Analyzing Regulatory Rules for Privacy and Security Requirements

Travis D. Breaux, *Student Member*, *IEEE*, and Annie I. Antón, *Senior Member*, *IEEE*

Abstract—Information practices that use personal, financial, and health-related information are governed by US laws and regulations to prevent unauthorized use and disclosure. To ensure compliance under the law, the security and privacy requirements of relevant software systems must properly be aligned with these regulations. However, these regulations describe stakeholder rules, called rights and obligations, in complex and sometimes ambiguous legal language. These "rules" are often precursors to software requirements that must undergo considerable refinement and analysis before they become implementable. To support the software engineering effort to derive security requirements from regulations, we present a methodology for directly extracting access rights and obligations from regulation texts. The methodology provides statement-level coverage for an entire regulatory document to consistently identify and infer six types of data access constraints, handle complex cross references, resolve ambiguities, and assign required priorities between access rights and obligations to avoid unlawful information disclosures. We present results from applying this methodology to the entire regulation text of the US Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule.

Index Terms—Data security and privacy, laws and regulations, compliance, accountability, requirements engineering.

1 Introduction

Increasingly, regulations in Canada, Europe, and the US are governing the use and disclosure of information in both the industry and government. This presents different challenges to information systems that support established or emerging business practices. In the US, e.g., federal regulations enacted under the Health Insurance Portability and Accountability Act[1] (HIPAA) require members of the healthcare industry who use electronic information systems to protect the privacy of medical information. Unlike the finance industry, which is known for employing modern security measures, the healthcare industry was largely unprepared. The 10-year cost to comply with HIPAA for the healthcare industry is projected by industry and government stakeholders to be between $12-42 billion [43].

For emerging and evolving businesses, however, existing regulations present a very different challenge. Because regulations are written to address past problems due to market and social changes, new information-driven business models may not be adequately vetted before they are put into practice. For example, the US Fair Credit Reporting Act[2] (FCRA) was enacted to ensure accuracy in the maintenance and reporting of personal information by credit bureaus. Recently, an "information broker" called ChoicePoint has acknowledged that records on more than 163,000 consumers were acquired by identity thieves [18]. A prior review of ChoicePoint's business products suggests that the company has, with or without intent, developed these products without proper controls mandated by the FCRA [17]. The Federal Trade Commission confirmed this suspicion in 2006: under the FCRA, ChoicePoint was fined $15 million in civil penalties and consumer redress and will undergo biennial security audits for the next 20 years [18], [19]. Violations similar to the one by ChoicePoint are believed to be due to how regulations are interpreted by companies in the context of their information system designs [17].

To support software and requirements engineers, system administrators, and policy makers, we developed a methodology for extracting formal descriptions of rules that govern stakeholder actions from policies and regulations [13]. Actions that are permitted by regulations are called *rights*, whereas actions that are required are called *obligations*. From stakeholder rights and obligations, we can infer system requirements that implement these rules to comply with regulations. In this paper, we build upon our prior work by presenting two extensions to this methodology using a tabular format that includes 1) a method for acquiring and presenting data access requirements and 2) a method for acquiring and managing priorities between data access requirements. We validated these extensions in a case study by using the HIPAA Privacy Rule [30] to yield 300 rules that govern stakeholder access to medical information. The HIPAA Privacy Rule affects some 545,000 different establishments in the US who employ over 13.5 million people [16]. The contributions presented in this paper are the extended methodology and a catalog of constraint types that resulted from validating the methodology in the HIPAA case study.

The remainder of this paper is organized as follows: In Section 2, we review related work and discuss the rule-making process that yielded the HIPAA Privacy Rule. We

[1] US Pub. Law 104-191, est. 1996.
[2] US Pub. Law 91-508, est. 1970.

• *The authors are with the Computer Science Department, North Carolina State University, Engineering Building II, Suite 3231, Raleigh, NC 27606. E-mail: {tdbreaux, aianton}@ncsu.edu.*

Manuscript received 24 Feb. 2007; revised 11 July 2007; accepted 29 Aug. 2007; published online 17 Sept. 2007.
Recommended for acceptance by P. McDaniel and B. Nuseibeh.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number TSESI-0082-0207.
Digital Object Identifier no. 10.1109/TSE.2007.70746.

Published by the IEEE Computer Society

follow with important terms and definitions in Section 3. In Section 4, we present the basic methodology for extracting access rights, obligations, and constraints with an example from the HIPAA Privacy Rule. In Section 5, we present the results of a case study by using the Privacy Rule, including a catalog of access constraints and a review of exceptions. In Section 6, we conclude with our discussion and summary.

2 Background and Related Work

In software engineering, Zave and Jackson define software requirements to be desirable environmental phenomena and they distinguish systems by their ability to exert control over the environment [42], [23]. Moreover, they identify the challenge that engineers face in distinguishing between descriptions of the domain and descriptions of systems [24]: the latter include system requirements and specifications. Because regulations include both statements about systems and, more often, statements about stakeholder behavior, this distinction is especially important for software engineers who work with legal requirements. The terms *due diligence*, *due care*, and *standard of care* refer to reasonable efforts that people make to satisfy legal requirements or discharge their legal obligations [20]. In the US, *standard of care* means "under the law of negligence or of obligations, the conduct demanded of a person in a situation, and typically, this involves a person who gives attention both to possible dangers, mistakes, and pitfalls and to ways of minimizing those risks" [20]. To make claims that software complies with a regulation, engineers must employ traceability from regulatory descriptions about the world to system requirements and specifications. Engineers must further justify that their interpretations of regulations are valid and consistent with their specifications and that system behaviors do not contradict those interpretations. The rigorous methodology that we propose in this paper will provide part of this important justification, which is necessary for establishing due diligence and a reasonable standard of care in software engineering.

In requirements and software engineering, researchers have investigated methods for analyzing security requirements by using aspects [41], goals [22], [32], problem frames [34], [27], trust assumptions [26], and structured argumentation [28]. More recent work focuses on the rigorous extraction of requirements from security-related policies and regulations [35], [13], [33]. In our earlier work, we presented a methodology for extracting stakeholder rights and obligations from regulations [9], [13]. Rights and obligations are similar to the notions of "what is permissible" and "what ought to be," as modeled by Deontic Logic [29]. The methodology combines the Goal-based Requirements Analysis Method (GBRAM) [1] and a process called Semantic Parameterization for acquiring formal models from natural language statements [11]. The methodology was validated in a pilot study that uses a patient fact sheet which summarizes the HIPAA Privacy Rule [9] and in a larger case study that uses four sections of the Privacy Rule which are concerned with privacy notices, requests for access restrictions, and patient access to review and amend their medical information [13].

In this paper, we extend this methodology to address issues that are specific to deriving data access requirements. The first extension includes applying four natural language patterns from Semantic Parameterization to extract formal models from policies [7], [8]. The products of this extension are access control elements that include data subjects, objects, and purposes, the relevant principals (e.g., authorized actors), and preconditions in data access [38]. The second extension includes methods for identifying, managing, and prioritizing important exceptions that must be respected to avoid illegal and unauthorized information use and disclosures.

May et al. describe a methodology for extracting formal models from regulations that they applied to one section of the HIPAA Privacy Rule [35]. Our work for extracting formal models from four sections of the Privacy Rule [13] presents contradictory insight into several of their basic assumptions, including 1) each paragraph has exactly one rule and 2) external and ambiguous references are satisfiable by the environment [35]. Although their models can be shown to be logically consistent, their methodology lacks explicit techniques to handle ambiguities and constraints acquired from cross references; thus, their models are prone to being inconsistent with the HIPAA Privacy Rule. Lee et al. employ an ontological approach to extract requirements from regulations [33]. Because their approach categorizes requirements by using an ontological model, the approach helps engineers in rigorously identifying inconsistencies between the model and the regulations. This is an improvement over May et al. [35]. To varying degrees, our methodology [13], [11] solves many of the problems that Lee et al. identify and that May et al. do not address, including issues of verbosity, ambiguity, polysemy, redundancy [33], and cross references [35].

It is important to distinguish access control rules (ACRs) from stakeholder rights and obligations that pertain to access. ACRs are triples that consist of a principal, an action, and an object in which the principal is or is not permitted to perform the action on the object [38]. The principal may represent a user or software process, the actions include read, write, and execute, and the object may be data or a function in software. In stakeholder rights and obligations, the object may be an abstract collection of data such as "protected health information" (PHI) or a specific data element such as "an individual's name." What constitutes these objects in software is a nontrivial matter of design. Moreover, the constraints on stakeholder rules often describe environmental circumstances that require considerable refinement, design, and engineering before they are realized within software systems, as shown in Section 5.1. Although we could express stakeholder rights and obligations by using an access control language such as the Extensible Access Control Markup Language (XACML) [44], the resulting expressions will only trivialize the exceptional software engineering effort that remains to ensure that systems comply with the law. Despite this important distinction, several researchers have proposed directly mapping natural language policies that describe stakeholder rights and obligations into ACRs [15], [37], [40]. Although these efforts yield formal mappings to natural language policies, they stop short of demonstrating how systems will interpret and comply with the intent of these policies. Nevertheless, we believe that ACRs may be inferred from stakeholder rules with proper analysis and requirements engineering, based, in part, on the results that our methodology provides.

2.1 Evolution of the United States Health Insurance Portability and Accountability Act

The US legislation HIPAA was passed in August 1996 for numerous reasons, including the need for increased protection of patient medical records against unauthorized use and disclosure. HIPAA requires the US Department of Health and Human Services (HHS) to develop, enact, and enforce regulations that govern electronically managed patient information in the healthcare industry. Consequently, from 1998 to 2006, a special committee of the HHS prepared several recommendations based on extensive expert witness testimony from academe, industry, and government that concluded in three regulations:

The *Security Rule* requires implementing a security program that includes physical and electronic safeguards, policies for authorizing and terminating access to electronic information, and technical security controls for logging access, password management, and encrypted transactions [31].

