A Personalized Access Control Framework for Workflow-Based Health Care Information

Nazia Leyla and Wendy MacCaull

Centre for Logic and Information
St. Francis Xavier University
Antigonish, Canada
{x2009gte,wmaccaul}@stfx.ca

Abstract. Access control is one of the key features of any health care organization. Without a strong access control mechanism, there is a risk of inappropriate use of personal health information. Here we focus on Personalized Access Control (PAC) [1] where the patient decides who can access his/her health record. We enhance the PAC model of [1] by proposing a prototypical framework, which incorporates a workflow into the PAC model to express the context of health care processes, and by providing a mechanism to capture a patient's consent to enforce the PAC policy. We enforce the "need to know" principle by associating roles with each task in a workflow and handle problems with delegation. We present a case study outlining the present working procedures of the Seniors' Wellness Program in our local health authority, using NOVA Workflow for workflow modeling and Ponder2 for representing and enforcing policy.

Keywords: personalized access control, workflow, health care information system, EHR.

1 Introduction

The world-wide use of information systems, allowing us to store, organize, gather, extract, and investigate an array of services, is increasing rapidly and with it, there is a growing need for security. Access control is one of the most important security aspects for protecting information from unauthorized use. Access control is particularly important in the health care information systems. As more and more patient information is recorded electronically, it becomes essential to protect that information from unauthorized access and, therefore, misuse. Currently, Electronic Health Record (EHR) systems are becoming increasingly common for storing health information [2]. These EHRs contain a great deal of health data, including sensitive information, such as fertility status and abortion history, emotional problems, HIV status, physical abuse, and so on. Without a strong access control mechanism, there is a risk of breach of security resulting in an inappropriate access to personal health information, which can not only adversely affect the patient but also lead to complaints, allegation of negligence and possible liability for the organization. Protecting information from abuse, thus, ensuring people's right to privacy is, therefore, a major concern in the management, design, and development of health care infrastructure [3].

F. Daniel et al. (Eds.): BPM 2011 Workshops, Part II, LNBIP 100, pp. 273–284, 2012.

The vast majority of articles, dealing with the development and implementation of generic access control polices, models, and mechanisms (38 in 52 articles) [4], use the Role Based Access Control (RBAC) model in order to develop their access control systems. The RBAC [5] is the most common method used in health care organizations and acts as a basis for other methods. RBAC associates permissions to groups of users according to their roles within the organization. However, health related data is owned by the patient, and it should be disclosed only when permission is obtained from the patient [6]. In [1], the authors provide a model for a Personalized Access Control (PAC) framework where the patient is the administrator of his/her health record. PAC is about making sure information is accessible only to authorized users, which allows the patient to grant a person read and/or write access to his/her health record and to revoke this when they choose.

In a health care organization, it is also necessary to ensure that works are performed in a planned way meeting health care requirements. A Workflow Management System (WfMS) enables health care organizations to automate their health care process, in order to enhance efficiency and effectiveness. To ensure that only authorized users execute workflow tasks, appropriate authorization mechanisms must be in place, so that authorization is granted only when the task starts and is revoked as soon as the task finishes [7]. Getting patients' permissions for the disclosure of their health records can be represented as a task in a health care workflow.

In this paper, we enhance the PAC model [1] and propose a prototypical framework, which incorporates a workflow into the PAC model, and provides a mechanism of capturing a patient's permissions and enforcing the PAC policy. We focused on three problems of an access control mechanism: the incorporation of a patient's permissions with the access control mechanism, the "need to know" principle, and delegation. We collected information about the present working procedures of a Seniors' Wellness Program and constructed a workflow model using the NOVA Workflow [8] modeling tool. We identified the access control requirements of this program, converted them into policies, represented by Ponder [9] policy language, integrated the Ponder2 policy interpreter with NOVA Workflow to enforce those policies, and conducted a case study validating the proposed framework.

