A Framework for Privacy-Enhanced Access Control Analysis in Requirements Engineering

Qingfeng He
North Carolina State University
900 Main Campus Dr, 165C-197
Raleigh, NC, USA
+1.919.513.5082
qhe2@eos.ncsu.edu

Annie I. Antón
North Carolina State University
900 Main Campus Dr, 165C-199
Raleigh, NC, USA
+1.919.513.5764
aianton@eos.ncsu.edu

ABSTRACT
Access control is a major security mechanism for achieving confidentiality and integrity in software systems. Specifying access control policies is a tedious and error-prone process and needs requirements-level analysis support. Given that there is no systematic method in requirements engineering (RE) for access control analysis, we present a comprehensive set of criteria to support this kind of analysis. We survey several existing RE approaches and compare their ability to support access control analysis. We present an analytical framework that guides the analysis of: data, goal/scenario-based tasks, organizational structures, and information flows. Our framework has at least two advantages. First, unlike other RE methodologies, it provides systematic support for access control analysis. Second, it supports analysis of privacy-enhanced features in access control. We employ a healthcare example to illustrate how to apply the framework.

Categories and Subject Descriptors
D.2.1 [**Requirements/Specifications**]: Methodologies – *goals, scenarios, abstraction, data types;* D.4.6 [**Operating Systems**]: Security and Protection – *access controls, verification*

General Terms
Documentation, Design, Security, Theory, Verification

Keywords
Access control analysis, requirements engineering

1. INTRODUCTION
Access control is one of the major security mechanisms to achieve confidentiality and integrity in software systems. Confidentiality means that information is not disclosed to unauthorized persons, processes or devices. Integrity means that unauthorized persons, processes or devices cannot modify information. Additionally, access control is important for protecting data privacy. Although privacy means different things to different people within various contexts, in general, privacy means protecting personal information from being accessed, modified or disclosed to unauthorized persons without consent. Privacy has become an increasingly important issue and has recently received attention from consumers, government officials, legislators, and software developers. These concerns stem from increasing personal information collection, non-consented information disclosures and intra/inter-organizational information transfer. Privacy poses new challenges to access control. For example, the principal of purpose binding (which means personal data collected for one purpose should not be used for another purpose [19]) is not currently supported by traditional access control mechanisms.

Access control analysis entails analyzing business tasks and organizational structures to specify access control policies. Defining and deploying complex access control policies is a tedious and error-prone process because a complex software system could have many users performing various tasks and many resources that need to be protected via access control [33]. Currently, most policy specification occurs at the deployment level as part of security administration [9]. Thus, this process is isolated from requirements analysis and may result in policies and requirements are not compliant with one another. Access control policies are derived from system requirements. Prescriptive guidance is needed to aid in this specification process. In this paper we discuss how RE can help.

Researchers are recognizing the need to bridge the gap between requirements modeling and complex access control policy specification [9]. Existing RE approaches, such as KAOS [15], *i** [37, 38], NFR [26, 10] and the analytical role modeling framework [9], provide limited support as we discuss herein. In this paper, we propose a Requirements-level Access Control Analysis Framework (RACAF) to support systematic access control analysis. This late-phase RE activity helps specify access control policies and serves as a bridge between requirements analysis and design activities.

The rest of the paper is structured as follows. Section 2 presents a comprehensive set of criteria to guide access control analysis. Section 3 surveys the strengths and limitations of several existing RE approaches in supporting access control analysis. Section 4 presents the RACAF. In Section 5, a healthcare example is employed to illustrate application of this framework. Finally, Section 6 summarizes the paper and discusses our plans for future work.

2. ACCESS CONTROL ANALYSIS CRITERIA

In order to adequately address access control during early-phase system development, we first identify the key elements and aspects of access control expressed as 14 criteria to guide access control analysis.

The 14 criteria presented herein are based on the nature of access control in security and privacy protection. These criteria have the following characteristics: First, they cover not only security protection, but privacy protection as well. Second, they are comprehensive in that they cover the important aspects of access control. Independently, each, or subsets of these criteria have been addressed by others [9, 17]. However, to the best of our knowledge, they have not been addressed collectively. Third, the criteria are intended for analyzing access control in data processing software systems, not for access control in security kernels (e.g., access control in operating systems).

It is important to address these criteria in RE because access control policies are basically security and privacy requirements that restrict access to valuable resources and data. Requirements-level analysis provides the rationale to justify the specification of access control policies.

Access control analysis requires the ability to meet the following 14 criteria:

(1) model the purpose of tasks;
(2) model contexts and constraints;
(3) model permissions and obligations;
(4) model information flow;
(5) model various data types;
(6) model users' privacy preferences;
(7) model actor relationships;
(8) model organizational hierarchies;
(9) define roles;
(10) model necessity (least privilege);
(11) model separation of duties;
(12) model delegations;
(13) support formal reasoning; and
(14) facilitate the automation of access control policy specifications.

