Mining Rule Semantics to Understand Legislative Compliance

Travis D. Breaux and Annie I. Antón
Department of Computer Science
North Carolina State University
{tdbreaux, aianton}@eos.ncsu.edu

Abstract

Privacy legislation in the United States is distributed throughout separate documents that empower different federal authorities to regulate industry. Federal authorities in turn develop corresponding regulations intended to ensure that organizations satisfy legislative objectives. Organizations in regulated industries (e.g. healthcare and financial institutions) face significant challenges when developing policies and systems that are properly aligned with relevant privacy regulations. We analyze privacy regulations derived from the Health Insurance Portability and Accountability Act (HIPAA) that affect information sharing practices and consumer privacy in healthcare systems. Our analysis shows specific natural language semantics that formally characterize rights, obligations, and the meaningful relationships between them required to build value into systems. Furthermore, we evaluate semantics for rules and constraints necessary to develop machine-enforceable policies that bridge between laws, policies, practices, and system requirements. We believe the results of our analysis will benefit legislators, regulators and policy and system developers by focusing their attention on natural language policy semantics that are implementable in software systems.

1. INTRODUCTION

Legislation in the United States, including the Health Insurance Portability and Accountability Act (HIPAA)[1] and the Gramm-Leach-Bliley Act (GLBA)[2], establishes high-level objectives to protect both commerce and consumer privacy. These objectives describe administrative responsibilities for federal regulators who respond by issuing standards, recommendations and rules intended to implement legislation in a regulatory framework. *Standards* establish minimum or baseline performance expectations for the activities of governed parties. *Recommendations*, or guidelines, encourage governed parties to better meet legislative objectives while providing freedom to accommodate unforeseen circumstances. Finally, *rules*, which may be either standards or guidelines, express how to satisfy objectives by associating context-sensitive information with pre-conditions, effects, or constraints.

Companies seeking to achieve legal compliance must (a) ensure that their company policies comply with legal regulations (e.g., standards and recommendations) and (b) guarantee that their business processes and operational systems implement their policies. Non-technical stakeholders (e.g. corporate or executive officers, policy analysts and lawyers) interpret these legal regulations in the context of their organizations and develop their organizational policies accordingly. In response, technical stakeholders (e.g. information technology (IT) managers and system administrators) interpret organizational policies to configure and deploy software systems that support the organization's overall business processes.

Ensuring that regulations and organizational policies are properly aligned to satisfy federal law is a significant challenge. Baumer et al. have observed that healthcare professional perceptions of the misalignment between law, policies and practices with regards to HIPAA are problematic [BEP00]. They found that healthcare professionals are concerned that system-based protections that restrict access to

[1] Health Insurance Portability and Accountability Act of 1996, 42 U.S.C.§1320.
[2] Gramm-Leach-Bliley Act of 1999, 15 U.S.C. §§ 6801- 6809.

patient medical records for both medical and non-medical purposes are not satisfactory. Mercuri further highlights the corresponding technical challenge proposed by HIPAA [Mer04]; she identifies organizations that are falling back on traditional general security frameworks such as the ISO Common Criteria [ISOCC] given the lack of new frameworks targeted at HIPAA compliance standards.

Deploying operational software systems that comply with legislation requires technical stakeholders to understand regulations in such a way that compliance can be guaranteed in the systems they implement. To this end, we are investigating and modeling regulatory semantics to support the development of a policy language that can codify law, policies, and system requirements that are properly aligned. Our prior work in analyzing Internet health care [AE04, AEV05] and financial [AEB04] privacy policies provides a foundation for this work. These studies yielded over 1,200 unique, semi-structured goal statements that were extracted from over 100 Internet privacy policies using a technique called goal mining (the extraction of goal statements from texts during content analysis) [AER02, AE04]. We then specified formal semantic models that distinguish goals as either rights or obligations; we refer to this specification process as *semantic parameterization* [BA05a, BA05b]. *Rights* are activities that people or systems are permitted to do while *obligations* are activities that people and systems must do. Semantic models have properties that are desirable for comparing and disambiguating policy statements, regenerating natural language policy statements, and answering specific *what?, where?, how?* and *why?* questions.

In this paper, we present our most recent study in which we conducted an in-depth analysis of the Health Insurance Portability and Accountability Act (HIPAA). Employing our experience in applying semantic parameterization to goal statements, in this paper we apply semantic parameterization to the HIPAA Fact Sheet [HFS], to develop formal rule semantics that can bridge the gap between natural language privacy policies and technical system policies. Furthermore, we validate our observations by cross-referencing our results with the HIPAA Privacy Rule [HPR] from which the Fact Sheet was originally derived. Our analysis reveals that certain keywords in regulatory text are indicative of compliance rules and constraints for both people and systems. In addition, we discuss the relevance of these semantics to existing privacy policy languages and note the importance of balancing specific rights with obligations to ensure that rights and obligations both have value. We believe the results our analysis can help legislators, regulators and policy and software developers focus their attention on developing language with machine-enforceable semantics.

The remainder of this paper is organized as follows. In section 2 is an overview of the relevant related work and background to our previous research experience with analyzing privacy policies. Section 3 describes the organization of our case study. In section 4, we present the natural language patterns identified in our analysis that correspond to rights, obligations and constraints. In section 5, we generalize our observations to describe formal semantics for rules and constraints. Finally, in section 6 we discuss the relevance of our work to the privacy community with our summary and future work in section 7.