The paper is organized as follows: in Section 2, we analyze the PAC requirements and describe a high-level design of our PAC framework. Section 3 presents a patient scenario for the Seniors' Wellness Program. Section 4 presents the implementation and the validation of our framework. In Section 5, we conclude the paper and give a discussion of related and future work.

2 A Personalized Access Control Framework

Based on the literature [1,10,11], discussions with different health care providers (HCP) from the local health care authority - the Guysborough Antigonish Strait Health Authority (GASHA) - and the existing GASHA forms for the Seniors' Wellness Program, we articulated the following requirements for our PAC framework:

- The patient decides what permissions to assign to whom.
- Two policy sets: Common access policy, determined by the hospital (or other institution) where the patient is being treated; and Personalized access policy, determined by the patient to protect the privacy of the information stored in the EHR must be detailed.
- The patient is not allowed to update or delete the Common policies.
- The patient is allowed to update or delete any of his/her Personalized access policies at any time.
- Who may give consent if the patient is unable to give consent must be known.
- A specific relationship must be established between the patient and the required HCP before a health service is started. There are two kinds of situations in which a HCP offers a health care service:
  - The patient and the HCP have not established a specific relationship yet, e.g., a new patient and/or an outpatient. In this case, an authorization setup must be established before service can be given to the patient.
  - The patient and HCP have already established a specific relationship, e.g., an inpatient and/or a follow-up patient.
- HCP can conduct appropriate operations on a patient's EHR. Read and write are two common types of operations.
- Whenever a new HCP is added to a patient's care team, the patient is notified immediately via an e-mail, phone, fax, or any other communication service, so that he/she can give/deny consent for that HCP.

We propose a PAC framework addressing these requirements, which is illustrated by Fig. 1. The framework begins by a subject (we use subject and HCP interchangeably) executing a task (1). The Role Management service authenticates the subject with the information stored in the Database (e.g., MySQL) (2). While executing the task the engine generates an access request on behalf of the task (3) and sends this request to the Ponder Policy Interpreter (4). The Interpreter executes the access request and sends back the patient's personal decision to the subject (5). The main components of the framework are: the NOVA Workflow Management System, a Role Management Module, a Policy Interpreter, and an EHR. We discuss each below.

NOVA Workflow Management System. The WfMS we use in our framework is NOVA Workflow [8], an innovative workflow management system developed by our research group. It provides a workflow execution engine, and a graphical editor for workflow specifications. We integrated our policy interpreter with NOVA Workflow using service classes, which are generated automatically for each task in the workflow. In our framework, a workflow represents a process of a health care organization as a set of well-defined tasks which are executed according to the health care organization's policies to achieve certain objectives. We develop workflow models using NOVA Workflow editor and the workflow is executed by the workflow engine. There are some tasks in the workflow that need access to an EHR for their successful execution. While executing a task, the workflow engine may generate an access request. This access request includes

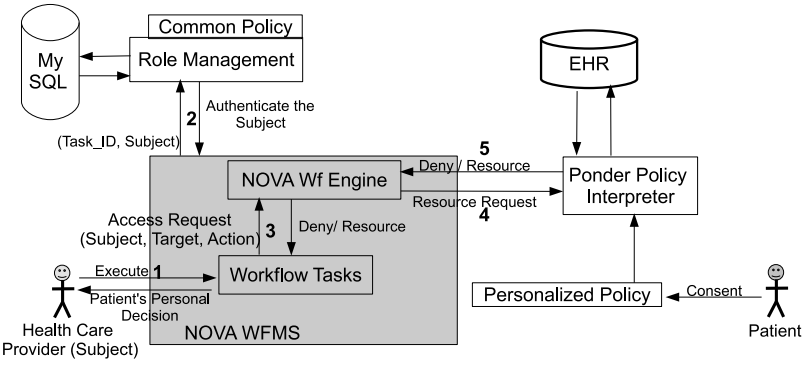

Fig. 1. Workflow-based PAC Framework

the following information related to this task: who wants to access the patient records (the subject), what the information is that the subject wants to access (the target), the type of operation (e.g., read, write, etc.), and on which instance the task is performed (the patient). A task in the workflow contains the information necessary to generate an access request.

