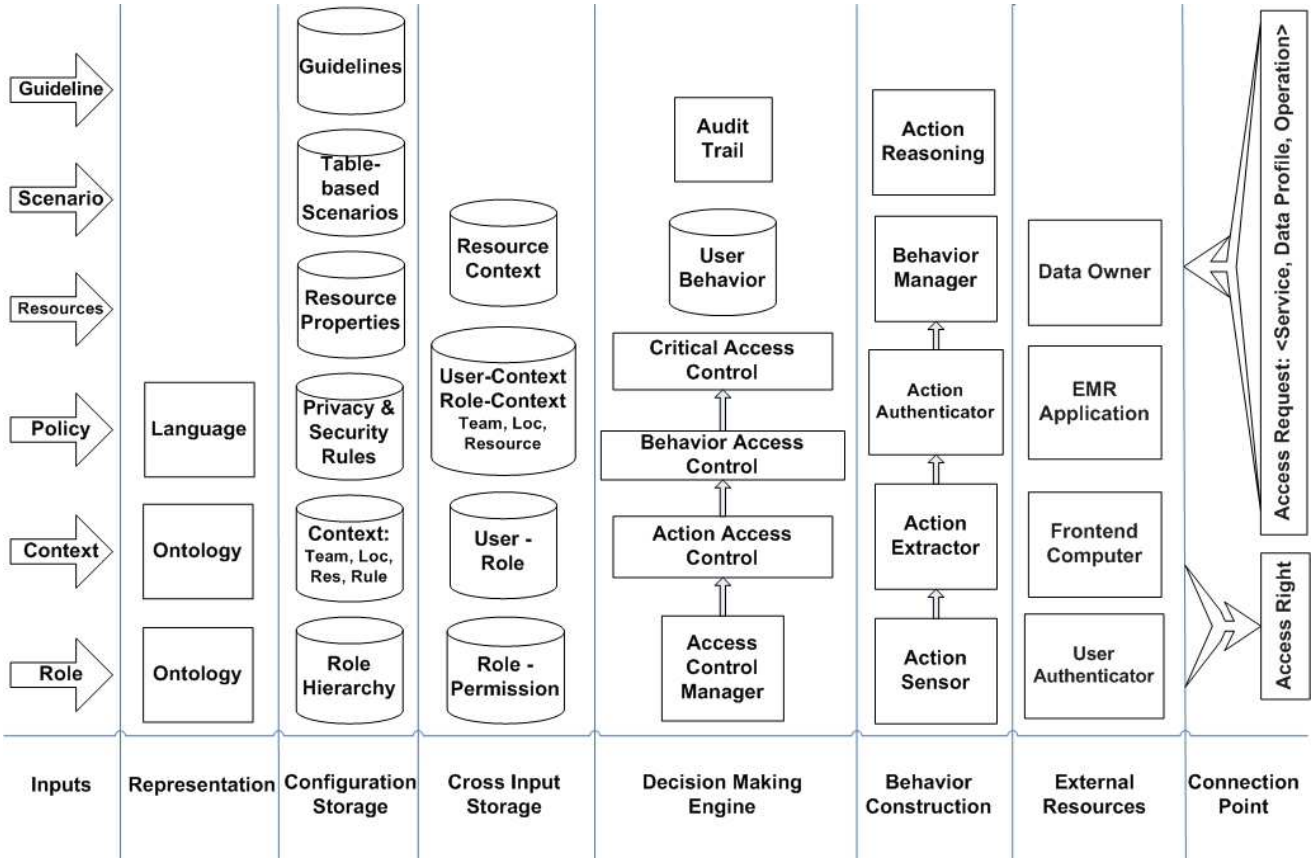


**Figure 2. The Proposed Access Control Model**

defined above. The behavior of the user is compared with the expected user behavior. Different mathematical models are used for representing various aspects of behavior. Guidelines are used to model a sequence of attribute values in an action tuple. Matrices represent the association relations discovered between two or three of the attributes. The rules resulting from the analysis are dynamically generated and new inquires are verified against them. The *Access Control Manager* manages decisions based on the results gained from each of the access control blocks in the decision making engine.

Different security factors and user behaviors have different effects on the access control decision. For example RBAC constraints in a policy class have greater effect than spatial proximity of user and server. Therefore we use effectiveness coefficients for different factors to reach the final decision. Each access request must obtain a minimum credit to be granted access. A possible refinement of this algorithm follows.

The variable "Merit" is initialized to a positive value and each time the user violates one of the access control checks, a penalty value is deducted from the Merit value. This penalty value is selected based on effectiveness of the associated access control check, i.e., failure to satisfy the most important factor causes the greatest reduction from "Merit". At the end of the merit evaluation process if the Merit is greater than zero, access is granted and the Merit value is returned; otherwise access is denied. The only exception is the emergency situation which grants access regardless of the current Merit value. A user who is not authorized by RBAC might be authorized either through a team membership or delegation and consent rules. In such a case the penalty would be ineffective and Merit is restored.

Since access decisions are made in this layer, it is the best place to place the Audit Trail block . The audit trail establishes a historical record of user or system actions over a period of time and provides an answer to the question: "*what have you done*?". The project "Integrating to Healthcare Enterprise" (IHE) [17] has a refined audit trial for distributed healthcare environments.

**Behavior construction**. This layer is responsible for constructing the basis for the engine layer, the *Action* and *Behavior* concepts. The blocks which are required to capture and represent these concepts are explained below.

