A Framework for Modeling Privacy Requirements in Role Engineering

Qingfeng He and Annie I. Antón
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8207, USA
{qhe2, aianton}@eos.ncsu.edu

Abstract

Privacy protection is important in many industries, such as healthcare and finance. Capturing and modeling privacy requirements in the early stages of system development is essential to provide high assurance of privacy protection to both stakeholders and consumers. This paper presents a framework for modeling privacy requirements in the role engineering process. Role engineering entails defining roles and permissions as well as assigning the permissions to the roles. Role engineering is the first step to implement a Role-Based Access Control (RBAC) system and essentially a Requirements Engineering (RE) process. The framework includes a data model and a goal-driven role engineering process. It seeks to bridge the gap between high-level privacy requirements and low-level access control policies by modeling privacy requirements as the contexts and obligations of RBAC entities and relationships. A healthcare example is illustrated with the framework.

1. Introduction

As the Internet and e-commerce have prospered, privacy has become of increasing concern to consumers, developers, and legislators. Legislative acts, e.g. Health Insurance Portability and Accountability Act (HIPAA) for healthcare [HIP96] and Gramm Leach Bliley Act (GLBA) for financial institutions [GLB01], require these industries to ensure consumer data's security and privacy. Companies and organizations protect consumer privacy in various ways, including publishing a privacy policy on their websites, enabling a P3P [P3P02] compliant privacy policy, incorporating a privacy seal program (e.g. Truste, BBBOnline), etc. However, these approaches cannot truly safeguard consumers because they do not address how personal data is actually handled after it is collected [AER02, AEP01, GHS00]. Companies' and organizations' actual practices might intentionally or unintentionally violate the privacy policies they published on their websites. Privacy violations are increasingly disclosed over the Internet, TV, newspaper and other medias, such as the famous Toysmart [Toy00] and Eli Lilly [Eli02] cases.

Privacy protection can only be achieved by enforcing privacy policies within an organization's online and offline data processing systems. Most organizations have one or more privacy policies posted on their websites. Due to separation of duties in an organization, privacy policies are usually defined as high-level natural language descriptions by an organization's privacy group, chaired by the Chief Privacy Officer (CPO). High-level natural language privacy policy descriptions are difficult to enforce directly via access control. Similarly, security polices are usually defined by another group of people in the organization, chaired by the System Security Officer (SSO). However, privacy requirements are often not reflected in the design and implementation of security policies. Thus, there exists a gap between security and privacy protection that is exacerbated by conflict of interests between stakeholders, system developers, and consumers. Researchers contend security and privacy requirements should be considered during initial system design [AE01, AEP01, AEC02]. Thus, modeling security and privacy requirements in the early stages of system development is essential for security and privacy enforcement.

Role-Based Access Control (RBAC) [SCF96, FSG01] has received increasing attention because it offers many additional benefits compared with traditional Discretionary and Mandatory Access Controls (DAC and MAC) [AS00]. RBAC is considered as a promising alternative to traditional MAC and DAC models [OSM00], especially in the healthcare domain. "It is generally accepted that RBAC is more suited to healthcare than other access control mechanisms to meet the requirements for the security of healthcare information" [ZAC02]. The Privacy-Aware RBAC (PARBAC) model enforces privacy policies in an organization [He03a], but it lacks a mechanism for mapping privacy requirements into the PARBAC model.

Role engineering for RBAC is the process of defining roles, permissions, role hierarchies, constraints and assigning the permissions to the roles [Coy96]. It is the first step to implement an RBAC system and essentially an RE process. Before a system can realize all the benefits of RBAC, the role engineering activities must occur, yielding a complete specification.

Security requirements are modeled in the role engineering process. For example, the well-known separation of duties security requirement is modeled by defining exclusive roles; least privilege security requirement is modeled by assigning each role a minimum set of permissions to perform each task. However, privacy requirements are not addressed in role engineering. For example, purpose binding, i.e. data collected for one purpose should not be used for another purpose without user consent, is an important privacy requirement. To date the security and RE literature does not address purpose elicitation and modeling in role engineering. Another issue regarding to privacy protection is user privacy preferences modeling and the integration of these preferences with access control authorizations. A mechanism is needed to model privacy requirements and user privacy preferences in a systematic way so that privacy policies can be enforced in the software system.

This paper presents a goal-driven framework for modeling privacy requirements in the role engineering process. We model privacy requirements as contexts and constraints of permissions and roles using goal-based RE techniques. These contexts and constraints serve as a basis for defining access control policies. The proposed framework seeks to bridge the gap between high-level privacy requirements and low-level access control policies in the early stages of system development and provide a basis for enforcing privacy requirements with RBAC.

The rest of this paper is organized as follows. Section 2 provides a summary of related work. Section 3 describes privacy protection elements modeling. In Section 4, the framework for modeling privacy requirements is described. Then in Section 5, a healthcare example is illustrated with the framework. Finally, a summary of the paper is given in Section 6. The limitations of the framework and future work are also discussed in this section.

2. Related work

This section provides an overview of relevant work in role engineering, goal-driven requirements engineering, and privacy policies and requirements.

2.1. Role engineering for RBAC

There exist several role engineering approaches, the first of which applies scenarios. Neumann and Strembeck proposed a scenario-driven approach for engineering functional roles in RBAC [NS02]. In this approach, each task is depicted using a collection of scenarios and each scenario is decomposed into a set of steps. Because each step is associated with a particular access operation, each scenario is linked to a set of permissions. The work is limited in that it is only effective to derive functional roles. Fernandez and Hawkins suggested determining the needed rights for roles from use cases [FH97].