The Role Management (RM) Module. After the representations of the tasks into NOVA Workflow, the allocation of roles to tasks must be made. We developed a module for NOVA Workflow to support role management. In our framework, the roles associated with each task are stored for authentication. Assigning a role to each task guarantees that at runtime, work items for each of these tasks are offered only to participants that perform that role. Unlike Personalized policies which are enforced on the individual subject while accessing a patient's record, the RM module enforces Common policies on the subjects based on their roles (with or without some constraints, such as time, location, etc. E.g., the policy may not allow a HCP to execute a task after 5 P.M., even if the HCP has the patient's consent). This ensures the "need to know" principle.

The Ponder2 Policy Interpreter. Another important component is the Ponder Policy Interpreter, which uses the policy framework Ponder2 [9]. To protect resources from unauthorized access, Ponder2 provides Authorization policy, which is a set of (subject, target, action)-tuples, which defines the activities that a member of the subject domain can perform on the set of objects in the target domain. We specify policies using the Ponder2 policy language and the interpreter organizes the subjects and the targets based upon which policies operate in hierarchical domains of Managed Objects (MO), which are an abstract representations of subjects and targets specified in a workflow. Each MO has methods for operations of those workflow tasks that need access to the patient's record. The execution of a task in the workflow corresponds to the execution of a method in the corresponding MO. When the method is executed the operation will be performed; the corresponding authorization policy is activated

dynamically when the corresponding workflow task is executed. Hence there is a direct mapping between access policies and workflow tasks, which ensures the HCP can only perform the operations they are allowed to do.

We assign access rights directly to the HCP who needs access to a record. These rights define what actions can be performed on a health record by the subject executing the task. The policy interpreter derives the access rights from the patients's Personalized policies and enforces them. The interpreter allows the insertion and activation of new authorization policies at run time. This feature allows us to dynamically adapt the access rights of a subject to the actual needs expressed in the task that is executing.

Electronic Health Record (EHR). The EHRs act as the resources in our work. Suppose an EHR is already established by the health care organization for which we are specifying the workflow. Workflow generates a request on behalf of a subject to access the records of this EHR. The subject may wish to perform several operations, like reading, writing, or updating a patient's data on this EHR. Patients have control over their own EHR by giving/denying the subject permission for each of the operations.

3 Case Study

This work is a part of a research and development project in collaboration with GASHA and a technology industry partner; the goal is to develop a next generation careflow management system for information, communication and process management and pilot it in several programs in GASHA. GASHA has established a service area for Seniors' Health, called the Seniors' Wellness Program, in response to an aging population and ongoing pressures on the Acute Care system created by the increasing number of individuals in hospital medical beds waiting for nursing home placement. The program strives to enhance coordination and continuity along the continuum of care, including Outpatient, Inpatient, Continuing Care, Adult Day Program, Seniors Health Services, Geriatric Assessment Clinic (GAC), Community Rehabilitation Services (CRS), and Volunteers.

Here we consider two aspects of this Program: the GAC, which focuses on the prevention and treatment of diseases and disabilities in older adults, and decreases the effects of aging on the body; and the CRS, which is a short-term outpatient program specializing in intensive rehabilitation.

We interviewed different HCPs in the GAC and analyzed the recorded interviews, the paper-based forms (for assessment and other purposes) and other documents, and built the GAC workflow model. We finalized the model with the Project Manager of the Program in three iterations. Fig. 2 shows the high-level model of the GAC consisting of composite tasks. Each of the composite task has a subnet workflow. Here we outline a representative set of tasks of the workflow involving the GAC and the CRS for describing our proposed framework. Before executing the tasks, HCPs (Secretary1, Nurse1, Nurse2, Doctor1) must be registered in the system and authorized to execute tasks appropriate for their role.

