Analyzing Regulatory Rules for Privacy and Security Requirements

Travis 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 U.S. 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 be properly 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 are implementable. To support the software engineering effort to derive security requirements from regulations, we present a methodology to extract access rights and obligations directly 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 U.S. Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule.

Index Terms—*Data security and privacy; Law and regulations; Compliance; Accountability; Requirements engineering.*

I. Introduction

Increasingly, regulations in Canada, Europe and the United States are governing the use and disclosure of information in both industry and government. This presents different challenges to information systems that support established or emerging business practices. In the U.S. for example, 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 ten-year cost to comply with HIPAA for the healthcare industry is projected by industry and government

* T.D. Breaux and A.I. Antón are with the Computer Science Department, North Carolina State University, Raleigh, NC, 27695-8207 USA (e-mail: {tdbreaux,aianton}@ncsu.edu).

This research was supported, in part, by two grants from the 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), as well as the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University and the IBM PhD Fellowship, Center for Advanced Studies, Raleigh, NC.

[1] U.S. Pub. Law 104-191, est. 1996.

stakeholders to be between $12-$42 billion dollars [10823].

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 change, new information-driven business models may not be adequately vetted before they are put into practice. For example, the U.S. 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 acknowledged that records on more than 163,000 consumers were acquired by identity thieves [Far06]. 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 [EPIC04]. 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 [Far06, FTC06]. 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 [EPIC04].

To support software and requirements engineers, system administrators and policy makers, we developed a methodology to extract formal descriptions of rules governing stakeholder actions from policies and regulations [BVA06]. 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 using the HIPAA Privacy Rule [HPR] to yield 300 rules that govern stakeholder access to medical information. The HIPAA Privacy Rule affects some 545,000 different establishments in the U.S. who employ over 13.5 million people [BLS06]. The contributions presented in this paper are the extended methodology as well as a catalogue of constraint types that resulted from validating the methodology in the HIPAA case study.

The remainder of the paper is organized as follows: in Section II, we review related work and discuss the rule-making process that yielded the HIPAA Privacy Rule; we follow with important terms and definitions in Section III; in Section IV, we present the basic methodology to extract access rights, obligations and constraints with an example from the HIPAA Privacy Rule; in Section V, we present the results of a case study using the Privacy Rule, including a catalogue of access constraints and a review of exceptions; in Section VI, we conclude with discussion and summary.

II. 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 [ZJ97, Jac95]. Moreover, they identify the challenge that engineers face in distinguishing between descriptions of the domain and descriptions of systems [JZ93]; 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 persons make to satisfy legal requirements

[2] U.S. Pub. Law 91-508, est. 1970.

or discharge their legal obligations [Gar04]. In the U.S., *standard of care* means "under the law of negligence or of obligations, the conduct demanded of a person in a situation; typically, this involves a person giving attention both to possible dangers, mistakes and pitfalls and to ways of minimizing those risks" [Gar04]. 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 we propose in this paper will provide part of this important justification, which is necessary to establish due diligence and a reasonable standard of care in software engineering.

In requirements and software engineering, researchers have investigated methods to analyze security requirements using aspects [XGN06], goals [GMM05, Lam04], problem frames [LNI03, HLN04], trust assumptions [HLM04] and structured argumentation [HML05]. More recent work focuses on the rigorous extraction of requirements from security-related policies and regulations [MGL05, BVA06, LGM06]. In earlier work, we presented a methodology to extract stakeholder rights and obligations from regulations [BA05c, BVA06]. Rights and obligations are similar to the notions of "what is permissible" and "what ought to be" as modeled by Deontic Logic [Hor01]. The methodology combines the Goal-based Requirements Analysis Method (GBRAM) [Ant96] and a process called Semantic Parameterization for acquiring formal models from natural language statements [BAD06]. The methodology was validated in a pilot study using a patient fact sheet summarizing the HIPAA Privacy Rule [BA05c] and in a larger case study using four sections of the Privacy Rule concerning privacy notices, requests for access restrictions and patient access to review and amend their medical information [BVA06].

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 [BA05a, BA05b]. The products of this extension are access control elements that include data subjects, objects and purposes, the relevant principals (e.g., authorized actors) and pre-conditions in data access [SV01]. The second extension includes methods to identify, manage and prioritize important exceptions that must be respected to avoid illegal and unauthorized information use and disclosures.

May et al. describe a methodology to extract formal models from regulations that they applied to one section in the HIPAA Privacy Rule [MGL05]. Our work to extract formal models from four sections in the Privacy Rule [BVA06] presents contradictory insight to several of their basic assumptions, including: 1) each paragraph has exactly one rule; and 2) external and ambiguous references are satisfiable by the environment [MGL05]. 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 be inconsistent with the HIPAA Privacy Rule. Lee et al. employ an ontological approach to extract requirements from regulations [LGM06]. Because their approach categorizes requirements using an ontological model, the approach helps engineers rigorously identify inconsistencies between the model and the regulations; this is an improvement over May et al. [MGL05]. To varying degrees, our methodology [BVA06, BAD06] 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 [LGM06] and cross-references [MGL05].