A basic access control policy rule should contain at least three elements: the subject, the object and the permission (or action or operation) that the subject is allowed to perform on the object. For example, <Jack, all emails in his email account, read> is an access control policy, which means Jack has the read permission to all the emails in his email account. Most of the above criteria are based on modeling these three elements or their extensions, as we discuss herein.

Criterion (1) concerns an important privacy element: purpose. In the privacy domain, the purpose of data usage is very sensitive. For example, personal data may be usable for medical treatment, but not for pharmaceutical marketing. This is the purpose binding principal in privacy protection. Access control authorizations must consider the purpose of a task or an operation [21, 22, 30].

Criterion (2) concerns the context of an action/operation. For example, in the healthcare domain, authorization decisions may be made depending on the location of a request (e.g., emergency room) or the time of a request, etc. [4].

Criterion (3) concerns permissions and obligations [5]. Permissions are basic elements of access control policies that specify actions an agent is allowed to perform. Obligations must be fulfilled if a request to access an object is granted. For example, in an access control policy we may specify "destroy customer data in 30 days after the service or transaction is completed" as an obligation that must be fulfilled if a request to access customer data is granted.

Criterion (4) concerns information flow. For example, many consumers are very concerned about the sharing and disclosure of their personal data without notice or consent as in the recent JetBlue privacy policy violation case [3]. The ability to model information flows across organizational boundaries is imperative [3].

Criterion (5) concerns the object of an access control policy rule. Data are usually organized according to data types. The same type of data is often handled in the similar ways [21, 22].

Criterion (6) concerns a special type constraint for access control, privacy preferences. Within the privacy protection context, it is very important to model and express user's privacy preferences as policies attached to the data items throughout various courses of data processing. This is referred to as the sticky policy paradigm [22]. These preferences serve as authorization constraints in access control.

Criterion (7), (8) and (9) concern the subject of an access control policy rule within an organizational context. Role-based access control (RBAC) [31] is widely used in many systems, such as Oracle 9i database management systems, information systems [33]. To model roles and role hierarchies, it is helpful to first model organizational structures [27, 9] and actor relationships [4]. The ability to define roles and assign appropriate permissions to these roles, role engineering [11], is a major challenge in RBAC because of complex organizational hierarchies, many jobs and positions, and the potential for millions of permissions in a system [32, 33].

Criterion (10) and (11) concern two security properties: least privilege and separation of duties [31]. The principal of least privilege means a subject should only be given the minimum set of permissions that are necessary to perform a task. The separation of duties principal means some permissions are considered mutually exclusive, that is any user should not be given two of these permissions. For example, an accounting clerk and an account manager could be defined as mutually exclusive. If a user is assigned both roles, he/she might exploit the permissions of both roles to perform some illegal tasks.

Criterion (12) concerns a widely used security mechanism: role or permission delegation [7, 39]. For example, when a user is on leave, he may delegate some of his permissions to another user and later revoke that delegation upon his return from leave.

Criterion (13) concerns the formal analysis support. It is important to support formal reasoning to ensure we can verify the system's security properties as well as the access control model [18].

Criterion (14) concerns the ability to automate access control policy specification. This criterion is important because automation can greatly reduce human effort and errors.

3. A SURVEY OF EXISTING RE APPROACHES

This section summarizes four RE approaches and compares the ability of each to support access control analysis for modeling security and privacy requirements.

3.1 The KAOS Framework

The KAOS framework is a goal-based requirements acquisition and elaboration method [15, 24, 14, 23]. KAOS provides a formal and expressive conceptual modeling language, rich requirements elaboration strategies and tool support to help requirements engineers specify requirements derived from high-level goals.

KAOS defines a rich set of meta-concepts and meta-relationships. Some of these meta-concepts (e.g., object, agent, action) are basic elements of an access control policy. The KAOS framework thus provides a natural foundation for supporting access control analysis. Fontaine employs KAOS to refine security requirements into specific authorization rules and access control policies expressed in Ponder [17]. Ponder is a language for specifying management and security policies for distributed systems [13]. Fontaine's work is an important step towards requirements-level access control analysis for security policy specification. However, it is limited in that only two kinds of policies in Ponder can be mapped from KAOS specifications: authorization and obligation policies. Refrain and delegation policies in Ponder cannot be mapped from KAOS specifications because KAOS does not provide support to analyze these two types of policies.

3.2 The *i** Framework

The *i** framework is an early-phase RE method used to model and reason about organizational contexts and rationales [37, 10]. It was initially developed to provide support in modeling, analyzing and redesigning organizations and business processes [10], but has recently been used to model trust [36] as well as security and privacy requirements [25, 35].