The *Privacy Rule* requires implementing policies and procedures to restrict access to patient information for specific purposes such as to provide emergency treatment, inform law enforcement of a crime, or conduct workplace medical surveillance [30].

The *Enforcement Rule* states the actions that must be taken by the HHS to ensure compliance and accountability under the HIPAA, including the process for reviewing complaints and assessing fines [25].

The infrastructure requirements in the Security Rule are not revolutionary and will likely raise security standards in the healthcare industry closer to the standards that have existed in finance for decades. On the other hand, organizations must currently interpret the Privacy Rule to individually align each regulation with relevant business processes and transactions in their organization. This degree of coordination requires not only understanding the rule of law (the domain of lawyers) but also understanding the technical capabilities of the software systems responsible for managing these transactions (the domain of software engineers and system administrators). Furthermore, due to heterogeneity in business practices and software systems, there will never be one road to HIPAA compliance.

To facilitate compliance under regulations such as HIPAA, we developed a requirements engineering methodology for extracting stakeholder rights and obligations from regulations [13]. *Rights* describe what actions stakeholders are permitted to perform, while *obligations* describe what action stakeholders are required to perform. From stakeholder rights and obligations, engineers can reason about which requirements are necessary to comply with the law. The methodology provides statement-level coverage to improve compliance by ensuring that each regulation either aligns with one or more software requirements or has been deemed irrelevant to current business practices. In addition, the methodology provides constraint-level traceability across statements and cross references. This degree of traceability improves accountability by aligning software artifacts derived from rights and obligations with specific paragraphs in the regulation text [12].

3 Terminology and Definitions

The methodology uses the following terms:

- A *definition* is a statement that restricts the meaning of a term by using one or more constraints. For example, the statement "a healthcare provider is an entity who provides health services" defines the term "healthcare provider" (a concept) by their role as a provider of health services, in which the role is a constraint on the concept. We discuss definitions in Section 4.1.
- A *property* is an attribute or characteristic associated with a formal representation of a concept. For example, a person's name is a property of a person or an action is a property of an activity. We discuss properties in Section 4.2.
- A *constraint* is a statement that restricts or limits the possible interpretations for a concept via one of its properties or a relationship to another concept such as a role in an activity. For example, the phrase "a patient who receives healthcare services" constrains the set of all possible patients to the possibly smaller set of only those patients who are also recipients of healthcare services. We present a catalog of constraints in Section 5.1.
- A *right* is a statement about one or more actions that a stakeholder is permitted to perform. If a stakeholder is expressly not obliged to perform an action, called an *antiobligation*, then this statement also describes a right.
- An *obligation* is a statement about one or more actions that a stakeholder is required to perform. If a stakeholder is expressly not permitted to perform an action, called a *refrainment*, then this statement also describes an obligation.
- A *rule* can be a right, obligation, or refrainment as per our definitions above. Rules are often restricted in some ways by constraints.
- An *exception* denotes a relationship between two properties or rules in which all of the possible interpretations of the one thing (e.g., a property or rule) exclude the possible interpretations of the other. For example, the exception "health information except for psychotherapy notes" refers to the set of all possible interpretations for "health information," excluding the set of all things that comprise "psychotherapy notes." An exception between two rules establishes a *priority*, in which case if the higher priority rule applies to a specific situation, the other, that is, the lower, priority rule would not apply. We discuss exceptions in Section 5.2.

4 Encoding Rules from Regulations

The requirements engineering methodology for encoding rules from regulations discussed in this paper was developed using Grounded Theory [21], in which observations from a data set are relevant to that data set. Although the methodology has only been validated using HIPAA-related documents, based on our experience in developing similar goal-based methodologies in other domains, we believe that this methodology is generalizable beyond HIPAA [1], [2], [3]. We review this methodology in Sections 4.1 and 4.2 to provide important background and as a foundation to our new extensions for modeling data access requirements presented in Section 4.3.

The methodology requires the requirements engineer to analyze each statement in a regulation text and identify the statement as a definition, right, obligation, or constraint [13]. As previously mentioned, right and obligation statements may contain constraints on various properties, such as the subject or recipient in an information disclosure. In Section 4.1, we illustrate the role of definitions in establishing a classification hierarchy of stakeholders, which helps engineers identify which rules apply to their organization and to disambiguate them. In Section 4.2, we describe the process for extracting rights, obligations, and constraints from regulatory texts by using extensively validated natural language patterns [7], [8], [9], [13]. In Section 4.3, we show how we can map these rule elements to parameterized rules by using six properties and demonstrate how we can derive priorities between these rules from exceptions.

4.1 Definitions and Stakeholder Hierarchies

Definitions in the US federal and state regulations define terms by enumerating their specializations, which are additional terms that correctly exemplify a concept, or by elaborating the concept's role in relevant activities. Legal professionals refer to these concept terms as a *term of art*, defined as "a word or phrase that has a specific precise meaning in a given specialty, apart from its general meaning in ordinary contexts" [20]. In these regulations, the concept term may be substituted with one of the specializations or elaborations without yielding an incorrect interpretation of the affected regulatory rules. Consider the following definition for the term "covered entity" (CE) from §160.103 of the HIPAA Privacy Rule:

> *Covered entity* means 1) a health plan, 2) a healthcare clearinghouse, or 3) an health care provider who transmits any health information in electronic form in connection with a transaction covered by this subchapter.

This definition includes three additional terms (a health plan, a healthcare clearinghouse, and a health care provider (HCP)) that, in conjunction with their roles in the act of electronically transmitting health information, are specializations of the term *CE*. These three terms are themselves defined using other more specialized concepts, as illustrated by the following partial definition for a health plan, also from §160.103:

> *Health plan* includes the following, singly or in combination: a group health plan, a health insurance issuer, a health management organization . . .

Due to the specialization relationships between these concepts, we can derive a corresponding stakeholder hierarchy (see Fig. 1). The shaded boxes in this hierarchy indicate stakeholders who were identified during the case study described in Section 5 but who do not have separate definitions in either definition section of the HIPAA Privacy Rule, that is, §160.103 and §164.503.

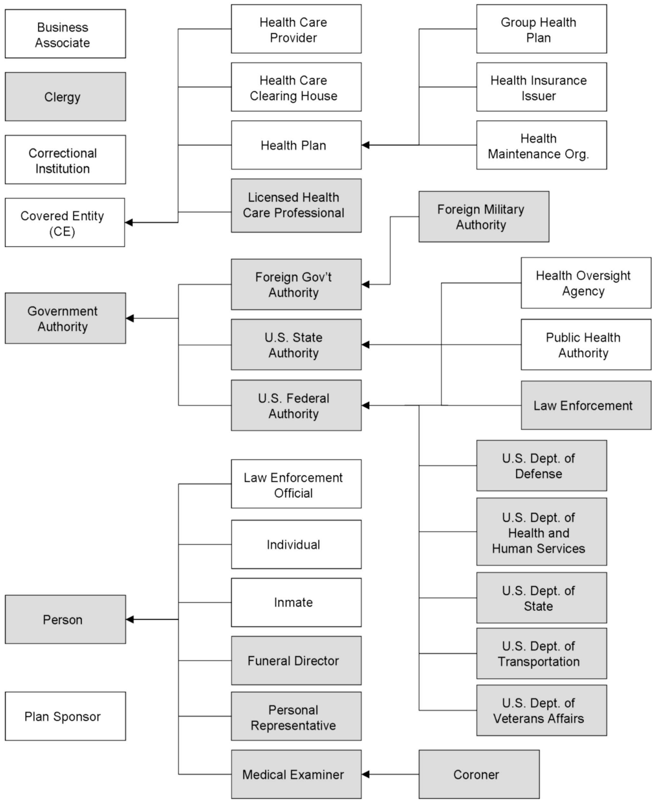


Fig. 1. Stakeholder hierarchy for the HIPAA Privacy Rule.

For a particular stakeholder, identifying which rules apply in a given situation includes evaluating rules that apply to general classifications of that stakeholder (e.g., via the transitive closure). For example, group health plans must consider rules that directly apply to them and rules that apply to their more general classifications, including health plans and covered entities. In addition to the classification hierarchy, software engineers must also consider stakeholder membership in an organization. For example, the actions of a person who is a law enforcement official are subject to rules that govern that classification and to rules that govern law enforcement (the agency) in general. Not all memberships are transitive, however. Rules that apply to correctional institutions do not apply to inmates, or vice versa, despite the fact that inmates have membership in a correctional institution. In summary, the stakeholder hierarchy defines a rule's scope of impact, thereby helping engineers visually understand which classes of stakeholders are affected by a rule and by supporting formal reasoning about similarities and conflicts between rules [13].

4.2 Rights, Obligations, and Constraints

Rights and obligations describe actions that stakeholders are permitted and required to perform, respectively. Consequently, in developing systems to support these actions, developers must ensure that these systems also satisfy any constraints on those actions. Constraints restrict the scope of possible interpretations of requirements to relevant systems, environmental circumstances, and stakeholders who satisfy these constraints. In privacy regulations, constraints are expressed by lawyers who have given careful consideration to the intent of the law. In this context, removing or overlooking constraints can permit unintended uses or disclosures of confidential information. As natural language constraints (as opposed to constraints formally expressed) can be subtle and easily overlooked, we developed several patterns to consistently extract natural language constraints into formal predicates in first-order logic [7], [8], [9], [11]. The patterns that we use here include the *basic activity pattern with modality* [7], [8], *purposes* [8], *nouns distinguished by verb phrases* [8], and *rules or conditions* [9]. We first present the results of applying these patterns before discussing each pattern in detail. The results are derived from the two excerpts of §164.510 and §164.522 of the HIPAA Privacy Rule. These excerpts describe the covered entity (CE), health care provider (HCP), and protected health information (PHI). In these excerpts, the rule statement is *italicized* with the inline modal phrases (must, may, etc.) in ***bold***, the constraints are underlined, and the condition keywords (except, if, and, and or) are in **bold**.