It is important to distinguish access control rules (ACR) from stakeholder rights and obligations pertaining to access. ACRs are triples consisting of a principal, action and object in which the principal is or is not permitted to perform the action on the object [SV01]. 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" or a specific data element such as "an individual's name." What constitutes these objects in software is a non-trivial 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 V.A. Although we could express stakeholder rights and obligations using an access control language, such as the eXtensible Access Control Markup Language (XACML) [XACML], 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 to directly map natural language policies that describe stakeholder rights and obligations into access control rules [BKK05, RC99, VPW95]. While 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.

A. Evolution of the HIPAA

The United States legislation titled the Health Insurance Portability and Accountability Act (HIPAA) was passed in August 1996 for numerous reasons, including the need for increased protection of patient medical records against unauthorized use and disclosure. The HIPAA requires the U.S. Department of Health and Human Services (HHS) to develop, enact and enforce regulations governing electronically managed patient information in the healthcare industry. Consequently, from 1998 to 2006, a special committee in HHS prepared several recommendations based upon extensive expert witness testimony from academia, 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 [HSR].

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 [HPR].

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

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 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 to extract stakeholder rights and obligations from regulations [BVA06]. *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 [BAS06].

III. Terminology and Definitions

The methodology uses the following terms:

- A *definition* is a statement that restricts the meaning of a term using one or more constraints. For example, the statement "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 IV.A.
- 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 IV.B.
- A *constraint* is a statement that restricts or limits the possible interpretations for a concept via one of its properties or via 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 catalogue of constraints in Section V.A.
- A *right* is a statement about one or more actions that a stakeholder is permitted to perform. If a stakeholder is expressly not obligated to perform an action, called an *anti-obligation*, 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 an *anti-right* or *refrainment*, then this statement also describes an obligation.
- A *rule* is either a right, obligation or refrainment per our definitions, above. Rules are often restricted in some way by constraints.
- An *exception* denotes a relationship between two properties or rules, in which all the possible interpretations of the one thing (e.g., 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, lower priority rule would not apply. We discuss exceptions in Section V.B.

IV. Encoding Rules from Regulations

The requirements engineering methodology to encode rules from regulations discussed in this paper was developed using Grounded Theory [GS67], in which observations from a dataset are relevant to that dataset. While 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 this methodology is generalizable beyond HIPAA [Ant96, AEH04, AE04]. We review this methodology in Sections IV.A and IV.B to provide important background and as a foundation to our new extensions for modeling data access requirements presented in Section IV.C.

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 [BVA06]. 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 IV.A, we illustrate the role of definitions in establishing a classification hierarchy of stakeholders, which helps engineers identify which rules apply to their organization and disambiguate them. In Section IV.B, we describe the process for extracting rights, obligations and constraints from regulatory texts using extensively validated natural language patterns [BA05a, BA05b, BA05c, BVA06]. In Section IV.C, we show how to map these rule elements to parameterized rules using six properties, and demonstrate how to derive priorities between these rules from exceptions.

A. Definitions and Stakeholder Hierarchies

Definitions in U.S. Federal and state regulations include terms that describe concepts by their specializations, illustrating 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 having a specific, precise meaning in a given specialty, apart from its general meaning in ordinary contexts" [Gar04]. In these regulations, the concept term may be substituted with any 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" from §160.103 in the HIPAA Privacy Rule:

> "*Covered entity* means: (1) a health plan; (2) a healthcare clearinghouse; or (3) a healthcare 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, healthcare clearinghouse and healthcare provider) that, in conjunction with their role in the act of electronically transmitting health information, are specializations of the term covered entity. 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 Figure 1). The shaded boxes in this hierarchy indicate stakeholders who were identified during the case study described in Section V but who do not have separate definitions in either of the definition sections in the HIPAA Privacy Rule, §160.103 and §164.503.

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 as well as 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 as well as 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 to visually understand which classes of stakeholders are affected by a rule and by supporting formal reasoning about similarities and conflicts between rules [BVA06].