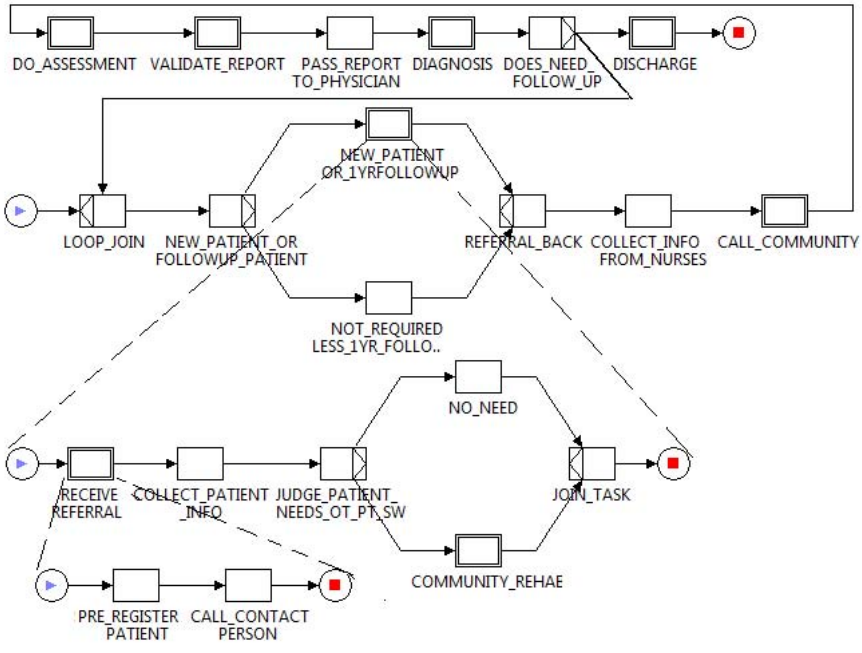

Fig. 2. Workflow Model for the GAC of the Seniors' Wellness Program

Task 1 (PRE_REGISTER_PATIENT). Secretary1 receives the referral of a patient, say Adam, and pre-registers the patient. At this point, Adam's demographic information is taken including a designated contact person.
Task 2 (CALL_CONTACT_PERSON). Once Secretary1 assigns Adam a suggested care team (from those authorized HCP in the GAC) she contacts Adam or his designated contact person to get their consent (or Secretary1 can build an initial care team by consulting with Adam, if he is available at that time). Then Adam explicitly assigns with whom he wants to share his record.
Task 3 (COLLECT_PATIENT_INFO). Nurse1 needs to see Adam's record to collect background information and to identify the relevant parts of the information from the consultations that are done by other HCPs.
Task 4 (JUDGE_PATIENT_NEEDS_OT_PT_SW). Nurse1 makes a decision about whether Adam needs any Occupational Therapy (OT) or Physiotherapy (PT) and makes a referral to the CRS.
Task 5 (COLLECT_INFO_FROM_NURSES). Secretary1 collects information from the nurses, e.g., which home care the patient uses, previous PT, OT etc.
Task 6 (CALL_COMMUNITY). Secretary1 contacts the CRS and asks for copies of their reports, the home care for the services they have, and the family physician for other doctors' consultation information. This information is sent to Secretary1.
Task 7 (DO_ASSESSMENT). Nurse2 does assessments and different tests.
Task 8 (VALIDATE_REPORT). After gathering all the test results, Nurse2 talks with family members to validate the information.

Task 9 (PASS_REPORT_TO_PHYSICIAN). Nurse1 and Nurse2 put all results, past histories, etc. into the system.
Task 10 (DIAGNOSIS). Doctor1 accesses the information for Adam's diagnosis.
Task 11 (DISCHARGE). Adam is discharged from the clinic.