Excerpt from the Privacy Rule §164.510(b)(1)(i):

(b)(1)(i) *A CE **may***, in accordance with paragraphs (b)(2) **or** (3) of this section, *disclose to a family member, other relative,* **or** *a close personal friend of the individual* **or** *any other person* identified by the individual *the PHI* directly relevant to such person's involvement with the individual's care **or** payment related to the individual's healthcare.

Excerpt from the Privacy Rule §164.522(a)(1)(i)-(iii)

(a)(1)(i) *A CE **must** permit an individual to request that the CE restrict*

(A) *Uses or disclosures of PHI about the individual* to carry out treatment, payment or healthcare operations; **and**

(B) *Disclosures* permitted under §164.510(b).

(ii) *A CE **is not required** to agree to a restriction.*

(iii) *A CE* that agrees to a restriction under paragraph (a)(1)(i) of this section ***may not*** *use* **or** *disclose PHI in violation of such restriction*, **except** that **if** the individual who requested the restriction is in need of emergency treatment **and** the restricted PHI is needed to provide emergency treatment, *the CE **may** use the restricted PHI* **or** *may disclose such information to a HCP to provide such treatment to the individual.*

Applying our patterns to the excerpts above yields the constraints listed in the following, which include C_1-C_{11}, in addition to rules that include obligation O_1, refrainments O_2, O_3, the antiobligation R_2, and the three additional rights R_1, R_3, and R_4. For traceability, each constraint and rule statement is followed by a reference to the paragraph from which it was extracted and, for rules, a first-order propositional logic expression over constraints, including preconditions and postconditions to the rule (in square brackets).

Constraints on rules (listed in order of extraction)

C_1. The individual identified the person (§164.510(b)(1)(i)).

C_2. The PHI is directly relevant to the person's involvement in the individual's care (§164.510(b)(1)(i)).

C_3. The PHI is directly relevant to the person's involvement in payment related to the individual's healthcare (§164.510(b)(1)(i)).

C_4. The use is to carry out treatment, payment or healthcare operations (§164.522(a)(1)(i)(A)).

C_5. The disclosure is for carrying out treatment, payment, or healthcare operations (§164.522(a)(1)(i)(A)).

C_6. The CE agrees to a restriction under paragraph (a)(1)(i) (§164.522(a)(1)(iii)).

C_7. The individual requested the restriction (§164.522(a)(1)(iii)).

C_8. The individual is in need of emergency treatment (§164.522(a)(1)(iii)).

C_9. The PHI is needed to provide emergency treatment (§164.522(a)(1)(iii)).

C_{10}. The use is for providing emergency treatment to the individual (§164.522(a)(1)(iii)).

C_{11}. The disclosure is for providing emergency treatment to the individual (§164.522(a)(1)(iii)).

Stakeholder rules (listed in order of extraction)

R_1. A CE may disclose PHI to a person (§164.510(b)(1)(i); $C_1 \wedge (C_2 \vee C_3) \wedge (\ldots)$).

O_1. A CE must permit an individual to request a restriction (§164.522(a)(1)(i)).

R_2. A CE is not required to agree to a restriction (§164.522(a)(1)(ii)).

O_2. A CE may not use PHI (§164.522(a)(1)(iii); $C_6 \wedge C_4$).

O_3. A CE may not disclose PHI (§164.522(a)(1)(iii); $C_6 \wedge (C_5 \vee (\ldots))$).

R_3. A CE may use PHI (§164.522(a)(1)(iii); $C_7 \wedge C_8 \wedge C_9 \wedge C_{10}$).

R_4. A CE may disclose PHI to an HCP (§164.522(a)(1)(iii); $C_7 \wedge C_8 \wedge C_9 \wedge C_{11}$).

We now discuss the four patterns that a software engineer applies to identify rights, obligations, and constraints.

The *basic activity pattern* describes a subject who performs an action on an object and *modality* distinguishes the activity as a right, obligation, or refrainment [7], [8], [13]. Each rule uses these two patterns to ensure that the statement has precisely one subject, action, object, and modality. For example, obligation O_1 uses the modal phrase "must" to denote the CE (subject) that permits (action) the individual to request a restriction (an act that is the object of the action). Constraint statements also satisfy the basic activity pattern but rarely contain modalities like rights or obligations.

The *purpose pattern* describes the high-level goal or reason for performing an action [8]. Consequently, a purpose is a constraint on the act and not a constraint on the actor who performs the action or on the object upon which the action is performed. In paragraph (a)(1)(i)(A), the phrase "to carry out treatment, payment, or healthcare operations" indicates the purpose of the use or disclosure of

PHI. This purpose is separately described in constraints C_4 and C_5 for the acts of "use" and "disclosure," respectively.

The *pattern to distinguish nouns by verb phrases* is often indicated by words that include "who," "that," and "which" followed by verb phrases [8]. In paragraph (a)(1)(iii), we apply this pattern to the underlined phrase "that agrees to a restriction ..." to yield the constraint C_6 on the CE. This constraint appears in the propositional formula for the two refrainments O_2 and O_3. The usual words "who," "that," and "which" are not always present, however. In paragraph (a)(1)(i)(B), the noun "disclosures" is followed by the verb phrase "permitted under ...," which omits the indicative words "that" or "which."

The *rule pattern* describes preconditions and postconditions (constraints) using condition keywords (e.g., if, except, unless, upon, and when) [9]. In paragraph (a)(1)(iii), the keyword "if" is followed by two underlined phrases that precondition the rights to use or disclose PHI. The underlined phrases are separated into the three constraints, C_7, C_8, and C_9, and appear in the propositional formula for the corresponding rights R_3 and R_4.

The English conjunctions (and, or) are often ambiguous and must be assigned strict logical interpretations. For example, paragraph (a)(1)(i) describes two obligations of a CE to "permit an individual to request that the CE restrict A) uses or disclosures of PHI about the individual to carry out treatment, payment or healthcare operations; **and** B) disclosures permitted under Section 164.510(b)." We interpret the English conjunction "and" in this statement as a logical OR because the individual may request any of these restrictions, independent of the other. Policy makers and software engineers may have differing views on these conjunctions. For example, using the English conjunction "and," §164.512(f)(2)(i) paragraphs (A)-(H) of the Privacy Rule lists eight specific types of information (e.g., name, date of birth, and social security number) that may be disclosed. In practice, this disclosure may be governed by a policy that requires only disclosing the minimum necessary information to complete a transaction (see Minimum Necessary in §164.502(b)). Interpreting this conjunction as a logical AND simplifies software designs because the software engineer only needs to consider a single case in which all of the information is disclosed during each transaction. However, if the conjunction is interpreted as a logical OR, there are 255 subsets of the eight information types that can be disclosed, depending on what information is minimally required. This latter interpretation requires more effort on the part of software engineers to implement a system that allows users to select which information subset to disclose during each transaction. Therefore, engineers should consider the legal and the technical ramifications of choosing a logical interpretation for English conjunctions.

Cross references require engineers to systematically copy constraints that are acquired from other sections of the regulation into the propositional formula for a rule. In §164.510(b)(1)(i), e.g., the phrase "in accordance with paragraphs (b)(2) or (b)(3)" indicates that these paragraphs may contain additional constraints on the right, R_1, to which this phrase applies. To complete right R_1, the engineer must identify these constraints and incorporate them into the space "(...)" that appears in the propositional formula for right R_1. Constraints are often copied across multiple cross references. For example, in §164.522(a)(1)(i)(B), the phrase "disclosures permitted under §164.510(b)" exclusively refers to rights that were extracted from §164.510(b), such as right R_1. This cross reference refers to rights that contain the action *disclose*. The object *PHI* is inferred from the sentence that contains this cross reference. To complete refrainment O_3, which is extracted from this same sentence, the engineer must identify the rights extracted from §164.510(b) and incorporate the constraints from the propositional formula of those rights such as $[C_1 \wedge (C_2 \vee C_3) \wedge (\ldots)]$ in R_1 into the space "(...)" in the propositional formula of O_3.

Cross references are challenging to software engineers, because the constraints that should be incorporated from other sections may not yet have been extracted from those sections by the engineer (as in the case with refrainment O_3 and right R_1 above). This can cause engineers to skip around the regulation text, which may lead to inconsistencies in applying the methodology. Moreover, the regulatory statements are often written to be intentionally ambiguous to support broad legal interpretations. For example, refrainment O_3 does not state "to whom" these disclosures are made. Rather, the recipient of these disclosures depends on the interpretations of rights expressed in §164.510(b) and includes family members, close personal friends of the individual, etc. The approach that we recommend is to extract all of the rules and constraints from each paragraph in the order in which they are identified but to postpone traversing all cross references until a complete pass through the entire regulatory text is completed. The engineer will then conduct a second pass, only traversing cross references, in which he/she will then copy the previously extracted constraints between corresponding rules. The extensions that we now discuss in Section 4.3 will help engineers more quickly isolate only the relevant constraints from cross references, thus simplifying cross-reference analysis and management during the first and second passes.

4.3 Extensions for Data Access Rules

The methodology has been applied to extract rights and obligations that govern a variety of practices supportable by software systems, including notice of privacy practices and rights to amend and restrict access to PHI [13]. In this section, we extend the methodology with two new methods to 1) identify allow or deny rules relevant to information access and parameterize these rules to separately denote principals and data subjects, objects, and purposes, if any, and 2) identify exceptions to rules that prioritize access rights, obligations, and refrainments. These procedures yield two separate tables, called the *rule table* and the *priority table*, respectively, as discussed in the following. These two extensions expose important details that will help software engineers design access control systems. In Section 5.1, we discuss how engineers can infer critical requirements for these systems from many of the constraints in this extended format.