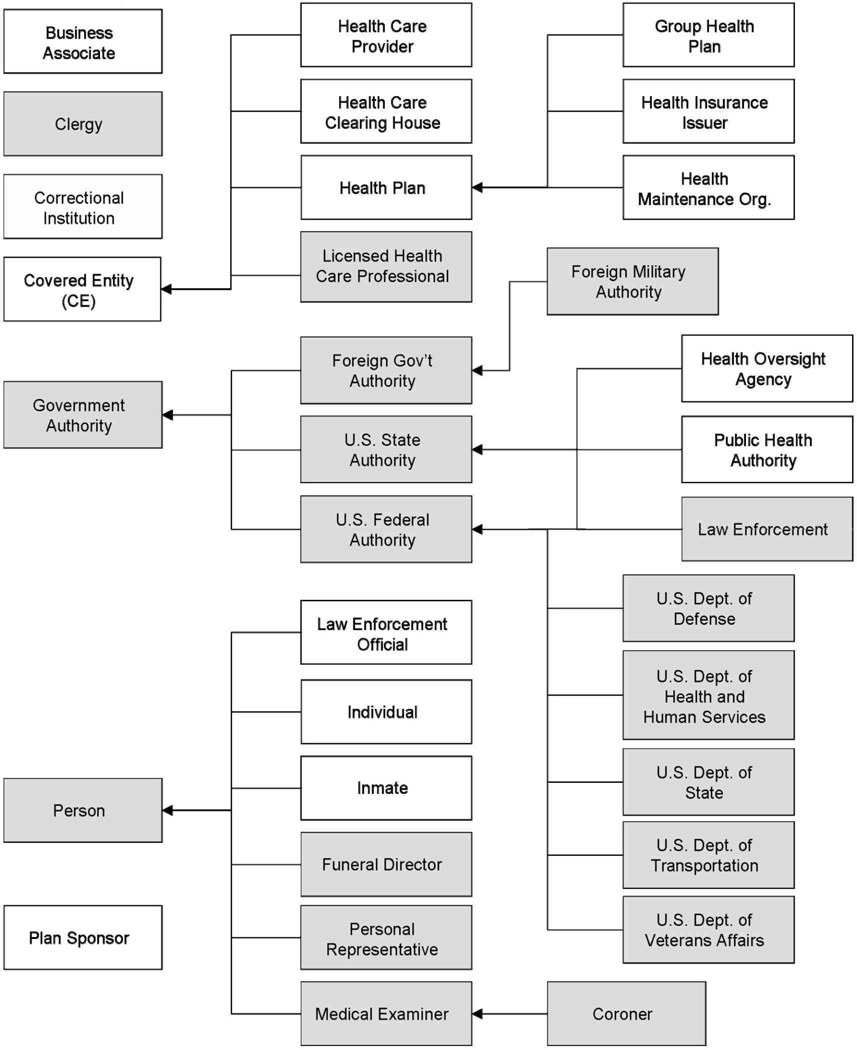

FIGURE 1: STAKEHOLDER HIERARCHY FOR THE HIPAA PRIVACY RULE

B. *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 expressed formally) can be subtle and easily overlooked, we developed several patterns to consistently extract natural language constraints into formal predicates in first-order logic [BA05a, BA05b, BA05c, BAD06]. The patterns we use here include: the *basic activity pattern with modality* [BA05a, BA05b]; *purposes* [BA05b]; *nouns* *distinguished by verb phrases* [BA05b]; and *rules or conditions* [BA05c]. We first present the results of applying these patterns before discussing each pattern in detail. The results are derived from the following two excerpts from the HIPAA Privacy Rule §164.510 and §164.522. These excerpts describe the covered entity (CE), healthcare provider (HCP) and protected health information (PHI). In these excerpts, the rule statement is *italicized* with the in-line modal phrases (must, may, etc.) in ***bold***; the constraints are underlined; and the condition keywords (except, if, 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 below that include C_1-C_{11}, in addition to rules that include obligation $O_{1,}$, refrainments O_2, O_3, the anti-obligation R_2 and the three additional rights R_1, R_3, 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 pre- and post-conditions, 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 to carry 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 to provide emergency treatment to the individual. §164.522(a)(1)(iii).

C_{11}: The disclosure is to provide 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 [BA05a, BA05b, BVA06]. 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 covered entity (subject) that permits (action) the individual to request a restriction (an act which 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 [BA05b]. 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 described separately 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 [BA05b]. 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 covered entity. 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 pre- and post-conditions (constraints) using condition keywords (e.g., if, except, unless, upon, when) [BA05c]. In paragraph (a)(1)(iii), the keyword "if" is followed by two underlined phrases that pre-condition 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 §164.510(b)." We interpret the English conjunction "and" in this statement as a logical-or because the individual may request any one of these restrictions independent of the other. Policymakers and software engineers may have differing views on these conjunctions. For example, using the English conjunction "and," paragraphs §164.512(f)(2)(i)(A) – (H) in the Privacy Rule list eight specific types of information (e.g., name, date of birth, social security number, etc.) 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, §164.502(b)). Interpreting this conjunction as a logical-and simplifies software designs because the software engineer need only 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 as well as 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 paragraph §164.510(b)(1)(i) for example, 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 paragraph §164.522(a)(1)(i)(B), the phrase "disclosures permitted under §164.510(b)" refers exclusively 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 that 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 "(…)" into 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 is 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 we recommend is to extract all 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 they will then copy the previously extracted constraints between corresponding rules. The extensions we now discuss in Section IV.C will help engineers to more quickly isolate only the relevant constraints from cross-references, thus simplifying cross-reference analysis and management during the first and second passes.