Suppose in Task 7, Nurse2 delegates her responsibility to another nurse, say Nurse4, to carry out Task 7. Nurse4 does assessments and different tests on behalf of Nurse2. An authorization setup and a relationship with Adam is needed here for Nurse4 as well. To execute Task 7, Nurse4 needs to be registered first. Then she is authorized to do the task, after that Secretary1 takes Adam's consent for Nurse4 for accessing his health record. Therefore, a delegation is same as adding a new HCP into the care team.

In some situations a patient does not want to be notified each time a new person is added to his domain. After the initial selection of the care team there will be an option of whether he/she wants to be notified each time a HCP is added to his/her care team.

Table 1 shows HCPs and their roles, tasks, and task-related access rights.

Table 1. Access Rights Need for the Health Care Providers

HCP	**Role**	**Task #**	**Access Rights**
Secretary1	Secretary	Task 1 & Task 2	Needs No Access
Nurse1	Nurse	Task 3 & Task 4	Needs Read Access
Secretary1	Secretary	Task 5 & Task 6	Needs Read Access
Nurse2	Nurse	Task 7 & Task 8	Needs Read and Write Access
Nurse1 & Nurse2	Nurse	Task 9	Need Read and Write Access
Doctor1	Doctor	Task 10	Needs Read and Write Access

4 Implementation and Validation of the PAC Framework

4.1 Implementation

Implementing the Role Management Module. In our framework, the Role Management Module provides two services: mapping the tasks to the roles in the workflow and generating an identification for each HCP. The task mapping function is defined as a (Task_ID, Role)-tuple, where Task_ID is the identification of a task in the workflow and Role is the job position held by the HCP in the health care domain. At design time a set of roles are mapped and associated with each Task_ID. Here we assume that an employee has a uniquely identified Role. A HCP must login with her authentication credentials and let the system authenticate her. Based on her credentials the system identifies her Role and determines if it corresponds to a (Task_ID, Role)-tuple. Common policy is applied to her in this way.

Implementing the Ponder Access Control Module. In order to provide access rights to a patient's record during the workflow execution, Personalized access policies are associated directly with those HCPs who execute the tasks. The implementation of the module consists of three steps:

I. Create and instantiate Managed Objects (MOs). Represent all the subjects and targets associated in the workflow as MOs and assign them to the appropriate place in the domain hierarchy for use in the policy specification.

To start the execution of the tasks in the workflow, it is required to instantiate the MOs corresponding to the subjects who execute the tasks. Fig. 3 illustrates how a task starts after the instantiation of the MO. The subject is registered in the system (1). The MO Initialization file for that subject will be created (2.1) along with a credential, which is stored for the authentication (2.2). The credential is provided to the subject (3). For executing the task the subject gives his/her credential (4). The authentication service then validates the given credential (5). The given credential for the subject then instantiates the MO and thus loads the workflow task (6).

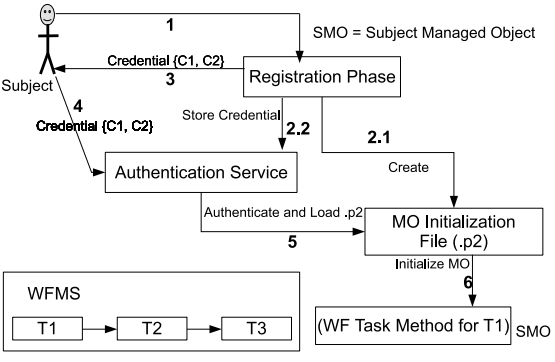

Fig. 3. Instantiating a MO and Loading the Workflow Task

II. Create and update the authorization policies from the consents given by the patients. An authorization policy is defined on a (subject, action, target)-triple; in the policy specification we use the domain path for specifying the subject and the target. Here the subject is the MO we have created for the HCP, the target is the patient's MO (the EHR is accessed through the patient's MO), and the action is the operation of a workflow task that the subject's MO needs to do on the Patient's MO. To specify authorization policies, the subjects executing the actions as depicted in Fig. 4 must be identified to the patient, who gives or withholds consent for each.