In our prior work, we described a process called Semantic Parameterization that is used to map words that describe concepts from simple sentences into first-order

Record Number: 270			
Row	**Paragraph**	**Property**	**Value**
1	164.522(a)(1)(iii)	Subject	CE
2	164.522(a)(1)(iii), 164.522(a)(1)(i)(B)	Action	Disclose
3	164.522(a)(1)(iii)	Modality	Refrainment
4	164.522(a)(1)(iii)	Object	PHI
5	164.510(b)(1)(i)	Target	Person

Fig. 2. Initial record for refrainment O_3.

predicate logic expressions [7], [8], [11]. These predicates distinguish important properties such as the subject, action, and object of an activity. For the purpose of constructing the rule table, we consider six properties (*italicized*) in information access-related activities:

1. The *subject* is the actor who performs an action on an object in the activity.
2. The *action* is a verb that affects information such as access, use, disclose, etc.
3. The *modality* modifies the action by denoting the action as a right, obligation, or refrainment.
4. The *object* is limited to information, including the name or date of birth of a patient or an accounting of disclosures.
5. The *target* is the recipient in a transaction such as the recipient of a disclosure.
6. The *purpose* is the goal of an activity. For example, patient information may be used for billing or treatment purposes.

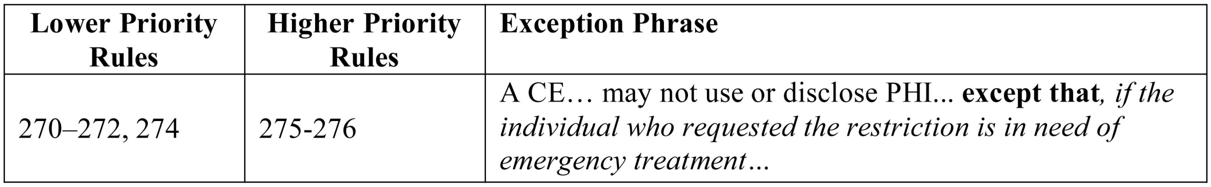
The *rule table* contains records, each of which corresponds to a rule that is a right or an obligation to access, use, or disclose information. Each row in the record is a constraint on the rule that includes the regulation paragraph number from which the constraint was acquired, one of the six property names that the constraint affects, and the constraint value. *Parameterized constraints* are those constraints whose value is a word or noun phrase from the rule statement that corresponds to one of the six properties. For example, refrainment O_3 from Section 4.2 has been parameterized and appears in Fig. 2. The value "disclose" of the action property is stated in both paragraphs §164.522(a)(1)(i)(B) and (a)(1)(iii); thus, the constraint is indexed by both to maintain traceability across the cross reference. The target property "person" is not stated in refrainment O_3: This is an ambiguity in the regulatory text. Instead, this constraint is acquired by following the cross reference to §164.510(b)(1)(i) to identify the records that were previously extracted from this paragraph and incorporating the relevant constraint rows from those records. This approach to addressing cross references resolves this type of ambiguity quite well and maintains traceability across multiple paragraphs. Because the constraints are recorded using the rule table, it is relatively easy to identify constraints that are derived from a specific paragraph number or that correspond to specific actions (e.g., disclosures or uses) or modalities (e.g., rights or refrainments).

In addition to the parameterized constraints, we add rows to the record for constraints C_1, C_2, and C_6 that appear in the propositional formula for refrainment O_3. These constraints are called *nonparameterized constraints* because the constraint statements were not parameterized like the right and obligations statements. Fig. 3 shows the nonparameterized constraints derived from C_1, C_2, and C_6, respectively. Each nonparameterized constraint value is derived from a constraint statement by replacing the subject with an anonymized word "who," "where," or "which," depending on whether the subject is a person, a place, or a thing, respectively. Constraint statements may refer to multiple entities. For example, the statement C_1, that is, "the individual identified the person," describes two entities: the individual and the person. If the nonparameterized constraint value of the same property name describes an entity other than the subject of the constraint statement, then the statement must be rephrased so that the subject is the correct entity and this process is called *retopicalization*. Therefore, we retopicalize the statement C_1 to yield the nonparameterized target constraint value "Who is identified by the individual" on row 7 in Fig. 3 that corresponds to the constraint value "Person" for the target property on row 5 in Fig. 2.

The *priority table* contains records that establish priorities between rules in the rule table. Priorities must be documented to resolve exceptions to rules and, later, to prioritize derived software requirements. As shown in Fig. 4, each record in a priority table contains an exception phrase that illustrates the context of the exception and two lists of rule numbers: the list of *higher priority* rule numbers that are exceptions to the list of *lower priority* rule numbers. The italicized portion in the exception phrase highlights the text from which the higher priority rules were extracted, the words that indicate the exception are in bold, and the nonitalicized portion highlights the text from which the

Record Number: 270			
Row	**Paragraph**	**Property**	**Value**
6	164.522(a)(1)(iii)	Subject	Who has an agreement with an individual to restrict disclosures of PHI. (C_6)
7	164.510(b)(1)(i)	Target	Who is identified by the individual. (C_1)
8	164.510(b)(1)(i)	Object	Which is directly relevant to the person's involvement with the individual's healthcare. (C_2)

Fig. 3. Extended record for refrainment O_3.



Lower Priority Rules	**Higher Priority Rules**	**Exception Phrase**
270–272, 274	275-276	A CE… may not use or disclose PHI... **except that**, *if the individual who requested the restriction is in need of emergency treatment…*

Fig. 4. Record for priority between rules 270-272 and 274-276.

lower priority rules were extracted. Fig. 4 illustrates an example record from the priority table. The example exception appears in §164.522(a)(1)(iii) in the excerpt in Section 4.2 in which the refrainment O_3 to "not disclose PHI" is followed by the right R_4 (an exception to O_3). The refrainment O_3 corresponds to a total of four nondisclosure rules that were extracted in our case study: rules 270-272 which include a unique constraint from the cross reference to §164.510(b) and rule 274 which includes constraint C_5 extracted from paragraph (a)(1)(i)(A) in §164.522. Rules 275 and 276 refer to the rights R_3 and R_4, respectively, which are the exceptions to use and disclose PHI in emergency situations.

To accurately identify the rules affected by an exception, the engineer must first isolate the constraints stated in the exception and then perform a two-factor comparison by 1) looking up rules that match the cross-referenced section or paragraph number and 2) matching the constraints from the exception with the constraints in those rules. Because a single constraint statement can be distributed across multiple rules in a section or paragraph, a single exception can affect priorities between multiple rules. For example, the exception phrase in Fig. 4 actually prescribes four different priorities between five different rules. In Section 5, we present several exception patterns that we used to standardize the identification and interpretation of priorities between rules.

5 Case Study in Information Privacy

The extended methodology in Section 4 was applied to the HIPAA Privacy Rule, including §160.310 and §164.502-§164.532, to yield 300 stakeholder access rules. The analysis encompassed four passes through all 55 pages of the Rule [30], with two people working in tandem. The rules were first extracted over two passes that required close to 26 hours. The priorities were then extracted over two more passes. These two passes required close to 29 hours. Subsequent passes led to insights that evolved and refined the methodology to the form presented herein. These insights occurred during 18 hours of analysis that overlapped with the time to extract both rules and priorities. Of the total stakeholder access rules and exceptions identified in this study, the initial passes discovered 90.3 percent of the total rules and 89.6 percent of the total exceptions. During the initial passes, we identified new heuristics (e.g., new action verbs or priority patterns) that yielded the remaining 9.7 percent of the rules and 10.4 percent of the exceptions. It is reasonable to expect that future studies would take less time because the refined methodology presented herein provides previously unavailable guidance to the engineer for identifying and extracting important elements, including rights, obligations, constraints, and priorities from regulatory texts. Although new phrases will inevitably be encountered, our experience with regulatory texts in other domains [14] shows that these phrases are often variations on the same elements that we report in this paper, suggesting that the methodology is generalizable to domains beyond healthcare.

The 300 extracted rules are expressed in the rule record format from Section 4.3 and are comprised of 1,894 constraints. Several of these constraints contain disjunctions over related concepts. Performing case splitting on these disjunctions, as explained in [8], would increase the number of extracted rules. Although only 50 rules were refrainments (deny access), the priorities between rules have a significant impact on shaping the access space when a refrainment overrules a right of access and vice versa. Among the 58 extracted exceptions, there are more than 12,205 priorities between different rules.

5.1 Catalog of Constraint Types

Constraints in rights and obligations restrict the set of situations in which regulatory rules are applicable. For software engineers, the accurate design of systems governed by these regulations depends upon the satisfiability of these constraints using available technology. A constraint is *satisfiable* if a hardware or software process will terminate and report true if and only if the constraint has been satisfied by the system. Because regulatory constraints usually describe stakeholder actions performed in the system environment, engineers must reason about the steps to implement these constraints to address their satisfiability concerns in terms of the environment. Mylopolous et al. have termed this procedure *satisficing* in the context of high-level goals [36].

In addition to satisfiability, software engineers must distinguish between compliance and accountability under regulations. A software system is noncompliant under a regulation if that system exhibits behavior that is not permissible under that regulation; otherwise, the system is deemed compliant. Separately, a software system is accountable under a regulation if, for every permissible and nonpermissible behavior, there is a clear line of traceability from the exhibited behavior to the software artifacts that contribute to this behavior and the regulations that govern this behavior. Consider information access for example. A compliant system ensures that only those stakeholders who are permitted access to information will receive access. An accountable system, on the other hand, can demonstrate which regulatory rules apply to every transaction and produce a corresponding audit trail [10], [12].

TABLE 1
Legal Determinations

Paragraph	Property	Value
164.504(e)(4)(i)	Target	Who needs the PHI to comply with its obligations under 29 CFR parts 1904 through 1928, 30 CFR parts 50 through 90, or under state law having a similar purpose.
164.510(b)(2)(i)	Target	Who has lawful custody of an inmate or individual.
164.512(b)(1)(ii)	Target	Who is authorized by law to receive reports of child abuse or neglect.
164.512(i)(1)(iii)(B)	Subject	Who treats the individual as required by law.
164.512(i)(1)(ii)(C)	Target	Who are authorized by 18 U.S.C. 3056.
164.514(g)	Subject	Who is authorized by law to notify persons to conduct public health interventions.