Crook et al. proposed an analytical role modeling framework to derive roles from organizational structures [CIN02]. Although this provides a way to derive roles, not all roles can be derived from organizational structures. The method is not general and does not address role constraints. Epstein proposed a layered model for engineering role-permission assignment by introducing three intermediaries between roles and permissions: jobs, workpatterns, and tasks [Eps02, ES01]. Epstein's approach provides an effective way to assign permissions to roles and aggregate permissions into roles. Roeckle et al. proposed a process-oriented approach for role finding to implement role-base security administration [RSW00]. Their approach provides a method to find roles but does not address how to find permissions and how to assign permissions to roles.

Unfortunately, neither of these approaches [Eps02, ES01, FH97, RSW00] considers constraints and role hierarchies. Epstein and Sandhu's UML based approach documents components of an RBAC model in UML syntax [ES99]. This approach can assist the role engineering process but it does not provide a method for deriving roles. Kern et al. proposed an iterative-incremental life-cycle model of a role in the context of enterprise security management [KKS02]. The role life-cycle concept is very important for security administration; however, this approach fails to support the derivation of roles and permissions. Schimpf argued role engineering is a critical success factor for enterprise security administration [Sch00]. He proposed to organize a role engineering project and follow a clearly defined life-cycle model for roles.

In conclusion, the above-discussed approaches focus on different aspects of role engineering. Each work has its own strengths and weaknesses. None of these approaches addresses privacy requirements.

2.2. Goal-driven requirements engineering

Goal-driven RE employs goals to elicit, specify, analyze, and validate requirements. Kavakli identified seven major goal-oriented methods in RE [Kav02]. A complete overview of goal-driven RE techniques is beyond the scope of this paper. Herein we only discuss goal-scenario combination approaches. A more complete overview of goal-driven RE approaches can be found in [Lam01, Kav02].

Goals and scenarios have complementary characteristics [Lam01]. Goals are usually abstract and declarative. They are high-level objectives of the business, organization or system. Scenarios are concrete, narrative, and procedural. They describe real situations using examples and illustrations. Hence combining goals and scenarios is an effective way to elicit and validate requirements. Goals are operationalized through scenarios and refined into requirements [AMP94]. Similarly, scenarios can be used to help discover goals [AP98].

The GBRAM uses goal hierarchies to organize requirements as scenarios, goal obstacles, and constraints [Ant96]. Others also organize scenarios hierarchically according to goals and goal obstacles [Coc97]. Rolland et al. proposed a bidirectional goal-scenario coupling approach between goal discovery and scenario authoring [RSA98]. Kaindl proposed a systematic design process based on a model combining scenarios with goals and functions [Kai00]. In the combined model, "purpose" serves as a link between functions and goals: a system's aggregated functions have some purposes and these purposes match the (sub)goals of the users. Purpose has also been integrated with scenarios to model tasks in one of Kaindl's early works [Kai95]. This paper herein builds upon this notion of purpose.

2.3. Privacy policies and requirements

Two major privacy protection principles are the OECD guidelines for data protection [OEC80] and the FTC Fair Information Practice (FIP) Principles [FIP98]. The OECD guidelines define eight privacy principles: collection limitation, data quality, purpose specification, use limitation, security safeguards, openness, individual participation, and accountability. The OECD principles intend to protect personal data privacy while pursuing free information flow between different organizations and different countries. The five FIP principles (notice/awareness, choice/consent, security/integrity, access/participation, and enforcement/redress) are less complete than the OECD guidelines. Both the OECD and FIP principles provide the general privacy requirements with which organizations should comply. Several industries have additional legislative acts (e.g. HIPAA and GLBA) regulating their data practices.

Based on these general privacy principles and acts, each organization defines its own privacy policies. These policies are the major privacy requirements that an organization should enforce in their data processing systems. For example, when websites collect information from customers, they need to inform customers for what purpose the data is being collected, who the data recipient is, how long the data will be kept, and how the data will be used, etc. (notice/awareness principle in FIP). They should also provide opt-in/opt-out choices for customers or obtain customer consent on how to use the collected data (choice/consent principle). The actual data operations of companies and organizations should be consistent with user consented privacy policies (enforcement/redress principle).

Fischer-Hubner summarized four privacy aspects that a system should protect: confidentiality of personal data, integrity of personal data, purpose binding of accesses to personal data, and necessity of personal data processing (i.e. the collection and processing of data shall only be allowed if it is necessary for completing appropriate tasks) [Fis01]. Confidentiality and integrity have been the focus of the security community for a long time. The principle of necessity can be enforced with task-based authorization models, such as the Workflow Authorization Model (WAM) [Fis01]. However, purpose binding is not addressed in traditional security models.

Similarities and differences between policies and requirements are identified in [AEP01]. Antón and Earp have proposed strategies to employ scenario management and goal-driven requirements analysis methods for specifying security and privacy policy for secure electronic commerce systems [AE01]. Antón et al. have also applied goal-based requirements analysis to align software requirements with security and privacy policies [AEC02]. A privacy requirements taxonomy for websites has been presented in [AE03] by using goal-mining techniques on privacy policies. In this taxonomy, privacy requirements are classified as either privacy protection goals or privacy vulnerabilities. This paper builds upon these specification techniques to better support modeling of privacy requirements in role engineering. All sample privacy policies given in this paper are privacy goals identified from 23 websites' privacy policies in Antón et al.'s goal-mining exercises [AE03].

3. Privacy elements modeling

High-level privacy policies and requirements that are specified with natural language must be formalized into authorization rules before they can be technically enforced. Therefore, it is necessary to identify privacy protection elements in the role engineering process.