III. Integrate NOVA Workflow with the Ponder2 Access Control Module. To integrate NOVA Workflow with our system we extended the service classes and implemented our actual work. When a task invokes a method in the service class, the method collects the task information (subject, target, type of operation). Based on the subject and the target, Ponder2 loads the MOs and enforces the authorization policy. Ponder2 can be run as either a stand-alone application or can be started within a Java Virtual Machine (JVM); we use the latter.

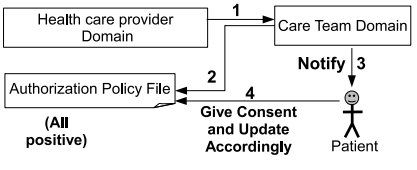

Fig. 4. Process of Creating Authorization Policy

4.2 Validation of the PAC Framework

We use the case study from Section 4 to show how patient consent is obtained and validate that the authorization policy is enforced on the tasks.

During the phase of a HCP registration, according to our case study Secretary1, Nurse1, Nurse2, and Doctor1 are registered in the system. Only the hospital administrator can register new subjects with the system. A Secretary needs to be authorized in the system first to receive the referral for the patient and to process the referral. When Secretary1 is registered into the system two things happen: (1) An e-mail goes to the registered Secretary along with her login e-mail_ID/Password. Her e-mail will be her login e-mail_ID and the password is generated randomly; and (2) A Secretary MO Initialization file is generated, which includes the domain path for the Secretary along with the specification of the Secretary to create an instance through the Secretary MO. The registration process is the same for all the other subjects.

After the registration process, Secretary1 gets her credentials (login e-mail_ID/Password). When Secretary1 provides the credentials to the system, the authentication service then matches her role with the stored {Task_ID, Role}-tuple and gives permission to execute the task, as the Role *secretary* is associated with the Task_ID *Receive_Referral*. Thus while a task is executing, the access control mechanism checks the access right of a task against the current needs of the HCP executing the task and the Common policy is applied. Therefore, the workflow allows us to enforce the "need to know" principle.

In the getting consent phase,

- the four subjects, Secretary1, Nurse1, Nurse2, and Doctor1 are assigned to Adam's care team, i.e., an authorization policy file is created for Adam that contains a segment of an authorization policy for each of the assigned subjects. The default authorization policy for all the subjects is initially positive, meaning that all actions are authorized unless a negative authorization policy is specified.
- an e-mail is sent to Adam, which includes a web service link for each of the subjects along with a link for giving consent to each subject. Note that, the patient/designated contact person can be notified for giving consents by means other than an e-mail, such as phone, fax, etc.

Suppose that Adam had some bad experiences with Nurse1, and does not want to share his record with Nurse1; he, therefore, gives a negative consent. Then a negative authorization policy is assigned to Nurse1.

When Nurse1 executes Task 3 (see Section 4) the authorization policy for Nurse1 is dynamically loaded. When a new task is activated in the workflow (here Task 3), its corresponding service classes start executing and thus the WfMS can identify the task, and enable the corresponding authorization policy. As we know from the previous section, the execution of a task in the workflow may correspond to the execution of a specific method of the subject's MO. Therefore, here the execution of Task 3 depends on the execution of a corresponding method in the Nurse MO that executes the access request for Nurse1. Task 3 shows that

Nurse1 needs to read Adam's therapy history. Nurse1 proceeds to execute Task 3 after giving the credentials. The workflow engine identifies the task and its associated role and associated authorization policy. At this point, the Common policy is applied and Nurse1 gets permission to execute the task. When Nurse1, via her Nurse MO, performs the action of reading Adam's therapy history for the Patient MO, the access control mechanism captures such an action and enforces the authorization policy, which is Adam's Personalized policy. Because the negative authorization policy for that action is already in place, the action of reading the therapy history cannot be authorized. This means the access request to Adam's health record is denied and the following alternative action takes place: the Secretary is notified that Nurse1 is not allowed to access Adam's EHR, the Secretary then removes Nurse1 from Adam's care team and assigns a new subject (a Nurse) to his care domain, and the process of taking Adam's consent begins again.