Improving accountability will demonstrate due diligence and improve compliance, whereas a compliant system may not be accountable at all. As we will illustrate in Section 5.2, a stakeholder can have access to information for multiple reasons. Having the ability to precisely identify which reasons justify the access is what distinguishes accountable systems from compliant ones. The means by which our methodology itemizes constraints and priorities helps software engineers achieve accountability by this definition.

The extracted constraints are indicative of additional requirements that stakeholders must satisfy before using or disclosing information. To demonstrate due diligence in software design and implementation, software engineers must reason about the necessary steps to satisfy these constraints. We provide a general catalog of constraints that distinguishes between nonephemeral and ephemeral constraints. *Nonephemeral constraints* are satisfiable by information that can be maintained across multiple transactions, whereas *ephemeral constraints* heavily depend on circumstances specific to a single transaction. Finally, we pose several questions that software engineers might ask about how a few of these constraints can be satisfied.

Among the total 1,894 constraints acquired from the Privacy Rule, 1,033 of these were parameterized constraints, as described in Section 4.3. Parameterized constraints that describe the subject, action, or object of access are nonephemeral by nature and are amenable to hierarchical or role-based classification and reasoning. This allows software engineers to classify users or data and then reason about their privileges within an information system across multiple transactions [39]. The act of classification often requires performing additional steps to authenticate the classification, e.g., by checking that a medical examiner is registered with an appropriate state board before conferring that role to a particular system user. Regardless, it is assumed that, once assigned, this role will persist across multiple transactions until revoked at a later time.

Among the 861 nonparameterized constraints, 235 were nonephemeral classifications, which means that the classification is determined by the actions regularly performed by a stakeholder, the physical content of the data, or the time that the data was created. The remaining 626 constraints require additional refinement and engineering on the part of software engineers before software systems can test their satisfiability. The nonparameterized constraints are catalogued and discussed as follows:

1. Stakeholder Beliefs and Determinations,
2. Contractual Statements,
3. Data Subjects, and
4. Intended and Inferred Purposes.

5.1.1 Beliefs and Determinations

A total of 431 constraints that were extracted from the Privacy Rule are satisfied by stakeholder beliefs and determinations. We further classify them into three nondisjoint subsets based upon legal training, medical training, or personal beliefs about circumstances that are required to satisfy these constraints. We separately discuss each of these categories in this section.

Legal determinations affect 231 constraints that refer to existing laws, statutes, or regulations. Of these, only 33 refer to specific laws. The other 198 constraints refer to activities that are required or authorized by laws or organizational charters, leaving it up to the stakeholder to identify which legal documents are relevant. In these situations, fully accountable transactions must identify and record which laws affect the satisfiability of those constraints. In either case, to decide satisfiability, these constraints require knowledgeable stakeholders who have an interpretation of the law that is defensible in court. Table 1 illustrates six example constraints, some of which refer to specific laws, whereas others refer to laws in general.

In Table 1, the constraint from §164.512(b)(1)(ii) applies to a disclosure in which the CE must decide if the recipient of the disclosure is authorized by law to receive reports of child abuse or neglect. The terms of this authorization are relevant to specific public health activities that are being performed by the recipient at the time of access. At that time, a legal determination identifies which laws, if any, authorize the receipt of such reports. Presumably, the CE retains legal counsel to make this determination. If the CE should catalog these authorized activities and the laws that govern them, in advance, they could conceivably automate the legal determinations for these transactions. As part of a

TABLE 2
Medical Determinations

Paragraph	Property	Value
164.510(b)(4)	Subject	Who determines the use and disclosure is necessary to respond to an emergency circumstance.
164.512(b)(1)(iv)	Target	Who may have been exposed to a communicable disease.
164.512(b)(1)(v)(B)	Object	Which concerns a work-related illness or injury.
164.512(c)(1)(iii)(B)	Subject	Who determines the individual is incapacitated.
164.512(k)(5)(B)	Subject	Who represents that the PHI is necessary for the health and safety of such individual or other inmates.
164.524(a)(3)(i)	Action	Which an LHP determines is reasonably likely to endanger the life or physical safety of the individual.

TABLE 3
Personal Beliefs and Determinations

Paragraph	Property	Value
164.502(j)(1)	Subject	Who believes in good faith the CE engaged in unlawful conduct, violates professional standards, or potentially endangers others.
164.506(a)(3)(i)(C)	Subject	Who determines the consent of the individual is inferred from the circumstances.
164.510(b)(3)	Subject	Who determines the disclosure is in the best interest of the individual.
164.510(b)(3)	Subject	Who determines the individual is not present.
164.512(c)(1)	Subject	Who believes the individual of the PHI is a victim of abuse, neglect or domestic violence.
164.512(f)(4)	Subject	Who believes the PHI constitutes evidence of criminal conduct on the premises of the CE.

transaction, if a recipient declares that they require access to PHI to fulfill the needs of an activity authorized by law, known and catalogued a priori, then the access could proceed without requiring a new legal determination at the time of access. The HIPAA Privacy Rule, however, does not collate these activities and associated laws, making the effort to automate this procedure duplicitous, redundant, and expensive for the 545,000 entities governed by HIPAA [16].

Medical determinations that are required to authorize or deny access to information appeared in 184 constraints. These determinations include identifying dangers to physical safety, work-related illness, exposures to specific diseases, emergency treatment situations, and incapacitation of individuals. Only three of these 184 constraints explicitly require a licensed healthcare professional to make the determination. The others require additional analysis to know who makes the medical determination. Table 2 shows six example constraints that require medical determinations. Among these examples, the object constraint from §164.512(b)(1)(v)(B) classifies information based on its content. This type of constraint is nonephemeral because these classifications can be maintained across multiple transactions. The subject constraint from §164.510(b)(4) and the action constraint from §164.524(a)(3)(i) are ephemeral because they must be individually satisfied for each transaction.

Personal beliefs and determinations of stakeholders are used to decide satisfiability in 71 constraints. These beliefs include that disclosures can be used to lessen threats to safety or to apprehend criminals or are in the best interest of the individual, that individuals are victims or perpetrators of crimes, that consent or the lack of objection to a disclosure is inferable from specific circumstances, and that a person is not present. In some cases, these constraints may be construed to imply a need for expert legal or medical knowledge. For example, evaluating whether or not an event constitutes a crime or whether a disclosure would lessen threats to safety has degrees of accuracy that improve with specialized training in law or medicine, respectively. The context in which these constraints were extracted, however, suggests that these determinations are made to the best of the ability of the stakeholder. This ambiguity can lead to noncompliant behavior if a stakeholder with inadequate training is permitted to satisfy one of these constraints. Table 3 contains six example constraints that describe personal beliefs and determinations.

5.1.2 Contractual Statements

There are 170 constraints in which stakeholders attest to the receipt of oral or written statements such as consent, authorizations, waivers, etc., to access information. In the case of written statements, the HIPAA Privacy Rule also

TABLE 4
Contractual Statements

Paragraph	Property	Value
164.504(e)(3)(i)	Subject	Who attempts to obtain satisfactory assurances in a memorandum or contract with the Business Associate.
164.506(a)(1)	Subject	Who has obtained the consent of the individual for the disclosure.
164.508(a)(2)	Subject	Who obtains a valid authorization.
164.512(i)(1)(i)	Subject	Who obtains an alteration or waiver of an individual's required authorization.
164.522(a)(1)(iii)	Subject	Who has an agreement with an individual to restrict disclosures of PHI.
164.524(c)(2)(ii)(B)	Target	Who agrees to the fees imposed for the summary of the PHI.

TABLE 5
Data Subjects

Paragraph	Property	Value
164.502(j)(2)(i)	Object	Which is about the suspected perpetrator of the criminal act.
164.506(a)(2)(ii)	Object	From an individual who is an inmate.
164.512(f)(4)	Object	Which is about an individual who has died.
164.512(k)(1)(i)	Object	About individuals who are Armed Forces personnel.
164.512(k)(1)(ii)	Object	About individuals who are Armed Forces personnel who have been separated or discharged from military service.
164.512(f)(3)	Subject	Who receives a request from the law enforcement official to receive PHI about an individual who is or is suspected to be a victim of a crime.

includes requirements that detail the minimum required content of such statements. These requirements can be used to derive data schemas for electronically recording and maintaining this information. In §164.512(e), e.g., the CE may disclose PHI to a judicial or administrative court if he/she receives satisfactory assurances from the court, documented in the form of written claims, that include 1) provision of notice to the individual of the requested PHI that the court is requesting the PHI, 2) ensuring that the notice contains sufficient information to allow the individual to raise an objection to the request, and 3) permitting the individual sufficient time to raise an objection. These three claims, although standard for this type of disclosure, may have different supporting evidence (e.g., the mailing address of the individual, the content of the notice, and the time allotted for objections) in different situations. Although the court bears the burden of providing these assurances, the separate burden of maintaining this assurance for a period of 6 years lies with the CE who discloses the PHI (see paragraphs (j)(1)(ii) and (j)(2) in §164.530). Thus, satisfying these and similar constraints corresponds to receiving such claims in written or electronic format and retaining them as necessary. Table 4 includes six example constraints that describe contractual statements.

5.1.3 Data Subjects

Data subjects are the people about whom information is collected, maintained, and transferred. In 42 constraints, the data subject was identified by a concept or a role in an activity. Of these constraints, 85.6 percent were assigned to the object property in this study. Table 5 presents six example constraints that illustrate from where data subjects are identified. These constraints are important because they limit the scope of access to specific sets of information based upon who the information is about. In addition to classifying the stakeholders who provide and receive information, software engineers must associate data subjects with information and account for the classifications of data subjects to satisfy these constraints. Moreover, as these classifications change (e.g., inmates are released from custody and military personnel are discharged), systems must respond by updating the respective assigned stakeholder classifications accordingly.