*Action sensor* senses any changes in the attributes of the action and informs another block to extract the required data. *Action Extractor* composes the action tuple based on the data sensed by Action Sensor. *Action authenticator* authenticates the context itself. Different methods such as: statistical analysis, distributed reputation, and confidence value, are used for authenticating contexts [23].

*Behavior manager* composes the behavior based on the new action tuple and the past history of user behavior and updates the user behavior repository. *Action reasoning* uses

the rules provided by context input to infer the contexts that can not be directly understood such as an emergency situation.

## 5 Case study environment

We are involved in a project with industrial partners to integrate their eHealth systems. This is one of the first integration projects compliant with new standards and employing the latest technologies in this field. The proposed access control ideas are deployed and evaluated in this project to provide a realistic case study environment. COMPETE III Vascular Tracker (C3VT) [1] is a decision support system that assists physicians to observe and ideally control patients' different risk factors within the domains of cardiovascular, diabetes, hypertension, and dyslipidemia diseases. The COMPETE research group would like to extend the scope of C3VT by providing its services to other research groups' specialized databases. The project allows C3VT to interoperate with a Cardiac Rehab Center (CRC). In this integration a portion of patient data form CRC is sent to C3VT. C3VT algorithms are run over these data and recommendations and guidelines are returned to CRC.

**System architecture**. Oracle's Healthcare Transaction Base (HTB) is a Service Oriented Architecture that supports the integration, development, and operation of a full spectrum of healthcare applications. The rational behind choosing HTB as implementation environment is that HTB follows HL7 v3 messaging standards and is compatible with the Infoway infostructure. Here is a list of services which are used in this project [20]: Enterprise Master Person Index, Messaging Services, Enterprise Terminology Services, Security Services, RIM Services and Service Discovery. In the integration engine, the selected HL7 messages are composed using RIM and the mapping between clinical terms and standard clinical terminologies are passed to HTB through messaging services.

## 6 Future work

An API must be offered for unifying the usage of the representation layer. The available technologies and equipments (both hardware and software) should be reviewed to determine a minimum set of technologies for our model to extract the required contexts. The formal definitions and appropriate technologies are specified, but not mentioned here due to space limitation. Another potential application is using constructed behavior to guide and influence future actions.

## References


[1] Compete official website. www.compete-study.com.

[2] Health Level Seven ballot. www.hl7.org/v3ballot/html/welcome/environment/index.htm.

[3] R. Bhatti, E. Bertino, and A. Ghafoor. A trust-based context-aware access control model for web-services. *Distrib. Parallel Databases*.

[4] Canada Health Infoway. EHR Privacy and Security Requirements, 2005. v1.1.

[5] Canada Health Infoway. EHRi Privacy and Security Conceptual Architecture, 2005. v2.

[6] Canada Health Infoway. EHRS Blueprint, an interoperable EHR framework, 2006. v2.

[7] E. Damiani, S. D. C. di Vimercati, and P. Samarati. New paradigms for access control in open environments. In *Signal Processing and Information Technology*, pages 540–545, 2005.

[8] N. Damianou, N. Dulay, E. Lupu, and M. Sloman. The ponder policy specification language. In *International Workshop on Policies for Distributed Systems and Networks*, pages 18–38, 2001.

[9] C. K. Georgiadis, I. Mavridis, G. Pangalos, and R. K. Thomas. Flexible team-based access control using contexts. In *Sixth ACM symposium on Access control models and technologies (SACMAT)*, pages 21–27, 2001.

[10] L. Giuri and P. Iglio. Role templates for content-based access control. In *Second ACM workshop on Role-based access control*, pages 153–159, 1997.

[11] HIPAA. Security standards: Technical safeguards, 2007. version 2.

[12] HL7. RBAC healthcare scenarios, 2005. v2.

[13] HL7. RBAC role engineering process, 2005. v1.1.

[14] HL7. Hl7 healthcare scenario roadmap, 2006. v2.2.

[15] J. Hu and A. C. Weaver. A dynamic, context-aware security infrastructure for distributed healthcare applications. In *the First Workshop on Pervasive Privacy Security, Privacy, and Trust*, 2004.

[16] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, and J. Reitsma. Context sensitive access control. In *Tenth ACM symposium on Access control models and technologies (SACMAT)*, pages 111–119, 2005.

[17] C. H. Infoway. IHE IT infrastructure techinical framework - volume 1 - integration profile, 2007. revision 4.

[18] C. J. Kuo and P. Humenn. Dynamically authorized role-based access control for secure distributed computation. In *The 2002 ACM workshop on XML security (XMLSEC)*, pages 97–103.

[19] LOINC Committee. LOINC users' guide, 2007. 2.22.

[20] ORACLE. Oracle Healthcare Transaction Base - datasheet, 2005.

[21] A. Samuel. Context-aware access control policy engineering for electronic health records. In *Research Saminar at CIMIC*, 2007.

[22] SNOMED. SNOMED clinical terms guide - abstract logical models and representational forms, 2006. version 5.

[23] K. Wrona and L. Gomez. Context-aware security and secure context-awareness in ubiquitous computing environments. In *Annales UMCS Informatica*, pages 332–348, 2006.

[24] S. Yau, Y. Yao, and V. Banga. Situation-aware access control for service-oriented autonomous decentralized systems. In *Autonomous Decentralized Systems*, pages 17–24, 2005.

[25] G. Zhang and M. Parashar. Dynamic context-aware access control for grid applications. In *Fourth International Workshop on Grid Computing*, page 101, 2003.

[26] L. Zhang, G.-J. Ahn, and B.-T. Chu. A role-based delegation framework for healthcare information systems. In *Seventh ACM symposium on Access control models and technologies (SACMAT)*, pages 125–134, 2002.