The newly assigned nurse is Nurse3 and we assume Adam does not have any problem with her, so he gives her permission to get his record. After the enforcement of the Common policy as above, when Nurse3 requests a read action on Adam's therapy history, she gets access to Adam's EHR as she has been given the read access right, and gets the record of previous OT and/or PT history. The tasks are designed in such a way that Nurse3 can get information about Adam's Therapy only at this point of Task 3 and thus we can ensure that the required part of the record is accessible precisely when it is needed. After this task, the WfMS engine invalidates the corresponding authorization policy. All policies related to tasks in Table 1 are enforced the same way as the above so the tasks' corresponding operations may be performed.

5 Discussion and Conclusion

Atluri et al. in [7] introduced the Workflow Authorization Model where authorization constraints for data and resources were synchronized with the execution of the workflows. In [12] an access control matrix was used for regulating access control of data in the execution of a workflow task. Russello et al. [13] presented a workflow-based access control mechanism that adapts the access rights of subjects to the actual tasks that they have to fulfill. Although in these three methods, access rights were provided on the basis of the workflow tasks, they did not consider getting patients' consents before accessing patients' health records. The authors in [6] described a framework for enforcing patients' consents based on YAWL WfMS, which did not show how to enforce Common policy. In addition, their approach to getting patient's consent was different as they considered different types of policies called consent meta policies. In [14] the authors proposed a decentralized approach to handle access control in a workflow, which modeled security policies jointly with the workflow specification; their approach was not task-based, and they neither considered of getting patients' consents nor how to handle delegation.

There are issues that were outside the scope of this work, but are important to mention. Here we use the PAC for protecting patients' information, but it

is not an absolute solution for securing a system. To improve security it could be coupled with auditing. Auditing requires the recording of all user requests and activities for later analysis. Research incorporating auditing into the access control system may be found in [15]. Our approach is not suitable for emergency situations; there may be emergency situations when waiting to get patient permissions could cause the death of the patient. The Privacy Legislation Act for urgent or emergency health care in [16], lists several situations where a HCP may provide health care to an adult without the adult's consent. In such emergency cases, a break-the-glass (BTG) [17] procedure should be applied in an ad-hoc manner, which would permit HCPs to override the existing access control rules and access what they need for continuing a patient's treatment. Our approach is suitable for long term health care processes where patient care spans over a long duration, e.g., Palliative care and Senior care. In our work, a single HCP is responsible for a task but in real life some tasks may require a team of HCPs; dealing with a team remains future work for us. For large and complex systems, the explicit description of the domain hierarchy will be a cumbersome process. However these complex relationships can be expressed precisely using an ontology. Research directions for supporting the access control model using an ontology can be found in [18,19].

In this paper, we described a PAC framework for a health care information system incorporating a workflow system, and validated it for a real world health care application. The main drawback of current access control mechanisms is that the granting of access rights requires statically binding a subject to a resource, where the subject and the resource must be known in advance. By using a workflow, however, we guarantee that access rights are dynamically adjusted to the actual needs of the subject. It is anticipated that the PAC framework can be integrated with other WfMS. More details including a formalism for our framework and some performance issues showing how the system scales may be found in [20]. We could also consider the hierarchical context of the workflow tasks by recognizing the composite task and its subnet workflows using the same variable. In the future, we will incorporate a mechanism for handling emergency health care situations, develop an audit-trail system for monitoring, and incorporate an ontology to structure the access control policies.