A typical access control rule is expressed as a tuple <*s*, *o*, *op*>, such that a subject *s* can access an object *o* on operation *op* [DD82]. A subject could be a user or a program agent. In an RBAC policy, this rule is expressed in another way: <*u*, *r*, *p*> [SCF96]. A user *u* can only access an object, if he/she is assigned a role *r*, and if the role is assigned certain permission *p*, which is allowed to access the object. A permission is usually represented as the combination of some operations on an object. Although the form is different, the basic elements of an RBAC rule are still subjects, objects, and operations.

These three elements, however, are insufficient to represent a privacy authorization rule. For instance, purpose binding is an important privacy requirement as we discussed in Section 2.3, but purpose is not reflected in the <*s*, *o*, *op*> tuple. In addition to the above three basic authorization elements (subjects, objects, and operations), three other privacy elements (purposes, conditions, and obligations) are identified in a privacy authorization rule [KS02]. Our framework builds upon these privacy protection elements as we now discuss.

3.1. Purposes

Purpose is a standard entity in most privacy policies as recognized in P3P [P3P02]. To enforce purpose binding privacy requirements, two kinds of purpose are identified: consumer data purpose and business purpose. Consumer data purpose is consented by a consumer and recorded by a data collector and expresses how the corresponding collected data can be used. Business purpose is the actual purpose for a business task that involves certain consumer data accesses or operations.

3.1.1. Data purposes. Customer consented data purposes are usually high-level and the number of such purposes is limited. According to the official P3P1.0 Specification [P3P02] released by the World Wide Web Consortium (W3C) on 16 April 2002, there are only 12 purposes[1] defined in P3P1.0. Table 1 shows these 12 purposes.

Table 1. Purposes defined in P3P1.0

Purpose Name	Description
current	Completion and Support of Activity For Which Data Was Provided
admin	Web Site and System Administration
develop	Research and Development
tailoring	One-time Tailoring
pseudo-analysis	Pseudonymous Analysis
pseudo-decision	Pseudonymous Decision
individual-analysis	Individual Analysis
individual-decision	Individual Decision
contact	Contacting Visitors for Marketing of Services or Products
historical	Historical Preservation
telemarketing	Telephone Marketing
other-purpose	Other Uses

3.1.2. Business purposes. Business purposes are defined in each organization according to its business process. They may be defined more specifically than data purposes. For example, the contact purpose may be divided into three categories: phone/fax contact, postal contact, and email contact. However, no matter how business purposes are defined, they must be connected with data purposes. We now introduce a purpose hierarchy to support this.

3.1.3. Purpose hierarchy. The relation between purposes can be modeled with a purpose hierarchy. The purpose relation is a partial ordered relation. A partial order is a reflexive, transitive, and antisymmetric relation. Partial ordered relations support complex purpose hierarchies, such as tree, inverted tree, and lattice structures. We employ the use of a purpose hierarchy to map high-level data purposes to low-level business purposes. If an operation is allowed for a given purpose, it is also allowed for all sub-purposes. Figure 1 illustrates a sample hierarchy for the marketing purpose. In this example, email marketing, postal marketing, and phone/fax marketing are sub-purposes of both direct marketing and third-party marketing.

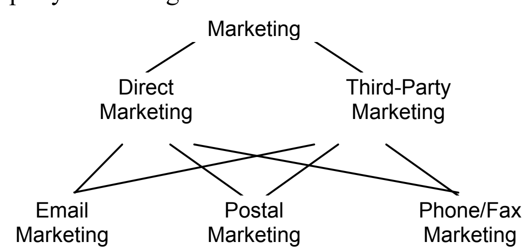

Figure 1. Purpose hierarchy for marketing

Purpose hierarchy allows unambiguous purpose lookup from business purposes to data purposes. The following is an example of an ambiguous purpose lookup. If a customer consents to have his personal information used only for email marketing purpose, the access decision of an operation (i.e. whether the data access request is granted or denied) with the purpose of direct marketing cannot be determined. This is because email marketing belongs to both the direct marketing and third-party marketing purposes. The system cannot determine its exact parent purpose.

The above problem can be solved by placing restrictions on the purpose hierarchy. We only allow business purposes to be mapped to the lowest level of the purpose hierarchy. The purpose for an operation must be defined as specifically as possible. In this way, data purposes are either in the same level as business purposes or in a higher level. This ensures there are no ambiguous purpose lookups from business purposes to data purposes.

3.2. Conditions

A privacy policy may express additional conditions that must be satisfied before a data access request can be granted. For example, one FIP principle is choice/consent, which means the data collector should provide opt-in/opt-out choices for consumers to allow

[1] There is some inconsistency in P3P1.0 specification. In the P3P1.0 XML DTD Definition (Non-Normative), two other purposes are defined: customization and profiling, which are not defined in XML Schema Definition (Normative).

them to decide how their personal information can be used. In the following sample privacy goal extracted from our goal library [AE03], G_{18}: *OPT-OUT from receiving emails from our company*, the access to customer data (e.g. email addresses) must be qualified by the condition Customer.EmailService.Optout = FALSE. In another example, G_6: *PREVENT disclosing PII (Personally Identifiable Information) without consent*, "obtaining consent" is a condition that must be satisfied if an organization wants to disclose PII.

Conditions are not solely for privacy protection. In security enforcement, conditions are usually modeled as authorization constraints [RZF01].

3.3. Obligations

Obligations are actions that must be carried out if a request to access data is granted. For example, in goal, G_{49}: *REQUIRE affiliates to destroy customer data after service are completed*, "destroy customer data" is an obligation for affiliates.

In current website privacy policies, obligations are seldom stated. We have reexamined the 171 privacy requirements taxonomy goals identified from 23 websites' privacy policies during the goal-mining exercises [AE03]. The above example is the only one we identified that involves obligations out of 171 privacy goals.