5.1.4 Intended and Inferred Purposes

The purpose of a transaction is an action for which data may be used. These purposes are an increasingly important issue in information security [4], [5], [6]. In traditional Role-Based Access Control (RBAC) systems [39], stakeholders are permitted or denied access to information based on the job functions that they perform, called *roles*. Apart from noting that roles are assigned to users, whereas purposes are assigned to data, roles (e.g., as job functions or actions performed by actors) often imply data purposes (e.g., actions for which data is used). In this study, purposes are stated with respect to the act of access or as a constraint

TABLE 6
Intended and Inferred Purposes

Paragraph	Property	Value
164.514(c)(2)	Object	Which can be used to re-identify de-identified PHI.
164.524(a)(1)(ii)	Object	Which is compiled for use in a civil, criminal or administrative proceeding.
164.512(f)(4)	Purpose	For alerting law enforcement to the death of the individual.
164.514(e)(1)	Purpose	For marketing.
164.512(k)(6)(i)	Subject	Who administers a government program providing public benefits.
164.512(h)	Target	Who is engaged in procurement, banking, or transplantation of cadaveric organs, eyes, or tissue.

TABLE 7
Catalog of Nonparameterized Constraints

Constraint Classification	Total	L	M	B	C	S
Total Beliefs and Determinations	431	231	184	71	73	11
– Legal Determinations (**L**)	231	231	15	26	37	4
– Medical Determinations (**M**)	184	15	184	19	27	4
– Personal Beliefs (**B**)	71	26	19	71	9	3
Total Contractual Statements (**C**)	170	37	27	9	170	4
Total Data Subjects (**S**)	42	4	4	3	4	42
Total Intended and Inferred Purposes (**P**)	389	109	122	18	25	2
– Inferred from Stakeholder Constraints	74	45	25	18	25	2
– Inferred from Objects	8	0	1	0	0	0

on the subject, object, or target properties. The purposes inferred from subject and target constraints are equivalent to roles because they describe actions performed by the affected stakeholders. The purposes expressed in an object constraint denote in which actions the information may be used. Table 6 provides six examples: two purposes stated on the object, two purposes stated on the act itself, and two purposes stated as roles (the subject and target properties).

All 389 constraints that describe valid purposes appear in nonparameterized pattern constraints. Among these, a total of 307 constraints explicitly state intended purposes, eight constraints were assigned to object properties, and, for roles, 34 and 40 constraints were inferred from subject and target properties, respectively.

Purposes present an exceptional challenge to software engineers who intend to guarantee that data is only used for intended purposes. Intended purposes provide explicit motivation for limiting retention, whereas inferred purposes provide insufficient cause for expiring data within a software system. In Table 6, the purpose in the second object property from §164.524(a)(1)(ii) and the purposes in the two purpose properties all describe the intended purpose for which the data is to be used or disclosed. When the purpose is fulfilled, further retaining this data is likely unnecessary. For the inferred purposes in the subject, target, and the remaining object properties, however, it is uncertain if other potential purposes are also intended for the data.

5.1.5 *Summary of Constraint Catalog*

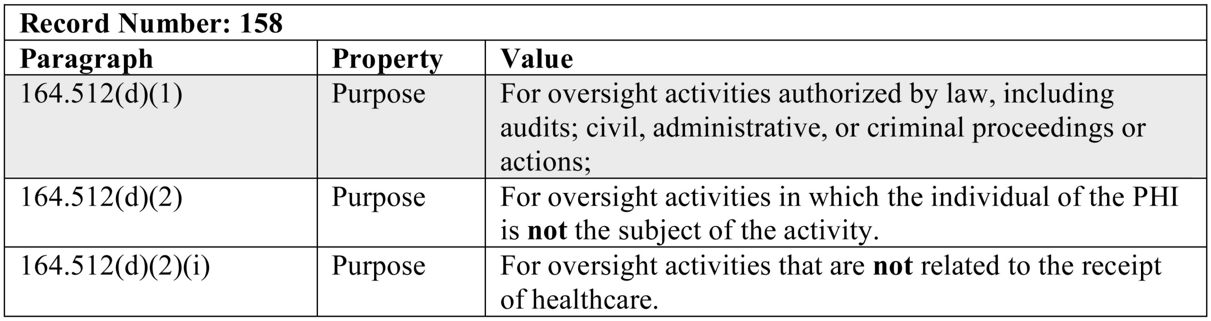
Table 7 provides a summary of the constraint catalog and illustrates how constraints share multiple classifications. The five columns to the right of the Total column show the number of constraints for each classification that are also classified as legal (L) and medical (M) determinations, personal beliefs (B), contractual statements (C), and data subjects (S). Because the constraints reported in Table 1 are nonparameterized, they can essentially contain "constraints upon constraints" that exhibit characteristics from multiple categories. Although not parameterizing these constraints definitely saves time and effort for an engineer, nonparameterized constraints will inherently be susceptible to this multicategorical ambiguity.

5.2 Handling Exceptions and Priorities

An exception is a special constraint that excludes interpretations from a set of properties or rules in a regulation. *Property-based* (e.g., subjects or objects) exceptions are handled by negation, whereas *rule-based* exceptions are handled by priorities. Exceptions that are negated are addressed at the time that the rules are extracted. For example, in §;164.512(d)(2), a property-based exception is stated as follows:

> For the purpose of disclosures permitted by paragraph (d)(1) of this section, a health oversight activity does not include . . .

The health oversight activity is the purpose of one or more rights to disclose information that is described in



Record Number: 158		
Paragraph	**Property**	**Value**
164.512(d)(1)	Purpose	For oversight activities authorized by law, including audits; civil, administrative, or criminal proceedings or actions;
164.512(d)(2)	Purpose	For oversight activities in which the individual of the PHI is **not** the subject of the activity.
164.512(d)(2)(i)	Purpose	For oversight activities that are **not** related to the receipt of healthcare.

Fig. 5. Record with negated-property-based exceptions.

paragraph (d)(1). This exception lists other purposes that these rights must exclude. Consequently, the constraints for the excluded purposes are first extracted from the remaining text in paragraph (d)(2) and then negated before they are added to each right extracted from paragraph (d)(1). Fig. 5 illustrates the partial record for rule 158 that was extracted from paragraph (d)(1) and crossed with the negated constraints extracted from paragraph (d)(2). The shaded row is an intended purpose and the nonshaded rows are the excluded purposes. The English conjunction "not" is in bold to illustrate the negation.

On the other hand, rule-based exceptions prioritize the application of one rule over another in an otherwise ambiguous context. Similar priorities have been used in access control systems to establish open or closed security models [38]. For example, the closed (e.g., deny-first, allow-later) model prioritizes allow rules above a general deny rule. In this situation, the allow rules are the exception: If no rule permits access, then access is always denied. This is the model used in the HIPAA Privacy Rule with the most general and lowest priority deny rules stated in §164.502. Notably, there are several exceptions to these rules that allow access and further exceptions to those allow rules that deny access. Moreover, for accountability purposes, exceptions are important because they may incur additional constraints and follow-on obligations that the stakeholder must satisfy which do not appear in a lower priority rule.

Fig. 6 illustrates 12 of the 58 rule-based exceptions that we extracted from the HIPAA Privacy Rule. These 12 exceptions are comprised of 66 priorities between rules that govern the use and disclosure of information. The boxes contain extracted rule numbers and brief descriptions of those rules in parenthesis. White boxes represent allow rules, whereas shaded boxes represent deny rules. The arrows denote a priority and lead from lower priority rules to higher priority rules. Higher priority rules are the "exceptions." Rules 1 and 2 are the lowest priority deny rules relative to all other extracted rules in the deny-first, allow-later scenario depicted in the HIPAA Privacy Rule.

Priorities between allow rules affect the types of constraints that must be satisfied prior to a permitted disclosure and any follow-on obligations (e.g., postconditions) incurred by disclosing information under those rules. For example, rule 183 denies disclosures of DNA to law enforcement (LE) unless the object of the disclosures is limited to a subset of descriptive features such as physical characteristics permitted by rule 182. Rules 184 and 185 permit disclosures to (LE) for reporting suspected victims of crimes: The agreement of the individual is not required if he/she is incapacitated. Rules 139-142 address specific issues of domestic abuse: Rule 139 provides a general exception pursuant to other unspecified laws, whereas rules 140-142 require agreement from the individual only if the individual is not incapacitated or if the CE believes that the disclosure will prevent further harm to the individual. Unlike rules 184 and 185, rules 139-142 incur the follow-on obligation to notify the individual of the disclosure if his/her agreement to the disclosure was not obtained. Presumably, this obligation of notifying the individual is specific to the nature of domestic violence crimes. Rules 116, 117, and 178, however, do not require past agreement or future notification for the case of reporting child abuse and gunshot wounds, respectively.

We extracted 58 priorities from the HIPAA Privacy Rule by using 11 unique natural-language priority patterns listed in Table 8. These patterns consist of an exception phrase that coordinates two other reference phrases, labeled *Higher* and *Lower* in Table 8, that correspond to sets of extracted rules. The rules that match the *Higher* phrase have a higher priority than the rules that match the *Lower* phrase. The table also lists each priority pattern's frequency of occurrence in the Privacy Rule.

Reference phrases either describe extracted rules or are cross references to other paragraphs in the regulation from which rules were extracted. In the latter case, these cross references may further be restricted using supporting phrases. For example, the supporting phrase "as permitted by" refers to rights, whereas the supporting phrase "as required by" refers to obligations. These supporting phrases may also include parameterized and nonparameterized constraints that must be used to screen rules extracted from other paragraphs. For example, the supporting phrase may refer to "disclosures" in another paragraph that denote rules in which the action is "disclose"; thus, rules from that paragraph with other actions such as "use" may be ignored when recording a corresponding priority.