The *i** framework is now also being used for access control analysis. Liu et al. applied the *i** framework to model the dependencies among actors, tasks and resources of a system, thus helping analysts understand their relationships [25]. However this approach does have its limitations. The approach assumes that the roles and permissions have been previously derived, providing no prescriptive guidance as to how roles and permissions are identified or derived, from where they originate, how permissions are assigned to these roles, how mutual exclusive permissions are defined, etc. These topics remain major challenges in access control analysis during late-phase RE. Additionally, it is difficult to model context and constraint information in the *i** framework.

3.3 The NFR Framework

The NFR (Non-Functional Requirement) framework is a goal-based requirements analysis method that systematically addresses non-functional requirements in the early stages of system development [26, 10]. The NFR framework represents non-functional requirements as *softgoals* that are *satisficed*, which means they are satisfied within acceptable limits instead of absolutely being accomplished.

Security requirements are non-functional requirements that can be analyzed using the NFR framework [10]. Basically, security requirements address confidentiality, integrity and availability. These requirements are operationalized into alternative security mechanisms (e.g., password authentication, encryption) and functional requirements to achieve the specific softgoals (e.g., confidentiality, accountability). Alternatives are evaluated according to design rationales and goal dependencies with functional requirements.

The objective of NFR is to provide a systematic method to analyze security requirements and make a variety of alternative security methods and their tradeoffs available to system stakeholders. By evaluating the design decisions, the framework may help provide a system design that can best achieve security requirements (and other non-functional requirements). Access control analysis of NFR is high-level in that access control is treated as an alternative to achieve softgoal confidentiality. There is no discussion in [10] about access control policies or modeling access control elements.

3.4 The Analytical Role Modeling Framework

Crook et al. proposed an analytical role modeling framework to model access control policies [9]. The framework is specifically designed for role-based access control (RBAC) and derives roles from organizational structures. Although other researchers have employed RE methods, such as scenarios [28] and use cases [16], to define needed permissions for roles, this framework was the first to explicitly clarify the importance of providing requirements-level support for modeling access control policies.

The ARMF (Analytical Role Modeling Framework) has two important contributions. First, the rationale to derive roles based on organizational structures is very useful. Job positions in an organization can be mapped to roles in RBAC. Organizational and seniority hierarchies in an organization can be mapped to the role hierarchies of RBAC. Thus, deriving roles from organizational structures facilitates the user assignment and authorization management process of access control. Second, this framework clarifies the need to model access control policies in requirements analysis.

3.5 Comparison of Existing RE Approaches

We now examine the extent to which each of the previously discussed RE approaches support access control analysis based on the 14 criteria presented in Section 2. Our assessment is qualitative and based our application of the approaches to evaluate the level of support each provides. We classify three levels of support:

(1) Yes: the method provides direct support for the "ability" expressed in the criteria;
(2) Partial: the method does not provide direct support, but could be extended without modifying the underlying concepts; or
(3) No: the method cannot support the "ability" without fundamental modification.

Table 1 shows the assessment results. For example, given the first criteria, KAOS, *i** and NFR are all goal-oriented methods. Because goals are conceptually similar to purposes, these three methods can be effectively used to model purposes. To date, however, there is no evidence that either of these three methods have been used to model purposes.

Table 1. Comparison of four RE approaches

Criteria	KAOS	*i**	NFR	ARMF
Purposes	Yes	Yes	Yes	No
Contexts & Constraints	Partial	No	No	Yes
Permissions & obligations	Partial	Partial	No	Partial
Information flow	No	No	No	No
Data types	Partial	No	No	Yes
Privacy Preferences	Partial	No	No	Partial
Actor relationships	Partial	Partial	No	Partial
Organizational structures	Partial	No	No	Yes
Roles	No	No	No	Yes
Least privilege	Partial	Partial	No	No
Separation of duties	Partial	Partial	No	No
Delegation	No	No	No	No
Formal reasoning	Yes	No	No	No
Support automation	Partial	No	No	Partial

As shown in Table 1, the NFR framework is least effective for use in access control analysis. The NFR framework primarily addresses operational security or high-level operational security goals [8] Additionally, because the NFR framework is a qualitative reasoning approach, it is hard to verify a system's security properties. Although NFR is very effective in systematically dealing with non-functional requirements at a high-level, it is not suitable for access control analysis, which is a late-phase RE or design level activity that requires formal verification of security properties (e.g., least privilege, separation of duties, etc.).

Although the *i** framework provides effective support for modeling dependencies and reasoning about rationale, it is also fundamentally unsuitable for access control analysis. Nine of fourteen assessments reveal criteria abilities that it cannot support and four others are only partially supported. The *i** framework is an early-phase RE methodology, and our analysis reveals the need for an approach that better supports late-phase RE activities such as access control analysis.

The strength of the ARMF is its ability to derive roles based on models of organizational hierarchies. However, it is not a complete requirements analysis methodology — ARMF provides no guidance for how to derive permissions, discover contexts and constraints, and assign permissions to roles. Unlike the other three methods, which have been refined for a long time (all over ten years), the ARMF was first introduced in 2002 and has yet to be evaluated within the context of a real system.