Obligation-based security policies can be enforced if they can be completely resolved within an atomic execution [RZF01]. However, with respect to the obligations in privacy policies, they are usually not an immediate action as the previous sample policy has shown. In most cases, it is a task or an action that should be executed in the future. Therefore, monitoring and auditing the execution of privacy obligations might be sufficient for obligation enforcement [BJW02].

4. The framework for modeling privacy requirements in role engineering

This section presents the goal-driven framework for modeling privacy requirements in role engineering. The framework includes a context-based data model and a goal-driven role engineering process. The data model expresses how the privacy elements can be modeled in RBAC. The goal-driven role engineering process addresses how privacy elements modeling can be achieved in the role engineering process.

4.1. A context-based data model

The data model models three privacy elements (purposes, conditions, and obligations) as attributes of roles, permissions, and objects, which we name contexts. Figure 2 depicts the data model architecture. We now discuss how these three elements are modeled in our framework.

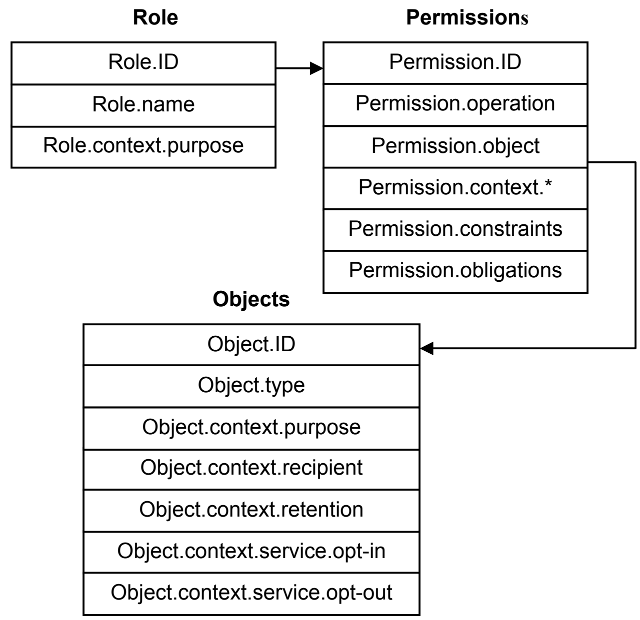

Figure 2. A context-based data model

Business purposes are identified in the role finding/definition process of role engineering. They are mapped as an attribute of roles, which we name *Role.context.purpose*. When a role is derived from a business process or an organization structure, some purposes are implicitly embodied. It is the job of role engineering to elicit and explicitly define these purposes associated with a role. For example, *system administrator* role implies that the purpose of this role is *administration*. From a more accurate and more specific aspect, business purposes not only depend on the role, but also depend on the operation the role intends to perform and the context under which the operation is performed. However, provided that business purposes are usually high-level and the number is limited, as described in Section 3.1, it is acceptable to associate business purposes with roles. In an RBAC model with role hierarchies, the super-role automatically inherits all the purposes associated with its sub-roles. This is different from the purpose relationship in the object model, in which a subtype object inherits all the purposes associated with its supertype object. This is not inconsistent because the purposes associated with roles are business purposes while the purposes associated with objects are data purposes.

Data purposes and other privacy preferences, such as the recipient of data, the retention period of data, etc., are modeled as object attributes in our data model. This work is more appropriate for data management than for role engineering. In this paper, we assume that data are organized into the specified structure. In our framework, object attributes are operands of permission constraints, as we will discuss now.

The conditions of an operation specified in a privacy policy are modeled as permission constraints. Permission constraints are Boolean expressions. The operands of these expressions are attributes of roles, permissions, and objects. The operators of these expressions include standard comparison (i.e. <, >, =, <=, >=, and !=) and logical operators (i.e. Boolean AND, OR, and NOT). To extend the constraint for purpose comparison, we informally define another type of operator for purpose comparison: <<.

Definition: Given two purposes *p1* and *p2*, we claim purpose *p2* contains *p1* (or purpose *p1* belongs to *p2*) if and only if *p2* is on the path from the root of the purpose hierarchy down to *p1* or *p2* is the same as *p1*, which is represented as *p1* << *p2*.

Based on the above definition, the permission constraint to enforce purpose binding is

Role.context.purpose << *Object.context.purpose*

The obligations of an operation are modeled as permission obligations that should be executed afterwards. As we discussed in Section 3.3, obligations in privacy policies are usually not immediate actions, and they are not enforced by the reference monitor. In our framework, we record such obligations so that the reference monitor can send these obligations to another module (e.g. an obligation execution module) for future execution and monitoring.

The proposed context-based data model is inspired from [KKC02], in which Kumar et al. extends RBAC by introducing the notions of role context and context filters. Kumar et al. employs user context and object context to construct a context filter for a role, which is named role context. However, this approach is not suitable for modeling purposes because business purposes are not associated with users or objects. This approach does not consider the context of roles and permissions. Our data model assimilates the basic idea from [KKC02] but goes beyond that in scope. We also take role context and permission context into account. For example, in addition to purpose, a role may have other attributes, e.g. *Role.context.lifetime* defines the life period of a role. This enables our framework to provide fined-gained, context-based access control. Context-based access control not only takes into account the person attempting to access the data and the type of data being accessed, but also the context of the transaction in which the access attempt is made. This is an additional advantage of our data model. The topic related to context-based access control is beyond the scope of this paper.

4.2. A goal-driven role engineering process

We propose a goal-driven role engineering process to demonstrate how the privacy contexts in the above data model can be elicited and modeled. We now discuss the main steps of this process as shown in Figure 3.