6 Summary

Increasingly, regulations are requiring software engineers to specify, design, and implement systems that are accountable to and in compliance with laws and regulations. These regulations describe stakeholder rules, called rights and obligations, which are often precursors to functional software requirements. These precursors must undergo extensive analysis and refinement before they can be implemented. To support this effort, we have developed

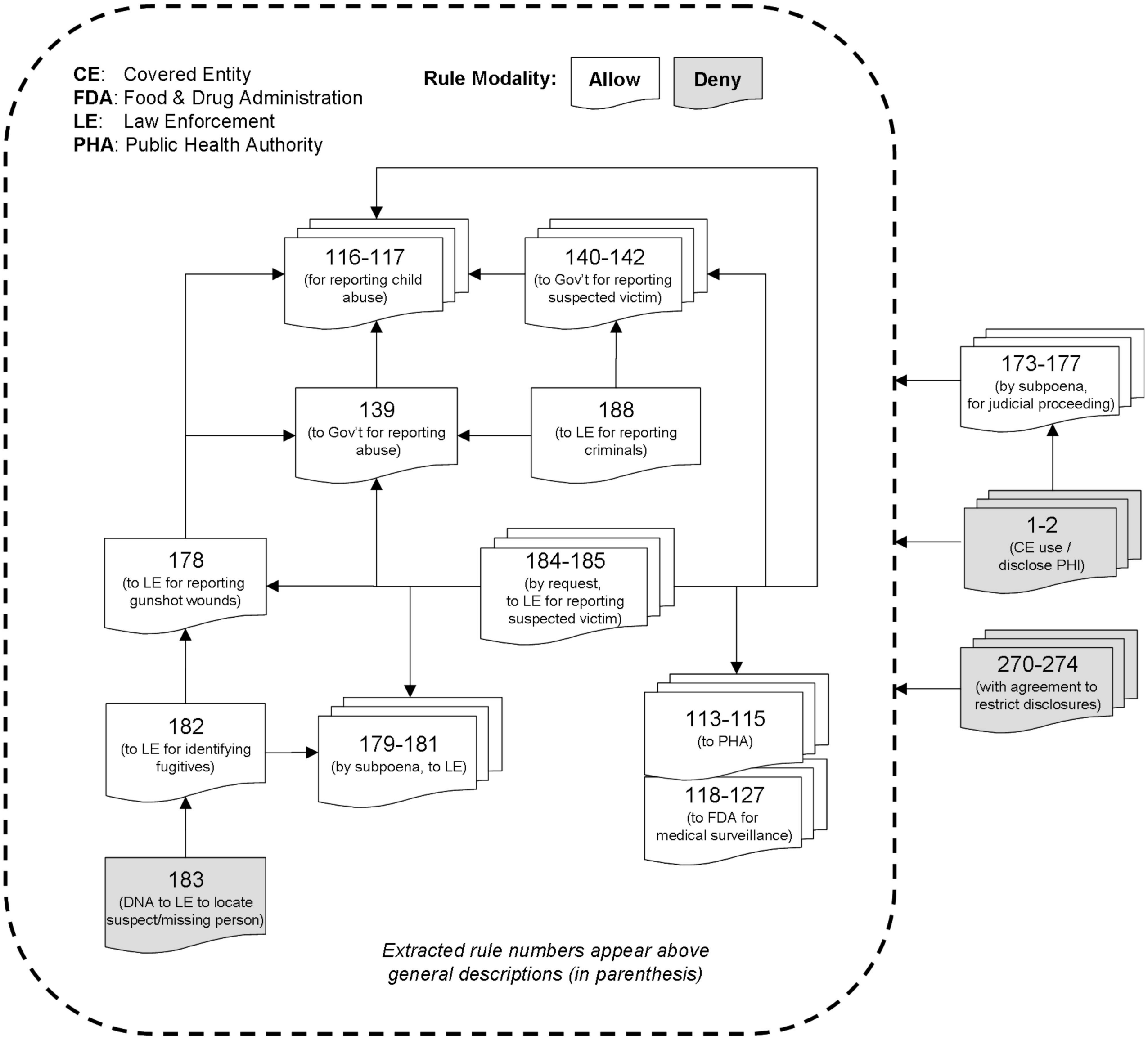

CE: Covered Entity
FDA: Food & Drug Administration
LE: Law Enforcement
PHA: Public Health Authority
Rule Modality:
Allow
Deny
116-117 (for reporting child abuse)
140-142 (to Gov't for reporting suspected victim)
173-177 (by subpoena, for judicial proceeding)
139 (to Gov't for reporting abuse)
188 (to LE for reporting criminals)
1-2 (CE use / disclose PHI)
178 (to LE for reporting gunshot wounds)
184-185 (by request, to LE for reporting suspected victim)
270-274 (with agreement to restrict disclosures)
182 (to LE for identifying fugitives)
179-181 (by subpoena, to LE)
113-115 (to PHA)
118-127 (to FDA for medical surveillance)
183 (DNA to LE to locate suspect/missing person)
Extracted rule numbers appear above general descriptions (in parenthesis)

Fig. 6. Example network of priorities between rules.

a methodology for extracting access rights and obligations from regulatory texts to ensure statement-level coverage for an entire regulation [13]. The method provides guidance to software engineers for creating stakeholder hierarchies, identifying six types of constraints on requirements, managing cross references, maintaining traceability, and resolving ambiguities. In this paper, we present extensions to this methodology to acquire data elements and assign law-preserving priorities between data requirements to prevent improper information disclosures. The extended methodology provides critical assistance to engineers in navigating a very complex set of constraints and requirements as expressed in regulations. The entire methodology has been developed over the course of several years by using the Grounded Theory [21] and has been validated using a substantial body of work.

TABLE 8
Patterns for Identifying Priorities

Frequency	Priority Pattern
19	*Lower*, except as permitted by *Higher*
1	*Lower*, except as authorized under *Higher*
9	*Lower*, except as required by *Higher*
4	*Lower*, except for *acts* pursuant to *Higher*
4	*Lower* does not apply to *Higher*.
1	*Higher*, without meeting the requirements of *Lower*
7	*Lower*, except as provided by *Higher*
4	Notwithstanding *Higher*, *Lower*.
5	Other than *Higher*, *Lower*.
1	*Lower* does not supersede *Higher*.
3	*Lower* is not effective under *Higher*.

Although our extended methodology has been applied to a large US regulation that governs information privacy in the healthcare domain, we believe that the two extensions, which include 1) the method for acquiring data elements and 2) the method for prioritizing data access requirements, can be used to analyze other regulations that govern information-intensive software systems. However, because the first extension requires that the engineer map nonparameterized constraints to one of the six constraint types (e.g., subject, action, or object), this extension will require additional work when all of the entities that appear in the nonparameterized constraint do not correspond to any of the values in the parameterized constraints. Based on our observations, this situation is only theoretical; however, more work is needed to understand the scope of this potential limitation to future work. On the other hand, we have observed numerous cross references in other US regulations that include exceptions to regulatory rules. These widespread cross references provide compelling evidence to believe that the second extension for prioritizing data access requirements can be applied to other US regulations that govern information systems. Future work is needed to assess this methodology on regulations outside the US. Finally, the constraint catalog classifies constraints from the HIPAA Privacy Rule into four categories:

1. beliefs and determinations,
2. contractual statements,
3. data subjects, and
4. intended and inferred purposes.

Although the constraint class for data subjects is most relevant to access control systems, the other three classes describe stakeholder actions that more likely appear in other regulations such as financial, insurance, and environmental laws. As we discuss next, more work is needed to determine if constraints in these classes share common strategies for refinement into verifiable functional requirements.

Because regulations that govern information systems are written to broadly govern industry-wide business practices, these regulations are mostly nonfunctional in nature. This observation is supported, in part, by two reasons: 1) These regulations are written to support marketplace diversity by intentionally offering broad interpretations that affect a variety of related nonspecific business practices and 2) they regularly describe the actions of stakeholders and less frequently describe the structure or processing of data that *may or may not* occur in support of those actions. To help businesses and the government reach agreement on how compliance can be verified with regulations, our future work includes developing a method to identify criteria for evaluating functional requirements derived from nonfunctional regulations. In addition to comprising a set of "best practices," these criteria should demonstrate to law and policy makers the efficacy of achieving compliance with a specific regulation under the restrictions of available technology and other resources.

To our knowledge, this work is the first attempt within the software engineering community to comprehensively analyze an entire regulation for the purpose of specifying system requirements that are accountably compliant with the law. The danger of not employing a systematic methodology is that it leaves organizations susceptible to security breaches. Furthermore, organizations which can systematically demonstrate how their software systems comply with policies and regulations can more effectively demonstrate due diligence and standard of care.

Acknowledgments

The authors wish to thank the members of ThePrivacy Place.org for their helpful comments and feedback. This research was supported, in part, by two grants from the US National Science Foundation—*ITR: Encoding Rights, Permissions and Obligations: Privacy Policy Specification and Compliance* (NSF 0325269) and *Cyber Trust Collaborative Research: A Comprehensive Policy-Driven Framework for Online Privacy Protection: Integrating IT, Human, Legal and Economic Perspectives* (NSF 0430166)—by the Center for Education and Research in Information Assurance and Security (CERIAS), Purdue University, and by the IBM PhD Fellowship awarded by the IBM Center for Advanced Studies, Research Triangle Park, North Carolina.

References

[1] A.I. Antón, "Goal-Based Requirements Analysis," *Proc. Second IEEE Int'l Conf. Requirements Eng.*, pp. 136-144, 1996.

[2] A.I. Antón, J.B. Earp, Q. He, W. Stufflebeam, D. Bolchini, and C. Jensen, "Financial Privacy Policies and the Need for Standardization," *IEEE Security and Privacy*, vol. 2, no. 2, pp. 36-45, Mar./Apr. 2004.

[3] A.I. Antón and J.B. Earp, "A Requirements Taxonomy for Reducing Web Site Privacy Vulnerabilities," *Requirements Eng.*, vol. 9, no. 3, pp. 169-185, 2004.