The KAOS framework is the best approach among these four methods in terms of modeling access control. However, it does have its limitations. Because KAOS is not specifically designed for access control analysis, its concepts are defined in general terms. To support a majority of the abilities expressed in our criteria (nine of fourteen), KAOS needs to be extended and the effort to do so could be rather large.

In conclusion, none of the surveyed RE approaches provides a systematic methodology to comprehensively support access control analysis.

4. RACAF

We now present a Requirements-level Access Control Analysis Framework (RACAF) that builds upon existing RE approaches but which goes beyond those approaches by providing a systematic access control analysis framework.

Four types of analysis are provided in RACAF as shown in Figure 1. Each type of analysis addresses several aspects of access control according to their abstraction level. From the bottom level to the top is data analysis, goal/scenario-based task analysis, organizational structure analysis, and information flow analysis. Note in Figure 1, information flow analysis is shaded. This is because the other three types of analysis are required to specify access control policies, whereas information flow analysis does not directly support access control policy specification and is not required. Criterion 1-12 are divided into four groups, which are modeled by these four types of analysis, respectively. Similar to KAOS, we plan to use formal specifications to express analysis results to ensure that RACAF supports formal reasoning (criterion 13). We are also developing tool support for RACAF to facilitate automation of access control specifications (criterion 14). We now discuss these four types of analysis in detail.

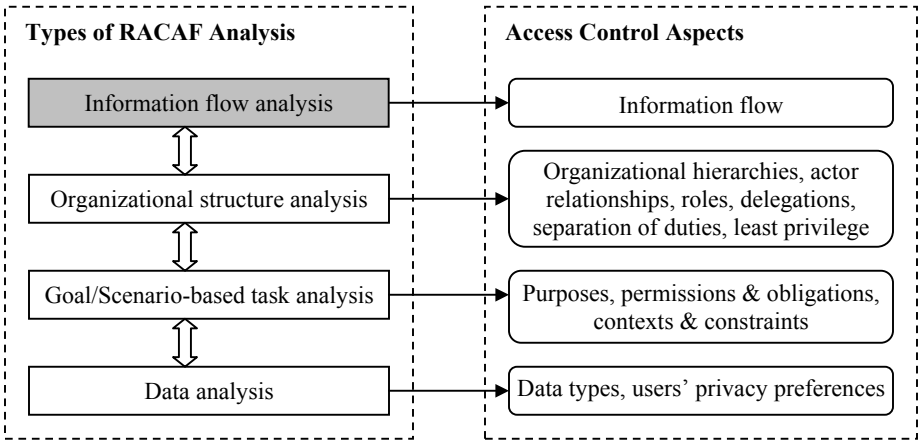

Figure 1. RACAF and access control aspects

4.1 Data Analysis

The ultimate goal of access control is to protect data. The objective of data analysis is thus to identify the resources to be protected and the preferences specified by data subjects about

how to handle the data. Data analysis needs to address the following questions:

- What data is or needs to be collected by the system?
- What privacy preferences are associated with this data?
- What is the data's type?
- What data needs to be protected by access control?

The first question addresses data collection for privacy protection. An important privacy principal is minimum collection, which means the amount of data collected by a system should be limited to that which is necessary to perform the corresponding transaction or provide the requested service.

For the second question, we need to model users' privacy preferences from at least four perspectives: <purpose, recipient, retention, consent> [29], which express the purpose for which data is collected, the recipient of the data, how long the data will be kept in the system, and whether the user's consent is required/obtained. These preferences serve as authorization constraints for access control policies.

Data are grouped together according to type. Sometimes the same type of data follows the same policy. For example, personal contact information, financial information, medical information, demographic information, etc., are data types, which may be processed according to the same processing rule.

The last question identifies existing data in the system that needs to be protected via access control.

Various object-oriented analysis (OOA) methods [6, 12] can be used to assist data analysis, such as abstraction, hierarchy, typing and classification.

4.2 Goal/Scenario-Based Task Analysis

As discussed in Section 3, most RE approaches support goal based task analysis.

In RACAF, we apply goal/scenario-based requirements analysis techniques to analyze tasks to derive purposes, permissions and obligations, contexts and obligations. Goals are the objectives of a task, a business process or a system. The nature of a goal makes it an intuitive way to elicit and model purpose, an important element in a privacy-aware system. Scenarios present possible ways for actors to interact with a system to perform some task or accomplish some desired function [34]. Scenarios are concrete, narrative, and procedural. They describe real situations using examples and illustrations. A scenario is usually associated with a sequence of events, which include actors and actions, pre-conditions and post-conditions, obstacles, requirements, goals, etc. [1]. We model actors as the subjects, actions as the permissions, pre-conditions as contexts and constraints, and post-conditions as obligations of an access control policy. This mapping is shown in Table 2.