C. *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 protected health information [BVA06]. 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 below. These two extensions expose important details that will help software engineers design access control systems. In Section V.A., we discuss how engineers can infer critical requirements for these systems from many of the constraints in this extended format.

In prior work, we describe a process called Semantic Parameterization that is used to map words that describe concepts from simple sentences into first-order predicate logic expressions [BA05a, BA05b, BAD06]. 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 the following 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 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 IV.B has been parameterized and appears in Figure 2. The value “disclose” of the action property is stated both in paragraphs §164.522(a)(1)(i)(B) and (a)(1)(iii); thus the constraint is indexed by both paragraphs 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 paragraph §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, uses) or modalities (e.g., rights or refrainments).

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

Figure 2: Initial Record for Refrainment O_3

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 *non-parameterized constraints* because the constraint statements were not parameterized like the right and obligations statements. Figure 3 shows the non-parameterized constraints derived from C_1, C_2 and C_6, respectively. Each non-parameterized 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, place or a thing, respectively. Constraints statements may refer to multiple entities, for example, the statement C_1 is "the individual identified the person" and describes two entities: the individual and the person. If the non-parameterized 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; a process called *re-topicalization*. Therefore, we re-topicalize the statement C_1 to yield the non-parameterized target constraint value "Who is identified by the individual" on row 6 in Figure 3 that corresponds to the constraint value "Person" for the target property on row 5 in Figure 2.

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)

Figure 3: Continued Record for Refrainment O_3

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 Figure 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 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 non-italicized portion highlights the text from which the lower priority rules were extracted. Figure 4 illustrates an example record from the priority table. The example exception appears in paragraph §164.522(a)(1)(iii) in the excerpt above in Section

IV.B, 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 non-disclosure rules that were extracted in our case study: rules 270-272 that include a unique constraint from the cross-reference to §164.510(b); and rule 274 that 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.

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…*

Figure 4: Record for Priority between Rules 270-272, 274-276.

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 Figure 4 actually prescribes four different priorities between five different rules. In Section V, we present several exception patterns that we used to standardize the identification and interpretation of priorities between rules.

V. Case Study in Information Privacy

The extended methodology in Section IV 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 [HPR] 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 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% of the total rules and 89.6% 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% of the rules and 10.4% 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. While new phrases will inevitably be encountered, our experience with regulatory texts in other domains [BA07] shows that these phrases are often variations on the same elements that we report in this paper, suggesting the methodology is generalizable to domains beyond healthcare.

The 300 extracted rules are expressed in the rule record format from Section IV.C and comprise 1,894 constraints. Several of these constraints contain disjunctions over related concepts; performing case-splitting on these disjunctions, as explained in [BA05b], would increase the number of extracted rules. Whereas 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 over 12,205 priorities between different rules.

A. Catalogue of Constraints 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 [MCY99].

In addition to satisfiability, software engineers must distinguish between compliance and accountability under regulations. A software system is non-compliant 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 non-permissible 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 [BAK06, BAS07]. Improving accountability will demonstrate due diligence and improve compliance, whereas a compliant system may not be accountable at all. As we illustrate in Section V.B, 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 catalogue of constraints that distinguishes between non-ephemeral and ephemeral constraints. *Non-ephemeral 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 to satisfy a few of these constraints.

Among the total 1,894 constraints acquired from the Privacy Rule, 1,033 of these were parameterized constraints as described in Section IV.C. Parameterized constraints that describe the subject, action or object of access are non-ephemeral by nature and 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 [SCF96]. The act of classification often requires performing additional steps to authenticate the classification, for example, 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 non-parameterized constraints, 235 were non-ephemeral classifications, meaning that the classification is determined by the actions regularly performed by a stakeholder, the physical content of the data or by the time 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 non-parameterized constraints are catalogued and discussed as follows: 1) Stakeholder Beliefs and Determinations; 2) Contractual Statements; 3) Data Subjects; and 4) Intended and Inferred Purposes.

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 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 while others refer to laws, in general.

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.

In Table 1, the constraint from §164.512(b)(1)(ii) applies to a disclosure in which the covered entity 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 covered entity retains legal counsel to make this determination. If the covered entity were to catalogue these authorized activities and the laws that govern them, they could conceivably automate the legal determinations for these transactions. As part of a 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 [BLS06].

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 non-ephemeral because these classifications can be maintained across multiple transactions. The action constraint from §164.524(a)(3)(i) and the subject constraint from §164.510(b)(4) are ephemeral because they must be individually satisfied for each transaction.

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.

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, apprehend criminals or are in the best interest of the individual; that individuals are victims or perpetrators of crimes; that consent, or a 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 ability of the stakeholder. This ambiguity can lead to non-compliant 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.