[4] P. Ashley, C. Powers, and M. Schunter, "From Privacy Promises to Privacy Management: A New Approach for Enforcing Privacy throughout the Enterprise," *Proc. 10th New Security Paradigms Workshop*, pp. 43-50, 2002.

[5] P. Ashley, S. Hada, G. Karjoth, and M. Schunter, "E-P3P Privacy Policies and Privacy Authorization," *Proc. ACM Workshop Privacy in the Electronic Soc.*, pp. 103-109, 2002.

[6] J-W. Byon, E. Bertino, and N. Li, "Purpose-Based Access Control of Complex Data for Privacy Protection," *Proc. 10th ACM Symp. Access Control Models and Technologies*, pp. 102-110, 2005.

[7] T.D. Breaux and A.I. Antón, "Deriving Semantic Models from Privacy Policies," *Proc. Sixth IEEE Int'l Workshop Policies for Distributed Systems and Networks*, pp. 67-76, 2005.

[8] T.D. Breaux and A.I. Antón, "Analyzing Goal Semantics for Rights, Permissions and Obligations," *Proc. 13th IEEE Int'l Conf. Requirements Eng.*, pp. 177-186, 2005.

[9] T.D. Breaux and A.I. Antón, "Mining Rule Semantics to Understand Legislative Compliance," *Proc. ACM Workshop Privacy in the Electronic Soc.*, pp. 51-54, 2005.

[10] T.D. Breaux, A.I. Antón, C-M. Karat, and J. Karat, "Enforceability vs. Accountability in Electronic Policies," *Proc. Seventh IEEE Int'l Workshop Policies for Distributed Systems and Networks*, pp. 227-330, 2006.

[11] T.D. Breaux and A.I. Antón, "Semantic Parameterization: A Conceptual Modeling Process for Domain Descriptions," Technical Report TR-2006-35, Dept. of Computer Science, North Carolina State Univ., Oct. 2006, *ACM Trans. Software Eng. Methods*, to appear.

[12] T.D. Breaux, A.I. Antón, and E.H. Spafford, "A Distributed Requirements Management Framework for Compliance and Accountability," Technical Report TR-2006-14, Dept. of Computer Science, North Carolina State Univ., July 2006.

[13] T.D. Breaux, M.W. Vail, and A.I. Antón, "Towards Compliance: Extracting Rights and Obligations to Align Requirements with Regulations," *Proc. 14th IEEE Int'l Conf. Requirements Eng.*, pp. 49-58, 2006.

[14] T.D. Breaux and A.I. Antón, "Impalpable Constraints: Framing Requirements for Formal Methods," Technical Report TR-2007-6, Dept. of Computer Science, North Carolina State Univ., Feb. 2007.

[15] C. Brodie, C-M. Karat, J. Karat, and J. Feng, "Usable Security and Privacy: A Case Study of Developing Privacy Management Tools," *Proc. First Symp. Usable Privacy and Security*, pp. 35-43, 2005.

[16] "Health Care," *Career Guide to Industries, 2006-2007.* Bureau of Labor Statistics, US Dept. of Labor, 2007.

[17] C.J. Hoofnagle and D.J. Solove, "Re: Request for Investigation into Data Broker Products for Compliance with the FCRA," Electronic Privacy Information Center, 2004.

[18] C.B. Farrell, "ChoicePoint Settles Data Security Breach Charges: To Pay $10 Million in Civil Penalties and $5 Million for Customer Redress," FTC File 052-3069, Office of Public Affairs, US Fed. Trade Commission, 2006.

[19] United States v. ChoicePoint, Inc., Case 1:06-CV-00198-JTC, (Northern District of Georgia), Feb. 2006.

[20] *Black's Law Dictionary,* B.A. Garner, ed., eighth ed., 2004.

[21] B.C. Glaser and A.L. Strauss, *The Discovery of Grounded Theory.* Aldine Publishing, 1967.

[22] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "Modeling Security Requirements through Ownership, Permission and Delegation," *Proc. 13th IEEE Int'l Conf. Requirements Eng.,* pp. 167-176, 2005.

[23] M. Jackson, "The World and the Machine," *Proc. 17th IEEE Int'l Conf. Software Eng.,* pp. 283-292, 1995.

[24] M. Jackson and P. Zave, "Domain Descriptions," *Proc. First IEEE Symp. Requirements Eng.,* pp. 56-64, 1993.

[25] "HIPAA Administrative Simplification: Enforcement—Parts 160 and 164," *Federal Register,* US Dept. of Health and Human Services, vol. 71, no. 32, pp. 8389-8433, Feb. 2006.

[26] C.B. Haley, R.C. Laney, J.D. Moffett, and B. Nuseibeh, "The Effect of Trust Assumptions on the Elaboration of Security Requirements," *Proc. 12th IEEE Int'l Conf. Requirements Eng.,* pp. 102-111, 2004.

[27] C.B. Haley, R. Laney, and B. Nuseibeh, "Deriving Security Requirements from Crosscutting Threat Descriptions," *Proc. Third Int'l Conf. Aspect-Oriented Software Development,* pp. 112-121, 2004.

[28] C.B. Haley, J.D. Moffett, R. Laney, and B. Nuseibeh, "Arguing Security: Validating Security Requirements Using Structured Argumentation," *Proc. Third Symp. Requirements Eng. for Information Security,* 2005.

[29] J.F. Horty, *Agency and Deontic Logic.* Oxford Univ. Press, 2001.

[30] "Standards for Privacy of Individually Identifiable Health Information—Part 164, Subpart E," *Federal Register,* US Dept. of Health and Human Services, vol. 68, no. 34, pp. 8334-8381, Feb. 2003.

[31] "Standards for the Protection of Electronic Protected Health Information—Part 164, Subpart C," *Federal Register,* US Dept. of Health and Human Services, vol. 68, no. 34, pp. 8334-8381, Feb. 2003.

[32] A. van Lamsweerde, "Elaborating Security Requirements by Construction of Intentional Anti-Models," *Proc. 26th IEEE Int'l Conf. Software Eng.,* pp. 148-157, 2004.

[33] S-W. Lee, R. Gandhi, D. Muthurajan, D. Yavagal, and G-J. Ahn, "Building Problem Domain Ontology from Security Requirements in Regulatory Documents," *Proc. Second Int'l Workshop Software Eng. for Secure Systems,* pp. 43-50, 2006.

[34] L. Lin, B. Nuseibeh, D. Ince, M. Jackson, and J. Moffett, "Introducing Abuse Frames for Analyzing Security Requirements," *Proc. 11th IEEE Int'l Conf. Requirements Eng.,* pp. 371-372, 2003.

[35] M.J. May, C.A. Gunter, and I. Lee, "Privacy APIs: Access Control Techniques to Analyze and Verify Legal Privacy Policies," *Proc. 19th IEEE Computer Security Foundations Workshop,* pp. 85-97, 2006.

[36] J. Mylopoulos, L. Chung, and E. Yu, "From Object-Oriented to Goal-Oriented Requirements Analysis," *Comm. ACM,* vol. 42, no. 1, pp. 31-37, 1999.

[37] J. Reagle and L.F. Cranor, "The Platform for Privacy Preferences," *Comm. ACM,* vol. 42, no. 2, pp. 48-55, 1999.

[38] P. Samarati and S. de Capitani di Vimercati, "Access Control: Policies, Models and Mechanisms," *Foundations of Security Analysis and Design,* vol. 2171, pp. 137-193, 2001.

[39] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, "Role-Based Access Control Models," *Computer,* vol. 29, no. 2, pp. 38-47, Feb. 1996.

[40] T. Verhannenman, F. Piessens, B. de Win, and W. Joosen, "Requirements Traceability to Support Evolution of Access Control," *Proc. First Workshop Software Eng. for Secure Systems,* pp. 1-7, 2005.

[41] D. Xu, V. Goel, and K. Nygard, "An Aspect-Oriented Approach to Security Requirements Analysis," *Proc. 30th Ann. Int'l Computer Software and Applications Conf.,* pp. 79-82, 2006.

[42] P. Zave and M. Jackson, "The Four Dark Corner's of Requirements Engineering," *ACM Trans. Software Eng. Methods,* vol. 6, no. 1, pp. 1-30, 1997.

[43] *HIPAA Medical Privacy and Transition Rules: Overkill or Overdue?,* Hearing before the Special Committee on Aging, US Senate, 108th Congress, Ser. 108-23, 23 Sept. 2003.

[44] Extensible Access Control Markup Language (XACML) Version 2.0, Oasis Standards Group, Feb. 2005.

Travis D. Breaux received the BA degree in anthropology from the University of Houston in 1999 and the BS degree in computer science from the University of Oregon in 2003. He is currently working toward the PhD degree in the Department of Computer Science at North Carolina State University. His research interests include requirements engineering for software systems that are regulated by policies and laws. He is a recipient of the 2006-2008 IBM PhD Fellowship, 2006-2007 Walker H. Wilkinson Research Ethics Fellowship, and 2005-2006 CISCO Information Assurance Scholarship. He is a student member of the IEEE and the IEEE Computer Society, ACM SIGSOFT, and the ACM US Public Policy Committee.

Annie I. Antón received the PhD degree in computer science from the Georgia Institute of Technology, Atlanta, in 1997. In 1998, she joined the faculty of North Carolina State University, Raleigh, where she is currently an associate professor, the founder and director of ThePrivacyPlace.org, and a member of the Cyber Defense Laboratory. Her research interests include software requirements engineering, information privacy and security policy, regulatory compliance, software evolution, and process improvement. She is a cochair of the Privacy Subcommittee of the ACM US Public Policy Committee. She is a member of the ACM, CRA Board of Directors, IAPP, Sigma Xi, and the Data Privacy and Integrity Advisory Committee, US Department of Homeland Security. She is a senior member of the IEEE. She is a recipient of the US National Science Foundation Faculty Early Career Development (CAREER) Award.