Table 2. Mapping from scenario elements to access control policy elements

Scenario elements	Access control policy elements
Goals	Purposes
Actors	Subjects
Actions	Permissions
Pre-conditions	Contexts and constraints
Post-conditions	Obligations

We provide an example scenario and show how these elements are mapped in Section 5.3. Detailed discussion on how to derive permissions and obligations, contexts and constraints, etc., using a goal/scenario-based requirements analysis method can be found in [19].

4.3 Organizational Structure Analysis

The previous two types of analysis produce raw data (e.g., permissions, obligations), which is insufficient to specify access control policies. In fact, users are seldom directly assigned permissions to perform some task. Associating users directly with permissions will make authorization management difficult because there could be thousands of permissions in the system. Thus, this data needs to be abstracted to high-level concepts, such as roles, by performing organizational structure analysis.

The main activities at this level include actor relationship analysis, organizational hierarchy analysis, delegation analysis and role analysis.

Consider role analysis as an example. According to Crook et al. [9], there exist three types of roles: seniority, functional and contextual. Seniority roles can be derived from actor relationships and organizational hierarchies, whereas functional roles can be derived from job positions. Contextual roles can be derived based on the results of previous task analysis. The contexts of scenarios modeled in task analysis are helpful to derive contextual roles.

The results of the previous three types of analysis produce enough information to specify access control policies.

Additionally, at this level, we can also verify certain security properties, such as least privilege and separation of duties. A formal specification of the access control policies will provide the basis for verifying these security properties.

4.4 Information Flow Analysis

Although information flow analysis does not directly help specify access control policies, it is useful to verify whether the access control policies are effective by tracing where certain sensitive information can flow.

Information flow analysis in access control is different from data flow analysis in RE, in which data flow diagrams are used to describe how data flows through a sequence of processing steps by a system. In access control, when a subject reads information from an object, information flows from the object to the subject. When a subject writes information into an object, information flows from the subject to the object. This analysis is more complex than data flow analysis. We are especially interested in when information flows beyond organizational boundaries, which sometimes suggests security or privacy vulnerabilities.

4.5 Discussion

There are several important considerations that need to be carefully addressed when performing access control analysis.

First, the traceability of an access control policy to the corresponding software requirements is very important.

Regardless of whether access control analysis is conducted in conjunction with requirements analysis or after requirements analysis is completed, we need a two-way relationship between access control policies and requirements. This is because both requirements and access control policies may be changed after the system is implemented. To effectively manage changes, we need to easily determine which access control policies are affected by the changing requirements and make appropriate changes to the corresponding policies, and vice versa.

Second, separation of dynamic aspects from static aspects of access control is very important. The permissions needed to perform a task are relatively stable and will not change very often over the period of system in place. However, what tasks a user may perform changes from time to time. Task assignment and delegation are dynamic aspects of access control. Separation of dynamic aspects from static aspects helps encapsulate static parts in the system, leaving dynamic parts flexible for system administrators to change. In this way, RACAF can provide best support for specifying access control policies.

Third, the compliance of access control policies with high-level security and privacy policies is very important. Antón et al. have examined the alignment of software requirements with security and privacy policies [2]. However, the compliance of access control policies with high-level policies has not been studied.

5. A HEALTHCARE EXAMPLE

This section illustrates the application of RACAF using a healthcare example. There are several reasons why we choose the example from the healthcare domain. First, medical records are considered sensitive and their confidentiality is extremely important in any medical information system. Second, access control analysis in this kind of systems is complex and challenging. In the healthcare industry, various actors from different organizations and departments need to access certain kinds of patient information. Sometimes the request to access a patient record is context-dependent (e.g., the location and the time of the request). Additionally, organizational structures in the healthcare domain are complicated and many roles are involved during the processing of patient data. Sensitive information flow within healthcare institutions is common. All these factors make access control analysis in these kinds of systems a complex and challenging task. Third, legislation (e.g., HIPAA in the U.S.) [20] requires the healthcare industry to protect patient privacy.

We use common healthcare scenarios to illustrate how to perform the three kinds of analysis that are required to specify access control policies in RACAF. The other one, information flow analysis is a challenging task. We have presented a good example in [3], in which sensitive information flows from one organization to another. Here we do not provide concrete example for this analysis. A thorough discussion is outside the scope of this paper, however.

5.1 Data Analysis

Scenario 1: *A patient, Paul, came to the hospital and his doctor David arranged a blood test for him in another department. Before Paul took the test, he was asked by the nurse to fill in a form, which requires him provide his name, address, gender, date of birth, social security number, telephone, and answer some questions about his physical condition and medical history. He was also given a privacy notice that describes how this information will be used. He was requested to sign the form and agree with the privacy notice. This information in the form together with the test results was later entered into the medical information system.*

In this scenario, various types of personal data were collected by the healthcare system. When the patient submitted his personal data via a form, he also submitted his privacy preferences by agreeing with the associated privacy notice. We can describe the relationship using the data model shown in Figure 2, which provides an example set of privacy preferences that are extendable. The user may specify other preferences, such as opt-in/opt-out choices to a particular service. For example, user may select the checkbox "Do not call me or send me marketing information via mail" to opt-out from marketing service of his/her contact information. These preferences can be expressed as context information of data in an access control policy rule.