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.

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 includes requirements that detail the minimum required content of such statements. These requirements can be used to derive data schemas for recording and maintaining this information electronically. In §164.512(e) for example, the covered entity may disclose PHI to a judicial or administrative court if they receive 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, while standard for this type of disclosure, in different situations may have different supporting evidence (e.g., the mailing address of the individual, the content of the notice, the time allotted for objections, etc.). While the court bears the burden of providing these assurances, the separate burden of maintaining this assurance for a period of six years lies with the covered entity 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.

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% were assigned to the object property in this study. Table 5 presents six example constraints that illustrate from where data subjects are identified.

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.
164.522(a)(1)(iii)	Subject	Who has an agreement with an individual to restrict disclosures of PHI.
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.
164.524(c)(2)(ii)(B)	Target	Who agrees to the fees imposed for the summary of the PHI.

TABLE 5: DATA SUBJECTS

Paragraph	**Property**	**Value**
164.512(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.
164.506(a)(2)(ii)	Object	From an individual who is an inmate.
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.
164.502(j)(2)(i)	Object	Which is about the suspected perpetrator of the criminal act.

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, military personnel are discharged), systems must respond by updating the respective assigned stakeholder classifications accordingly.

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 [APS02, AHK02, BBL05]. In traditional Role-Based Access Control (RBAC) systems [SCF96], stakeholders are permitted or denied access to information based on the job functions 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) are equivalent to 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 on the subject, object or target properties. The purposes expressed in 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).

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.

All 389 constraints that describe valid purposes appear in non-parameterized pattern constraints. Among these, a total 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 purpose. 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, the 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) *Summary of Constraint Catalogue*

Table 7 provides a summary of the constraint catalogue and illustrates how constraints share multiple classifications.

TABLE 7: CATALOGUE OF NON-PARAMETERIZED CONSTRAINTS

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

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 non-parameterized, they can essentially contain "constraints upon constraints" that exhibit characteristics from multiple categories. While not parameterizing these constraints definitely saves time and effort for an engineer, non-parameterized constraints will inherently be susceptible to this multi-categorical ambiguity.

B. Handling Exceptions and Priorities

An exception is a special constraint that excludes interpretations from a set of properties or rules in a regulation. *Properties-based* (e.g., subjects, objects, etc.) exceptions are handled by negation whereas *rule-based* exceptions are handled by priorities. Exceptions that are negated are addressed at the time 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 are described in 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). Figure 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 non-shaded rows are the excluded purposes. The English conjunction "not" is in bold to illustrate the negation.

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.

FIGURE 5: RECORD WITH NEGATED EXCEPTION

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 [SV01]. 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.

Figure 6 illustrates 12 of the 58 rule-based exceptions that we extracted from the HIPAA Privacy Rule. These 12 exceptions comprise 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" and *the priorities are not transitive*. 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.