The process is comprised of two phases: Role-Permission Analysis (RPA) and Role-Permission Refinement (RPR). These two phases are represented using dotted lines in Figure 3. During the RPA phase, we apply goal- and scenario-oriented requirements analysis techniques to analyzing business process and business tasks. The output of this phase is a collection of role candidates and permission candidates, as well as the corresponding role and permission contexts.

There are several possible input sources: (1) business process description, (2) policy statement (including legislative acts), and (3) requirements specification. The RPA phase starts by identifying tasks. Usually a task is performed to achieve some goals. For example, “schedule meeting” is a task in a meeting scheduler system. The goal to perform this task is to schedule a meeting.

After identifying the task domain, one or more scenarios are authored to model the task details. Every scenario contains a sequence of events, each of which

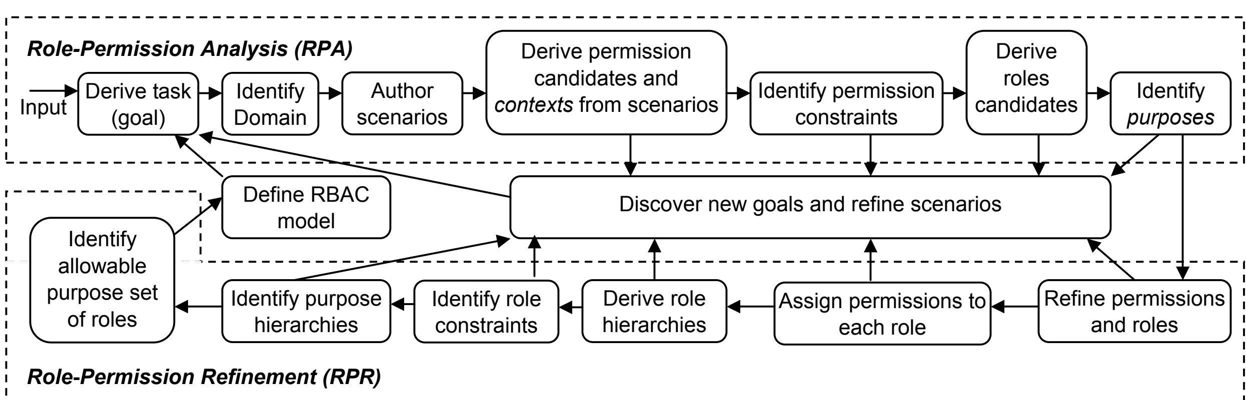

Figure 3. A goal-driven role engineering process for RBAC

may be modeled as an RBAC permission. Permission candidates are then identified. The object and operation are the most important elements of a permission. The next step is to identify permission contexts, the attributes of the permission, and permission constraints, the conditions that must be qualified to execute the permission.

After the permission identification step, role candidates can be identified from the actors of events. A set of permission candidates is associated with each role candidate. When a role is identified, the purpose is also identified and associated with this role.

The RPA phase continues until all module tasks have been identified. At this stage, we have a collection of role candidates and permission candidates, as well as the corresponding role contexts and permission contexts. These outputs are needed for the RPR phase.

It is very possible that the RPA phase does not generate a perfect role and permission set. The roles and permissions identified at this time are probably ambiguous and redundant. They must be refined in the RPR phase according to other factors, such as organization structure, policy statement, etc. As a result of role refinement, role hierarchy is defined and appropriate permissions are assigned to the roles. Finally, after all the purposes are identified, purpose hierarchies are defined and a role's allowable purpose set is identified. The RBAC model is defined thereof.

Although requirements analysis and role engineering analysis are interleaved in the above description, actual practices may not have to follow the exact sequence in Figure 3. Some requirements engineers may find it comfortable to complete requirements analysis first and then conduct role engineering analysis. Our example analysis in Section 5 adopts this scheme.

This process is convenient for modeling privacy requirements because it is easy to model the context of goals and permissions with goal- and scenario-based requirements analysis. A scenario's preconditions express possible permission constraints. The postconditions are possible obligations. The goal identified in this process is the possible purpose of the task and the possible purpose associated with a role. However, the RPR phase does not depend on the goal- and scenario-based requirements analysis. Other heuristics must be provided to facilitate role/permission refinement and the definition of role hierarchies.

The process shown in Figure 3 is simplified from a more complete life-cycle goal-driven role engineering process, which we are currently developing [He03b].

5. A healthcare example

This section presents an example analysis of a HIPAA scenario using our Scenario Management and Requirements Tool (SMaRT) [SMaRT03]. SMaRT is a web-based tool that supports scenario- and goal-based requirements analysis. It has been successfully applied in several case studies [AA03]. Because SMaRT does not currently support role engineering analysis, the derivation of RBAC elements was documented using a spreadsheet. We plan to extend SMaRT to support the proposed goal-driven role engineering process.

Consider the healthcare scenario below that is readily available in [HIP03]:

A patient, Mr. Stalwart, is brought to a hospital's Emergency Department (ED). He is unresponsive with a gunshot wound (GW) to the abdomen. Upon his arrival, Dr. Goodcare examines the patient, and begins resuscitative efforts.

First, the ward secretary (WS) registers Mr. Stalwart into the ED system. According to HIPAA security regulations, four security and privacy requirements apply to this task:

- *The secretary needs to have been trained in privacy and security.*
- *The hospital must document this training.*
- *The ward secretary needs to have been authenticated by the system, and his/her authority to perform the registration task confirmed (RBAC).*
- *The system should maintain an audit trail of information viewed and modified.*

The result of our scenario analysis is shown in Figure 4. The elements that appear above the line in Figure 4 correspond to the RE activities whereas the elements that appear below the line correspond to the role engineering activities. We now walk through the goal-driven role engineering process with the scenario.