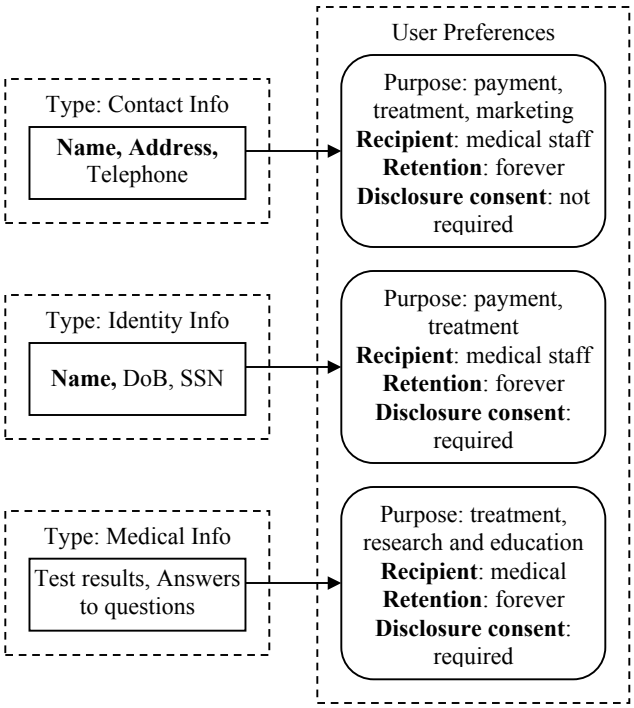

Figure 2. Data model for Scenario 1

5.2 Goal/Scenario-Based Task Analysis

Scenario 2: *Doctor David later retrieve the test results as well as other medical information from the system to make diagnosis*.

There are different ways to model and express a scenario. In the interest of space, we describe this scenario using the diagram shown in Figure 3.

Based on the mapping relationship in Table 2, we can derive access control related information from the scenario analysis as shown in Figure 4. For example, purpose is mapped from the scenario goal. Events are permission candidates. We derive three permissions from six events in this scenario. Constraints are derived from the scenario's preconditions. For example, we specify a constraint "Doctor is the responsible doctor for the

patient" for Permission P3: can access protected health information (PHI) based on the second precondition: Doctor must be the responsible doctor of this patient if he wants to look into the detailed records.

[**Goal**] Retrieve patient medical record to make diagnosis
[**Domain**] Medical diagnosis and treatment
[**Scenario**] A doctor retrieves the medical record of his patient
[**Actors**] Doctor
System
[**Actions**] Actions are listed in the events following actors.
[**Events**] Doctor invokes patient medical records search procedure.
System responds with the search interface.
Doctor enters the name, DoB, SSN or patient medical record number to search the patient.
System responds with the number of records returned
Doctor requests to access detailed information of a record
System responds with detailed information, partial information, or the request is denied
[**Preconditions**]
Doctor must be authenticated before he can invoke patient medical record search procedure.
Doctor must be the responsible doctor of this patient if he wants to look into the detailed records.
The purpose/recipient of data access must be compliant with user's privacy preferences.
[**Postconditions**] Medical records access audit trails are generated.

Figure 3. A scenario of retrieving patient records by his doctor

[**Purpose**] medical diagnosis and treatment
[**Permissions**]
P1: can invoke patient registration procedure
P2: can search patients via name, SSN, etc.
P3: can access protected health information (PHI)
[**Contexts**]
P1, P2, P3: permission domain is medical diagnosis and treatment
[**Constraints**]
P1, P2, P3: Doctor is authenticated.
P3: Doctor is the responsible doctor for the patient.
P3: The purpose for the data access request must be one the purposes associated with the requested data specified in user's privacy preferences
P3: Doctor must be one of the recipients associated with the requested data specified in user's privacy preferences
[**Obligations**]
P3: Medical records access audit trail is generated
[**Role candidates**] R1: the job position of the doctor

Figure 4. Access control information derived from the scenario

5.3 Organizational Structure Analysis

In the healthcare example, we have many players. For example,

- Individual patients
- Healthcare professionals who enter and maintain patient data, such as registrar
- Healthcare professionals who process payments
- Healthcare professionals who carry out diagnosis and treatment, such as physicians and surgeons
- Healthcare professionals who provide nursing services
- Patient family members who provide assistance or may request to access patient information
- Quality assurance professionals who provide initial testing and runtime monitoring of the system

A healthcare system usually has complex organization structures and collaborates with various outside organizations. Some example relationships within/across the healthcare domain are:

- Multiple hospitals in a healthcare system, such as children's hospital, women's hospital, neurology hospital, etc.
- Multiple departments in a hospital, such as nursing center, pharmacy, registration, radiology, etc.
- Regional collaboration with other healthcare systems
- Payment processing with various insurance companies
- Collaboration with college medical schools

By modeling actor relationships and analyzing organizational structures, we are able to assign permissions to abstract concepts, such as roles. For example, at the top of the healthcare hierarchy, we may define a medical staff role, which has the permissions to access information that are only available for staff in the healthcare domain, but not available to others. An example of this mapping is shown in Figure 5.