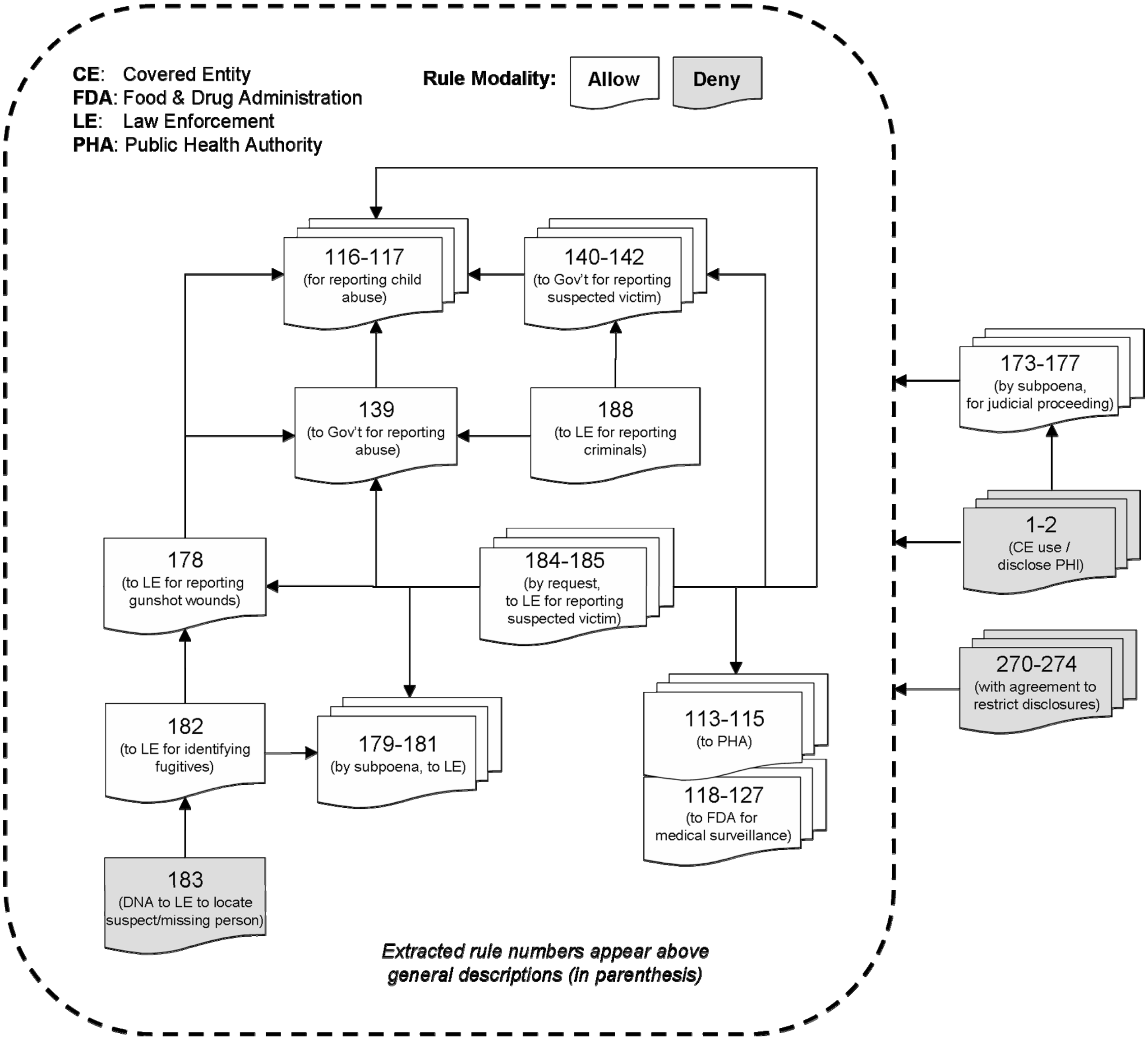
CE: Covered Entity
FDA: Food & Drug Administration
LE: Law Enforcement
PHA: Public Health Authority
Rule Modality:
Allow
Deny
116-117 (for reporting child abuse)
140-142 (to Gov't for reporting suspected victim)
139 (to Gov't for reporting abuse)
188 (to LE for reporting criminals)
173-177 (by subpoena, for judicial proceeding)
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)

FIGURE 6: EXAMPLE NETWORK OF PRIORITIES BETWEEN RULES

Priorities between allow rules affect the types of constraints that must be satisfied prior to a permitted disclosure as well as any follow-on obligations (e.g., post-conditions) 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 184-185 permit disclosures to LE for reporting suspected victims of crimes; the agreement of the individual is not required if he or 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 covered entity believes the disclosure will prevent further harm to the individual. Unlike rules 184-185, rules 139-142 incur the follow-on obligation to notify the individual of the disclosure if his or her agreement to the disclosure was not obtained. Presumably, this obligation to notify 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 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.

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*.

Reference phrases directly describe extracted rules or they are cross-references to other paragraphs in the regulation from which rules were extracted. In the latter case, these cross-references may be further 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 non-parameterized 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.

VI. Summary

Increasingly, regulations are requiring software engineers to specify, design and implement systems that are accountably in compliance with the law 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 a methodology to extract access rights and obligations from regulatory texts to ensure statement-level coverage for an entire regulation [BVA06]. 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 using grounded theory and has been validated using a substantial body of work.

While our extended methodology has been applied to a large U.S. regulation that governs information privacy in the healthcare domain, we believe the two extensions, which include (1) the method for acquiring data elements and (2) the method for prioritizing data requirements, can be used to analyze other regulations that govern information-intensive software systems. However, because the first extension requires that the engineer map non-parameterized constraints to one of six constraint types (e.g., subject, action, object, etc.), this extension will require additional work when all of the entities that appear in the non-parameterized 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 U.S. regulations that include exceptions to regulatory rules. These widespread cross-references provide compelling evidence to believe the second extension for prioritizing data requirements can be applied to other U.S. regulations governing information systems. Future work is needed to assess this methodology on regulations outside of the U.S. Finally, the constraint catalogue 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 are more likely to 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 non-functional in nature. This observation is supported, in part, for two reasons – these regulations: (1) are written to support marketplace diversity by intentionally offering broad interpretations that affect a variety of related, non-specific business practices; and (2) 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 government reach agreement on how to verify compliance with regulations, future work includes developing a method to identify criteria for evaluating functional requirements derived from non-functional regulations. In addition to comprising a set of “best practices,” these criteria should demonstrate to law and policy makers the efficacy of achieving a specific regulation under the restrictions of contemporary technology and available 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 law. The danger of not employing a systematic methodology is that it leaves organizations susceptible to security breaches. Furthermore, organizations who can systematically demonstrate how their software systems comply with policies and regulations can more effectively demonstrate due diligence and a standard of care.