We first conduct the goal-based requirements analysis process. From the task description, we identify the task domain is *ED Patient Info Management*, and the goal of this task is to *register patient into the ED system*. Then we author a scenario to model the task. To model a complex task, more than one scenario may be needed. A sequence of events is elicited to illustrate the scenario. An event includes an actor and an action. A collection of actors and actions are then identified. The preconditions are identified by asking what conditions must be satisfied to perform this task. The postconditions are identified by asking what are the results of the task, and what are the obligations if the task is performed. The information about the registration process may be obtained via interview with stakeholders or from existing job description manuals.

Based on the requirements analysis, we can then conduct the role engineering analysis. First, we map the actions to permission candidates and identify permission constraints from preconditions. We also identify permission obligations from postconditions, if there are any. After that, we identify role candidates and the purposes of the task. We associate this purpose with the role and model it as a role context. The roles are then associated with appropriate permissions. These are the major steps in the RPA phase.

[Goal] Register patient into the ED system
[Domain] ED Patient Info Management
[Scenario] Ward secretary registers patient into the ED system
[Actors] Ward secretary
System
[Actions] Invoke patient registration procedure
Request PHI (Protected Health Information)
Enter PHI
Submit PHI
Save PHI
Confirm PHI saved
Generate audit trail
[Events] Ward secretary invokes patient registration procedure
System requests PHI
Ward secretary enters PHI
Ward secretary submits PHI
System saves PHI
System confirms PHI saved
System generates audit trail
[Preconditions] Ward secretary authenticated
Ward secretary trained in privacy and security
Hospital security and privacy training process documented
[Postconditions] Registration audit trail generated
Patient registered in the ED system

[Permissions] P1: can invoke patient registration procedure
P2: can enter PHI
P3: can submit PHI
P4: can request PHI
P5: can save PHI
P6: can confirm PHI saved
P7: can generate audit trail
[Permission Context] No permission context identified
[Permission Constraints] user.training = T AND user.training.documenting = T
[Permission obligations] No permission obligations identified
[Roles] Ward Secretary (WS)
System (S)
[Role Context] WS.purpose = patient registration
[Role Permission Assignment] WS (P1, P2, P3)
S (P4, P5, P6, P7)
[Allowable Purpose Set] APS (WS) = {patient registration}

Figure 4. A healthcare example

Because we are only analyzing a single task, this example does not have a collection of roles/permissions nor does it include a role hierarchy, role constraints or purpose hierarchy. Hence, the RPR phase is outside the scope of this example. However, we have specified *patient registration* as one of the allowable purposes for role *WS*. Although we have only elaborated one scenario, other plausible scenarios would typically be identified and elaborated as well. For example, *Dr. Goodcare requests patient record* and *Ward Secretary updates patient status*.

Because the system is an agent that performs some tasks, we also model System as a role in the example. Generally speaking, we only model the permissions and roles from a user's perspective. The system's permissions are built into the implementation program. Note that the derived permissions may depend on the implementation. If the system is designed so that whoever can invoke the patient registration procedure has full control of everything in the procedure, then the three permissions assigned to role *WS* can be merged into one: *can invoke patient registration procedure*.

The above example analysis is only a proof-of-concept evaluation of the framework. We are validating the approach in the specification of Transnational Digital Government (TDG) project for Belize and Dominican Republic [Cav03]. This study will allow us to evaluate the effectiveness, scalability as well as suitability of our framework for integration with other RE methodologies.

6. Conclusions and future work

Privacy enforcement is important for many commercial software systems. Modeling privacy requirements in the early stages of system development is essential for privacy enforcement and ensuring quality in software systems used in environments that pose risks of loss as a consequence. This paper presents a framework for modeling privacy requirements in role engineering. Basic privacy requirements such as purpose binding can be modeled as permission constraints. Privacy preferences, such as opt-in/opt-out choices, data recipient, etc., can also be modeled using the context-based data model. The framework provides a basis for enforcing privacy requirements with RBAC.

Our framework demonstrates that RE can bridge the gap between high-level privacy requirements and low-level access control policies. Requirements engineers can elicit and model privacy requirements as RBAC entity contexts and constraints by analyzing business processes and privacy policies using the goal-driven role engineering process. Privacy officers can then define privacy authorization rules based upon the context-based data model. These rules are similar to the access control rules derived from security policies and they are enforced via RBAC.

Our framework also demonstrates that RE can bridge the gap between competing stakeholders' security and privacy requirements, i.e., companies' privacy practices may be in conflict with user preferences. The approach presented in this paper allows both perspectives to be modeled (e.g. business purposes and data purposes) and tradeoffs to be analyzed.

Our role-engineering process is a top-down approach; we derive roles and permissions based on business process analysis. Industry experiences report role analysis should ideally be a mixed bottom-up and top-down approach [Sch00, KKS02]. Our framework can be used with other bottom-up approaches to achieve best result.

Although our work is preliminary, early validation in the TDG project [Cav03] suggests that we will be able to address some of the following limitations in the future.

One limitation of the goal-driven role engineering process is that it is only effective in deriving functional roles/permissions in RBAC. Unfortunately, goals and scenarios are difficult to derive permissions that result from the chosen technology instead of functionality, for example, internal web server functions for a web-based application [NS02].

Our framework can model purpose binding but cannot directly model another privacy requirement, the principle of necessity. The principle of necessity can be enforced by RBAC if each task is granted a minimum set of permissions and users are allowed to perform one current task at the same time [Fis01]. Therefore, it is possible to support this requirement with our context-based data model by expressing tasks as permission context. We plan to support this in the future.