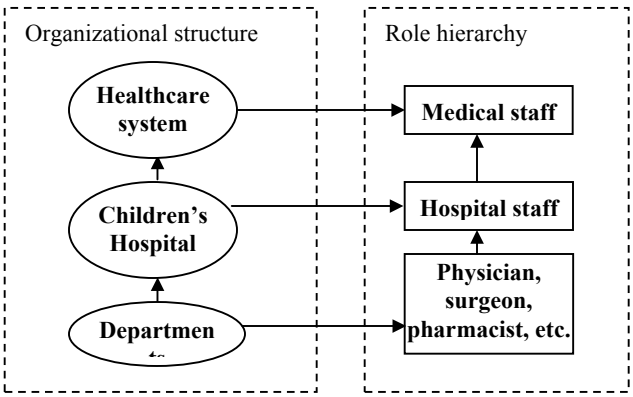

Figure 5. An example of mapping from organizational hierarchy to role hierarchy

6. SUMMARY AND PLANS FOR FUTURE WORK

As an important technique to achieve data security and privacy, access control has not been systematically studied in requirements engineering. Even though certain existing RE methods provide the ability to model one or more aspects of access control, the support is not straightforward and not systematic. One contribution of this paper is a comprehensive set of criteria to

measure whether an approach is sufficient to support access control analysis. These criteria cover various aspects of access control analysis and serve as the desiderata of the proposed methodology.

Based on the pre-defined criteria, this paper proposes a systematic framework RACAF for access control analysis in requirements engineering. To address the various aspects of access control, RACAF provides four types of analysis at different abstraction level: data analysis, task analysis, organizational structure analysis and information flow analysis. RACAF provides systematic support for requirements engineers and security engineers to specify access control policies.

RACAF is a general analytical framework. It is not targeted for specifying a particular type of access control policy or for a particular policy specification language. This is an advantage of RACAF. Additionally, RACAF is designed for analyzing access control in data processing systems. To support access control analysis in security kernels, such as access control in operating systems, we need to further examine the characteristics of access control in these types of systems and propose appropriate methodologies.

Although the ideas presented in this paper are preliminary, RACAF is a promising approach to provide requirements-level support for access control analysis. The framework can be used either during requirements analysis or after requirements specification is complete. We are developing detailed models and heuristics for each type of analysis in RACAF. As mentioned in Section 4, we plan to adopt formal specifications to express analysis results, which will allow RACAF to support formal reasoning. We are also developing tool support for RACAF to facilitate automation of access control specification.

7. ACKNOWLEDGMENTS

The authors wish to thank Dr. Ting Yu and The Privacy Place reading group at NCSU for their helpful comments.

8. REFERENCES

[1] T. Alspaugh, A.I. Antón, T. Barnes and B. Mott. An Integrated Scenario Management Strategy, *Proc. of the 4th IEEE International Symposium on Requirements Engineering (RE'99)*, pp. 142-149, 1999.

[2] A.I. Antón, J.B. Earp and R.A. Carter. Precluding Incongruous Behavior by Aligning Software Requirements with Security and Privacy Policies, Information and Software Technology, Elsevier, 45(14), pp. 967-977, 2003.

[3] A.I. Antón, Q. He, and D. Baumer. The Complexity Underlying JetBlue's Privacy Policy Violations, Accepted, to appear in: IEEE Security & Privacy, 2004 (Also as NCSU CSC Technical Report TR-2003-21).

[4] K. Beznosov. Requirements for Access Control: US Healthcare Domain, Proc. of the 3rd ACM Workshop on Role-Based Access Control, pp. 43, 1998.

[5] C. Bettini, S. Jajodia, S. Wang, D. Wijesekera, Provisions and obligations in policy rule management and security applications, *Proc. of the 28th International Conference on Very Large Data Bases (VLDB'02)*, pp. 502-513, 2002.

[6] G. Booch. *Object-Oriented Analysis and Design with Applications*, 2nd ed., Benjamin/Cummings, Redwood City, CA, 1994.

[7] E. Barka and R. Sandhu. A Framework for Role Based Delegation Model, Proc. of the 23rd National Information Systems Security Conference, pp. 101-114, 2000.

[8] L. Chung. Dealing with Security Requirements During the Development of Information Systems, Proc. of the 5th International Conference on Advanced Information Systems Engineering (CAiSE'93), C. Rolland, F. Bodat, C. Cauvet (editors), LNCS 685, pp. 234-251, 1993.