VII. Acknowledgements

The authors wish to thank the members of ThePrivacyPlace.org for their helpful comments and feedback. This research was supported, in part, by two grants from the 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), as well as the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University and the IBM PhD Fellowship awarded by the IBM Center for Advanced Studies, Research Triangle Park, NC.

VIII. References

[Ant96] A.I. Antón, “Goal-based requirements analysis,” *2nd IEEE Int’l Conf. Requirements Engineering*, pp. 136-144, 1996.

[AEH04] A.I. Antón, J.B. Earp, Q. He, W. Stufflebeam, D. Bolchini, C. Jensen, “Financial privacy policies and the need for standardization,” *IEEE Sec. and Privacy*, 2(2):36-45, 2004.

[AE04]A.I. Antón, J.B. Earp, “A requirements taxonomy for reducing web site privacy vulnerabilities,” *Requirement Engineering*, 9(3):169-185, 2004.

[APS02] P. Ashley, C. Powers, M. Schunter, “From Privacy Promises to Privacy Management: A New Approach for Enforcing Privacy throughout the Enterprise,” *New Security Paradigms Workshop*, Virginia Beach, VA, pp. 43-50, 2002.

[AHK02] P. Ashley, S. Hada, G. Karjoth, M. Schunter, “E-P3P Privacy Policies and Privacy Authorization,” *ACM Workshop on Privacy Elec. Soc.*, pp. 103-109, 2002.

[BBL05] J-W. Byon, E. Bertino, N. Li, “Purpose-based Access Control of Complex Data for Privacy Protection,” *10th ACM Symp. Access Control Models and Technologies*, Stockholm, Sweden, pp. 102-110, 2005.

[BA05a] T.D. Breaux, A.I. Antón, “Deriving semantic models from privacy policies,” *6th IEEE Int’l Workshop on Policies for Dist. Sys. and Net.*, pp. 67-76, 2005.

[BA05b] T.D. Breaux, A.I. Antón, “Analyzing goal semantics for rights, permissions and obligations,” *13th IEEE Int’l Conf. Reqs. Engr.*, pp. 177-186, 2005.

[BA05c] T.D. Breaux, A.I. Antón, “Mining rule semantics to understand legislative compliance,” *ACM Workshop on Privacy Elec. Soc.*, pp. 51-54, 2005.

[BAK06] T.D. Breaux, A.I. Antón, C-M. Karat, J. Karat, “Enforceability vs. Accountability in Electronic Policies,” *7th IEEE Int’l Workshop on Policies for Dist. Sys. and Net.*, pp. 227-330, 2006.

[BAD06] T.D. Breaux, A.I. Antón, “Semantic Parameterization: A Conceptual Modeling Process for Domain Descriptions,” In Sumission: ACM Trans. Soft. Engr. Methods, North

Carolina State University Computer Science Technical Report TR-2006-35, Raleigh, NC, Oct. 2006.

[BAS06] T.D. Breaux, A.I. Antón, E.H. Spafford, "A Distributed Requirements Management Framework for Compliance and Accountability," North Carolina State University Computer Science Technical Report TR-2006-14, Raleigh, NC, Jul. 2006.

[BVA06] T.D. Breaux, M.W. Vail, A.I. Antón, "Towards compliance: extracting rights and obligations to align requirements with regulations," *14th IEEE Int'l Conf. on Reqs. Engr.*, Minneapolis, MN, pp. 49-58, 2006.

[BA07] T.D. Breaux, A.I. Antón, "Impalpable Constraints: Framing Requirements for Formal Methods," North Carolina State University Computer Science Technical Report TR-2007-6, Raleigh, NC, Feb. 2007.

[BKK05] C. Brodie, C-M. Karat, J. Karat, J. Feng, "Usable Security and Privacy: A Case Study of Developing Privacy Management Tools," *2005 Symp. Usable Privacy and Security*, Pittsburgh, PA, pp. 35-43, 2005.

[BLS06] Bureau of Labor Statistics, U.S. Dept. of Labor, *Career Guide to Industries, 2006-07 Edition*, Health Care.

[EPIC04] C.J. Hoofnagle, D.J. Solove, "Re: Request for Investigation into Data Broker Products for Compliance with the FCRA," Electronic Privacy Information Center, Washington, D.C., 2004.

[Far06] C.B. Farrell, "ChoicePoint Settles Data Security Breach Charges; to Pay $10 Million in Civil Penalties and $5 Million for Customer Redress," FTC File No. 052-3069, Office of Public Affairs, U.S. Federal Trade Commission, 2006.

[FTC06] United States v. ChoicePoint, Inc., Case No. 1:06-CV-00198-JTC, N.D. Ga., Feb. 15, 2006.

[Gar04] B.A. Garner (ed.), *Black's Law Dictionary*, 8th ed., Thompson West, St. Paul, MN, 2004.