References

1. Rostad, L., Nytro, O.: Personalized access control for a personally controlled health record. In: CSAW 2008: Proceedings of the 2nd ACM Workshop on Computer Security Architectures, pp. 9–16. ACM, New York (2008)
2. Rostad, L.: Access control in healthcare applications. In: NOKOBIT 2005, pp. 241–253 (2005)
3. Jacobsson, A.: Privacy and Security in Internet-Based Information Systems. PhD thesis, Blekinge Institute of Technology (2008)
4. Ferreira, A., Chadwick, D., Antunes, L.: Modelling access control for healthcare information systems. In: Doctoral Consortium at the 9th International Conference on Enterprise Information Systems, ICEIS (2007)

5. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role based access control models. IEEE Computer 29(2), 38–47 (1996)
6. Russello, G., Dong, C., Dulay, N.: Consent-based workflows for healthcare management. In: Proceedings of the 2008 IEEE Workshop on Policies for Distributed Systems and Networks, pp. 153–161. IEEE Computer Society, Washington, DC, USA (2008)
7. Atluri, V., Huang, W.: An Authorization Model for Workflows. In: Martella, G., Kurth, H., Montolivo, E., Hwang, J. (eds.) ESORICS 1996. LNCS, vol. 1146, pp. 44–64. Springer, Heidelberg (1996)
8. Rabbi, F.: Design, development and verification of a compensable workflow modeling language. M.Sc., St. Francis Xavier University (expected 2011) Preliminary version, http://logic.stfx.ca/~software/DDVCWML.pdf
9. Twidle, K., Lupu, E., Dulay, N., Sloman, M.: Ponder2 - a policy environment for autonomous pervasive systems. In: POLICY 2008: Proceedings of the 2008 IEEE Workshop on Policies for Distributed Systems and Networks, pp. 245–246. IEEE Computer Society, Washington, DC, USA (2008)
10. Wei, D.: Privacy protection reference model for shared electronic health record, M.Sc. Thesis, Dalhousie University (2005)
11. Personal Information Protection and Electronic Documents Act, C.I.O.H.R.C, http://www.cihi.ca/CIHI-ext-portal/pdf/internet/protection_qa_EN (last accessed March 2011)
12. Knorr, K.: Dynamic access control through petrinet workflows. In: Proceedings of the 16th Annual Computer Security Applications Conference, pp. 159–167. IEEE Computer Society, New Orleans (2000)
13. Russello, G., Dong, C., Dulay, N.: A workflow-based access control framework for e-health applications. In: International Conference on Advanced Information Networking and Applications Workshops, pp. 111–120. IEEE Computer Society, Los Alamitos (2008)
14. Samiha, A., Cuppens-Boulahia, N., Cuppens, F.: Deploying access control in distributed workflow. In: Proceedings of the Sixth Australasian Conference on Information Security, AISC 2008, vol. 81, pp. 9–17. Australian Computer Society, Inc., Darlinghurst (2008)
15. Fernández-Medina, E., Trujillo, J., Villarroel, R., Piattini, M.: Access control and audit model for the multidimensional modeling of data warehouses. Decision Support Systems 42, 1270–1289 (2006)
16. (Consent), H.C., 181, C.F.A.A.R.C., http://www.bclaws.ca/EPLibraries/bclaws_new/document/ID/freeside/00_96181_01 (last accessed March 2011)
17. Ferreira, A., Chadwick, D., Farinha, P., Correia, R.C., Zhao, G., Chilro, R., Antunes, L.: How to securely break into RBAC: The BTG-RBAC model. In: Proceedings of the 25th Annual Computer Security Applications Conference (ACSAC), pp. 23–31. ACM press (2009)
18. Lymberopoulos, L., Lupu, E., Sloman, M.: Ponder policy implementation and validation in a cim and differentiated services framework. In: Proceedings of IFIP / IEEE Network Operations and Management Symposium, Seoul, South Korea, pp. 31–44 (2004)
19. Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W., Thuraisingham, B.: ROWLBAC - Representing Role Based Access Control in OWL. In: Proceedings of the 13th Symposium on Access control Models and Technologies. ACM Press, Estes Park (2008)
20. Leyla, N.: A personalized access control framework for workflow-based healthcare information. M.Sc. Thesis, St. Francis Xavier University (2011)