Recall in our example four HIPAA security and privacy requirements were identified from a policy statement. However, our framework does not address how to extract corresponding security and privacy requirements from existing legislative acts and organizational policies. We plan to develop techniques to elicit such requirements and associate them with the tasks we are modeling. Modal-Action Logic (MAL) [GF91] is one promising technique that we are exploring.

The goal-driven role engineering process described in Section 5 is high-level. Only the RPA phase is elaborated in this paper. We are developing detailed heuristics to elicit and refine roles, permissions, and role hierarchies. We also plan to integrate the proposed role engineering process into SMaRT to provide tool support.

Acknowledgements

This work is partially funded by NSF ITR Grant #0113792. The authors wish to thank Dr. Peng Ning for discussions concerning security and privacy protection and William Stufflebeam for helpful comments.

References

[AA03] T. A. Alspaugh and A. I. Antón. Contrasting Use Case, Goal, and Scenario Analysis of the Euronet System, *Submitted to the 11th IEEE International Requirements Engineering Conference (RE'03)*, 2003.

[AE01] A. I. Antón and J. B. Earp. Strategies for Developing Policies and Requirements for Secure Electronic Commerce Systems, In *E-Commerce Security and Privacy*, edited by A. K. Ghosh, Kluwer Academic Publishers, pp. 29-46, 2001.

[AE03] A. I. Antón and J. B. Earp. A Requirements Taxonomy to Reduce Website Privacy Vulnerabilities, To Appear: *Requirements Engineering Journal*, Springer-Verlag, 2003.

[AEP01] A. I. Antón, J. B. Earp, C. Potts, and T. A. Alspaugh. The role of Privacy and Privacy Values in Requirements Engineering, *IEEE 5th International Symposium on Requirements Engineering (RE'01)*, pp. 138-145, 2001.

[AEC02] A. I. Antón, J. B. Earp and R. A. Carter. Aligning Software Requirements with Security and Privacy Policies, *Proc. of International Workshop on Requirements Engineering for Software Quality (REFSQ'02),* 2002.

[AER02] A. I. Antón, J. B. Earp and A. Reese. Analyzing Web Site Privacy Requirements Using a Privacy Goal Taxonomy, *Proc. of the 10th Anniversary IEEE Joint Requirements Engineering Conference (RE'02)*, Sep. 2002.

[AMP94] A. I. Anón, W. M. McCracken and C. Potts. Goal Decomposition and Scenario Analysis in Business Process Reengineering, *Proc. of the 6th International Conference on Advanced Information Systems Engineering (CAiSE''94)*, Utrecht, The Netherlands, pp. 94-104, 6-10 June 1994.

[Ant96] A. I. Antón. Goal-Based Requirements Analysis, *Proc. of the 2nd IEEE International Conference on Requirements Engineering (RE'96)*, pp. 136-144, April 1996.

[AP98] A. I. Antón and C. Potts. The Use of Goals to Surface Requirements for Evolving Systems, *Proc. of the 1998 International Conference on Software Engineering (ICSE'98)*, pp. 157-166, Kyoto, Japan, ACM, April 1998.

[AS00] G.-J. Ahn and R. Sandhu. Role-Based Authorization Constraints Specification, *ACM Transaction on Information and Systems Security*, Vol. 3 (4), pp. 207-226, Nov. 2000.

[BJW02] C. Bettini, S. Jajodia, X. Wang, and D. Wijesekera. Obligation Monitoring in Policy Management. *Proc. of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY 2002)*, IEEE, 2002.

[Cav03] V. Cavalli-Sforza et al. Enabling Transnational Collection, Notification, and Sharing of Information, Accepted, to appear in *Proc. of the 2003 National Conference on Digital Government Research*, May 2003.

[CIN02] R. Crook, D. Ince, and B. Nuseibeh. Towards an Analytical Role Modelling Framework for Security Requirements, *Proc. of the 8th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02)*, Essen, Germany, 2002.

[Coc97] A. Cockburn. Structuring Use Cases with Goals, *Journal of Object-Oriented Programming*, Vol. 10 (5), pp. 56-62, 1997.

[Coy96] E. J. Coyne. Role Engineering, *Proc. of the 1st ACM Workshop on Role-Based Access Control (RBAC'96)*, Gaithersburg, MD, 1996.

[DD82] D. E. Denning and P. J. Denning, *Cryptography and Data Security*, Addison-Wesley, 1982.

[Eli02] *Eli Lilly Settles FTC Charges Concerning Security Breach*, Federal Trade Commission, http://www.ftc.gov/opa/2002/01/elililly.htm, January 2002.

[Eps02] P. A. Epstein. Engineering of Role/Permission Assignments, *Ph.D. Dissertation*, School of Information Technology and Engineering, George Mason University, Fairfax, VA, 2002.

[ES01] P. Epstein and R. Sandhu. Engineering of Role/Permission Assignments, *Proc. of the 17th Annual*

Computer Security Applications Conference (ACSAC 2001), pp. 127-136, IEEE, 2001.

[ES99] P. Epstein and R. Sandhu. Towards A UML Based Approach to Role Engineering, *Proc. of the 4th ACM Workshop on Role-Based Access Control (RBAC'99)*, pp. 135-143, 1999.

[FH97] E. B. Fernandez and J. C. Hawkins. Determining Role Rights from Use Cases, *Proc. of the 2nd ACM Workshop on Role-Based Access Control (RBAC'97)*, pp. 121-125, 1997.

[FIP98] Fair Information Practice Principles, Privacy Online: A Report to Congress (Part III), FTC, http://www.ftc.gov/reports/privacy3/fairinfo.htm, June 1998.