[9] R. Crook, D. Ince, and B. Nuseibeh. Modelling Access Policies Using Roles in Requirements Engineering, Information and Software Technology, 45 (14), pp. 979-991, Elsevier, 2003.

[10] L. Chung, B.A. Nixon, E. Yu, and J. Mylopoulos. Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, 2000.

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

[12] P. Coad and E. Yourdon, *Object-Oriented Analysis*, 2nd ed., Yourdon Press, Englewood Cliffs, NJ, 1991.

[13] N.C. Damianou. A Policy Framework for Management of Distributed Systems, PhD Thesis, Imperial College, London, 2002.

[14] A. Dardenne and A. van Lamsweerde. Formal Refinement Patterns for Goal-Driven Requirements Elaboration, Proc. of the 4th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (SIGSOFT 1996/ FSE-4), pp. 179-190, 1996.

[15] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-Directed Requirements Acquisition, Science of Computer Programming, 20: 3-50, 1993.

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

[17] P.-J. Fontaine. Goal-Oriented Elaboration of Security Requirements, Project Dissertation, Université Catholique de Louvain, Belgium, 2001.

[18] J. Glasgow, G. Macewen, P. Panangaden. A logic for reasoning about security, *ACM Transactions on Computer Systems*, Vol. 10 (3), pp. 226-264, 1992.

[19] Q. He and A.I. Antón. A Framework for Modeling Privacy Requirements in Role Engineering, Proc. of the 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03), pp. 137-146, 2003.

[20] *The 1996 Health Insurance Portability and Accountability Act (HIPAA)*, http://www.hhs.gov/ocr/hipaa/.

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

[22] G. Karjoth, M. Schunter, and M. Waidner. The Platform For Enterprise Privcy Practices – Privacy-enabled Management

of Customer Data. Proc. of the 2002 Workshop on Privacy Enhancing Technologies, 2002.

[23] A. van Lamsweerde, R. Darimont and E. Letier. Managing Conflicts in Goal-Driven Requirements Engineering, IEEE Transactions on Software Engineering, Vol. 24 (11), pp. 908-925, 1998.

[24] A. van Lamsweerde, R. Darimont and P. Massonet. Goal-directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learnt, Proc. of the 2nd IEEE International Symposium on Requirements Engineering (RE'95), pp. 194-203, 1995.

[25] L. Liu, E. Yu and J. Mylopoulos. Security and Privacy Requirements Analysis within a Social Setting, Proc. of the 11th International Requirements Engineering Conference (RE'03), pp. 151-161, 2003.

[26] J. Mylopoulos, L. Chung, B. Nixon. Representing and Using Nonfunctional Requirements: A Process-Oriented Approach, IEEE Transactions on Software Engineering, Vol. 18 (6), pp. 483-497, 1992.

[27] J.D. Moffett. Control Principal and Role Hierarchies, Proc. of the 3rd ACM Workshop on Role-Based Access Control (RBAC'98), pp. 63-69, 1998.

[28] 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.

[29] The Platform for Privacy Preferences 1.1 (P3P1.1) Specification, The World Wide Web Consortium (W3C), 10 February 2004. http://www.w3.org/TR/2004/WD-P3P11-20040210/

[30] C.S. Powers, P. Ashley, M. Schunter. Privacy Promises, Access Control, and Privacy Management, Proceeding of the 3rd International Symposium on Electronic Commerce, pp. 13-21, 2002.

[31] 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, 1996.

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

[33] A. Schaad, J. Moffett, J. Jacob. The Role-Based Access Control System of a European Bank: A Case Study and Discussion, *Proc. of the 6th ACM Symposium on Access Control Models and Technologies (SACMAT'01)*, pp. 3-9, 2001.

[34] K. Weidenhaupt, K. Pohl, M. Jarke and Peter Haumer. Scenarios in System Development: Current Practice, *IEEE Software*, Vol. 15 (2), pp. 34-45, 1998.

[35] E. Yu and L.M. Cysneiros. Designing for Privacy in a Multi-Agent World, In: Trust, Reputation and Security: Theories and Practice, R. Falcone, S. Barber, L. Korba and M. Singh (editors), LNCS 2631, Springer-Verlag, pp. 209-223, 2003.

[36] E. Yu and L. Liu. Modelling Trust for System Design Using the i* Strategic Actors Framework, In: Trust in Cyber-Societies - Integrating the Human and Artificial Perspectives, R. Falcone, M. Singh, Y.-H. Tan (editors), LNCS 2246, Springer-Verlag, pp. 175-194, 2001.

[37] E. Yu. Modeling Organizations for Information Systems Requirements Engineering, Proc. of the 1st IEEE International Symposium on Requirements Engineering, pp. 34-41, 1993.

[38] E. Yu. Towards Modelling and Reasoning Support for Early-phase Requirements Engineering, Proc. of the 3rd IEEE International Symposium on Requirements Engineering (RE'97), pp. 226-235, 1997.

[39] 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.