[GS67] B.C. Glaser, A.L. Strauss, *The Discovery of Grounded Theory*, Aldine Publishing Co., 1967.

[GMM05] P. Giorgini, F. Massacci, J. Mylopoulos, N. Zannone, "Modeling Security Requirements through Ownership, Permission and Delegation," *13th IEEE Int'l Conf. Req'ts Engr.*, Paris, France, pp. 167-176, 2005.

[Jac95] M. Jackson, "The World and the Machine," *17th IEEE Int'l Conf. Soft. Engr.*, Seattle, WA, pp. 283-292, 1995.

[JZ93] M. Jackson, P. Zave, "Domain Descriptions," *IEEE Symp. Req'ts Engr.*, San Diego, CA, pp. 56-64, 1993.

[HER] *HIPAA Administrative Simplification: Enforcement*, U.S. Dept. Health and Human Services, 45 CFR Part 160 and 164. Federal Register 71(32), February 16, 2006, pp. 8389-8433.

[HLM04] C.B. Haley, R.C. Laney, J.D. Moffett, B. Nuseibeh, "The Effect of Trust Assumptions on the Elaboration of Security Requirements," 12th IEEE Int'l Conf. Req'ts Engr., pp. 102-111, 2004.

[HLN04] C.B. Haley, R. Laney, B. Nuseibeh, "Deriving Security Requirements from Crosscutting Threat Descriptions," *3rd Int'l Conf. Aspect-oriented Soft.* Dev., Lancaster, UK, pp. 112-121, 2004.

[HML05] C.B. Haley, J.D. Moffett, R. Laney, B. Nuseibeh, "Arguing Security: Validating Security Requirements using Structured Argumentation," 3rd Symp. Req'ts Engr. for Info. Sec., Paris, France, 2005.

[Hor01] John F. Horty. *Agency and Deontic Logic*, Oxford University Press, 2001

[HPR] *Standards for Privacy of Individually Identifiable Health Information*, U.S. Dept. Health and Human Services, 45 CFR, Part 164, Subpart E. Federal Register, 68(34), February 20, 2003, pp. 8334-8381.

[HSR] *Standards for the Protection of Electronic Protected Health Information*, U.S. Dept. Health and Human Services, 45 CFR Part 164, Subpart C. Federal Register, 68(34), February 20, 2003, pp. 8334-8381.

[Lam04] A. van Lamsweerde, "Elaborating Security Requirements by Construction of Intentional Anti-Models," 26th IEEE Int'l Conf. Soft. Engr., Scotland, UK, pp. 148-157, 2004.

[LGM06] S-W. Lee, R. Gandhi, D. Muthurajan, D. Yavagal, G-J. Ahn, "Building Problem Domain Ontology from Security Requirements in Regulatory Documents," Int'l Workshop on Soft. Engr. for Secure Systems, Shanghai, China, pp. 43-50, 2006.

[LNI03] L. Lin, B. Nuseibeh, D. Ince, M. Jackson, J. Moffett, "Introducing Abuse Frames for Analysing Security Requirements," 11th IEEE Int'l Conf. Req'ts Engr., pp. 371-372, 2003.

[MGL06] M.J. May, C.A. Gunter, I. Lee, "Privacy APIs: Access Control Techniques to Analyze and Verify Legal Privacy Policies," *19th IEEE Workshop Computer Security Found*ations., pp. 85-97, 2006.

[MCY99] J. Mylopoulos, L. Chung, E. Yu, "From Object-oriented to Goal-oriented Requirements Analysis," *Comm. ACM*, 42(1), pp. 31-37, 1999.

[RC99] J. Reagle, L.F. Cranor, "The Platform for Privacy Preferences," *Comm. ACM*, 42(2), pp. 48-55, 1999.

[SV01] P. Samarati, S. de Capitani di Vimercati "Access Control: Policies, Models and Mechanisms," *Foundations of Security Analysis and Design*, vol. 2171, pp. 137-193, 2001.

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

[VPW05] T. Verhannenman, F. Piessens, B. de Win, W. Joosen, "Requirements Traceability to Support Evolution of Access Control," *2005 Workshop Soft. Engr. for Secure Sys*., St. Louis, MO, pp. 1-7, 2005.

[XGN06] D. Xu, V. Goel, K. Nygard, "An Aspect-oriented Approach to Security Requriements Analysis," 30th Int'l Conf. on Computer Software and Applications, pp. 79-82, 2006.

[ZJ97] P. Zave, M. Jackson, "The Four Dark Corner's of Requirements Engineering," *ACM Trans. Soft. Engr. Methods*, 6(1), pp. 1-30, 1997.

[10823] *HIPAA Medical Privacy and Transition Rules: Overkill or Overdue?* Hearing before the Special Committee on Aging, U.S. Senate, 108th Congress, Sept. 23, 2003, Ser. 108-23.

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