[Fis01] S. Fischer-Hübner. IT-Security and Privacy, *Lecture Notes in Computer Science 1958 (LNCS 1958)*, Springer-Verlag, 2001.

[FSG01] D. F. Ferraiolo, R. Sandhu, S. Gavrila, et al. Proposed NIST Standard for Role-Based Access Control, *ACM Transactions on Information and System Security*, Vol. 4 (3), pp. 224-274, August 2001.

[GF91] S. J. Goldsack and A. C. W. Finkelstein. Requirements Engineering for Real-Time Systems, *Software Engineering Journal,* Vol. 6 (3), pp. 101-115, May 1991.

[GHS00] J. Goldman, Z. Hudson and R. Smith. *Privacy: Report on the Privacy Policies and Practices of Health Web Sites*, California HealthCare Foundation, January 2000 http://www.chcf.org/topics/view.cfm?itemID=12497.

[GLB01] *Gramm-Leach-Bliley Act: Financial Privacy and Pretexting*, Federal Trade Commission, http://www.ftc.gov/privacy/glbact/index.html.

[He03a] Q. He. Privacy Enforcement with an Extended Role-Based Access Control Model, *NCSU Computer Science Technical Report*, TR-2003-09, 2003.

[He03b] Q. He. A Goal-driven Role Engineering Process for Privacy-Aware RBAC Systems, *Submitted to the 11th IEEE International Requirements Engineering Conference (RE'03) Doctoral Symposium*, 2003.

[HIP96] *The 1996 Health Insurance Portability and Accountability Act (HIPAA)*, HEP-C ALERT, http://www.hep-c-alert.org/links/hippa.html.

[HIP03] *Case Study - How HIPAA affects a patient visit*, NPower NY, http://www.npowerny.org/case+study+6.pdf, 2003.

[Kai00] H. Kaindl. A Design Process Based on a Model Combining Scenarios with Goals and Functions, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 30 (5), pp. 537-551, Sep. 2000.

[Kai95] H. Kaindl. An Integration of Scenarios with their Purposes in Task Modeling, *Proc. of the 1995 Symposium on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS'95)*, pp. 227-235, ACM, Aug. 1995.

[Kav02] E. Kavakli. Goal-Oriented Requirements Engineering: A Unifying Framework, *Requirement Engineering Journal*, Vol. 6 (4), pp. 237-251, 2002.

[KKC02] A. Kumar, N. Karnik, and G. Chafle. Context Sensitivity in Role-based Access Control, *ACM SIGOPS Operating Systems Review*, pp. 53-66, July, 2002.

[KKS02] A. Kern, M. Kuhlmann, A. Schaad, and J. Moffett. Observations on the Role Life-Cycle in the Context of Enterprise Security Management, *Proc. of the 7th ACM Symposium on Access Control Models and Technologies (SACMAT'02)*, pp. 43-51, 2002.

[KS02] G. Karjoth and M. Schunter. A Privacy Policy Model for Enterprises, *Proc. of the 15th IEEE Computer Security Foundations Workshop*, pp. 271-281, IEEE, 2002.

[Lam01] A. van Lamsweerde. Goal-Oriented Requirements Engineering: A Guided Tour, *Proc. of the 5th International Symposium on Requirements Engineering (RE'01)*, pp. 249-262, IEEE, 2001.

[NS02] G. Neumann and M. Strembeck. A Scenario-driven Role Engineering Process for Functional RBAC Roles, *Proc. of the 7th ACM Symposium on Access Control Models and Technologies (SACMAT'02)*, pp. 33-42, 2002.

[OEC80] *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*. Organization of Economic Cooperation and Development (OECD), 1980, http://www1.oecd.org/publications/e-book/9302011E.PDF.

[OSM00] S. Osborn, R. Sandhu and Q. Munawer. Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies, *ACM Transactions on Information and System Security*, Vol. 3 (2), pp. 85-106, May 2000.

[P3P02] *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*, The World Wide Web Consortium, April 16, 2002, http://www.w3.org/p3p/.

[RSA98] C. Rolland, C. Souveyet, and C. B. Achour. Guiding goal modeling using scenarios, *IEEE Transactions on Software Engineering*, Vol. 24 (12), pp. 1055-1071, 1998.

[RSW00] H. Roeckle, G. Schimpf, and R. Weidinger. Process-Oriented Approach for Role-Finding to Implement Role-Based Security Administration in a Large Industrial Organization, *Proc. of the 5th ACM Workshop on Role-Based Access Control (RBAC'00)*, pp. 103-110, 2000.

[RZF01] C. N. Ribeiro, A. Zuquete, P. Ferreira, and P. Guedes. SPL: An Access Control Language for Security Policies with Complex Constraints, *Proc. of Network and Distributed System Security Symposium (NDSS'01)*, pp. 89-107, 2001.

[SCF96] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman. Role-Based Access Control Models, *IEEE Computer*, Vol. 29 (2), pp. 38–47, Feb. 1996.

[Sch00] G. Schimpf. Role-Engineering Critical Success Factors for Enterprise Security Administration, *Proc. of the 16th Annual Computer Security Applications Conference (ACSAC'00)*, 2000.

[SMaRT03] Scenario Management and Requirements Tool. http://tigger.csc.ncsu.edu/~smart/

[Toy00] *FTC says Toysmart violated child Net privacy law*, Federal Trade Commission, http://www.ftc.gov/opa/2000/07/toysmart.htm, 2000.

[ZAC02] L. Zhang, G.-J. Ahn, B.-T. Chu. A Role-Based Delegation Framework for Healthcare Information Systems, *Proc. of the 7th ACM Symposium on Access Control Models and Technologies (SACMAT'02)*, pp. 125-134, 2002.