2. RELATED WORK AND BACKGROUND

2.1 Related Work

Several strategies have been proposed to derive formal models from the full scope of natural language (English), including conceptual dependencies [SH72] and conceptual graphs or semantic networks [SO84]. However, processing the full scope of natural language is excessive for analyzing privacy legislation. In privacy legislation, interesting natural language statements are limited to what tasks people and systems are entitled (rights and permissions) or obligated (responsibilities or requirements) to perform in order to satisfy legislative objectives. In addition to rules governing business processes, these statements include functional and non-functional system requirements. Within the limited scope of analyzing legislation, approaches include deriving first-order logic models [SSK86, She87, ABM98, KL03] and conceptual models using the Unified Modeling Language (UML) [EGB01].

First-order logic as a modeling notation provides sound and complete proofs of domain-specific properties. Generally, each variant of first-order logic provides certain benefits and limitations. In Section 5.2, we show that arithmetic operations are required to evaluate constraints from policy statements, yet these expressions are not decidable in first-order logic. However, it is still worth considering the strengths and weaknesses of logic-based models, since they uniquely describe the representational challenges to-date. Sergot et al. use deontic logic to model the British Nationality Act (BNA) of the United Kingdom [SSK86]. Deontic logic provides semantics for describing rights and obligations. They found that transcribing correct uses of negation from the BNA to logic were not straightforward and that counterfactual conditions within a single rule are prone to subjective interpretation. Sherman modeled the Canadian Income Tax Act in Prolog [She87] in which he noted difficulty representing time and events in a model based on first-order logic. Sherman's model was also limited to absolute temporal relations between an event and a specific date and time. We show the additional need to specify time periods and relative temporal relations between events; both are specifications independent of calendar time. Alternatively, Antonious et al. explore the use of defeasible logic in analyzing and reasoning about regulations [ABM98]. Defeasible logic allows prioritizing rules so that the highest priority rule fires unaffected by lower priority rules and they highlight its use in resolving logical inconsistencies. Finally, Kerrigan and Law describe the REGNET system developed for regulatory compliance assistance and tested in the domain of environmental law [KL03]. REGNET supports the user by managing cross-references, regulation subtexts and XML associations between regulation subtexts and simple logic rules. The system provides compliance assistance by checking the consistency of logic rules across subtexts.

An alternative to logic-based approaches includes the work by van Engers et al. called POWER that uses the UML and Object Constraint Language (OCL) to model legislation for the Dutch Tax and Customs Administration [EGB01]. The UML provides general semantics for visual classification and aggregation while OCL can express logical conjunctions, disjunctions and arithmetic operations. Unlike our approach which seeks to produce a machine-independent but enforceable policy language, their work is oriented towards code generation. Furthermore, our work is motivated by grounded-theory in which our language semantics are developed by analyzing the semantics of domain-relevant legislation and regulations [GS67].

Apart from the analysis of legal texts, formal languages are emerging to represent privacy policies and rules including the Platform for Privacy Preferences (P3P) [P3P] with the P3P Preference Exchange Language (APPEL) [APPEL] and the Enterprise Privacy Authorization Language (EPAL) [EPAL]. Ragawal et al. provide a thorough evaluation of APPEL in which they expose a design flaw making the expression of privacy preferences error-prone and ambiguous for the user [AKS03]. Stufflebeam et al. provide a comparative evaluation of P3P and EPAL whereby they instantiate these policy languages using real-world privacy policies [SAH04]. They discover semantic limitations in both languages wherein the natural language policies prescribe temporal constraints on events that the policy languages were not designed to express. Recently, the Oasis Standard XACML 2.0 [XACML] was adopted that includes a Privacy Policy Profile which assigns purposes to data types and data access. Limitations of XACML in the domain of privacy policies are discussed in section 6. For these reasons, we believe the semantics for a machine-enforceable policy language must be carefully developed by analyzing relevant legislation and existing privacy policies.

2.2. Background

Because the full scope of natural language can be overly complicated, we focus our attention on the subset of natural language necessary to encode rights, permissions, and obligations. Semantic parameterization transforms restricted natural language statements (RNLS) into unique and comparable semantic models. The semantic models are formally represented by a context-free grammar called the Knowledge Transformation Language (KTL) that has been used to compare and query policy statements and generate natural language policy statements from semantic models [BA05b]. An Eclipse plug-in supports composing KTL expressions, querying such expressions to answer open-ended questions, and generating natural language. Semantic parameterization has been validated using a database of over 1,200 policy goals [BA05a]. In that evaluation, we identified a need to formally represent numerical ranges, cardinal numbers and temporal relationships which we now address in this work. To date, the breadth of the natural language subset supported by KTL has been sufficient to describe actors and objects and their roles in a variety of activities, however, we still require an in-depth evaluation of rules and constraints necessary to specify system requirements and machine-enforceable policies.

Restricted natural language statements (RNLS) are useful to derive semantic models since they reduce the complexity commonly found in unrestricted natural language. RNLS(s) reduce complexity by describing exactly one activity but allowing for external references to other RNLS(s). In the unrestricted statement UNLS #1, the main activity "providers may charge patients" is re-stated in RNLS #1 but with the nested activities "providers copy records" and "providers send records" externalized into RNLS #2 and #3. When formulating RNLS(s), the implicit actors must be made explicit as seen in RNLS #2 and #3. Additional unstated information often becomes obvious and stating such information externalized RNLS(s) will inevitably clarify and disambiguate the meaning of the original statement.

UNLS #1: `Health care providers may charge patients for the cost of copying and sending their records.`

RNLS #1: `Health care providers may charge patients for the cost of` *`(RNLS #2)`* `and` *`(RNLS #3)`*`.`
RNLS #2: `Health care providers copy patient records.`
RNLS #3: `Health care providers send patient records.`

Semantic models represent information using a unary root relation and two binary, asymmetric relations: the associative and declarative relations. The root relation identifies the main idea in an expression. The associative relation is used to build conceptual relations between two concepts while the declarative relation is used to assign concepts to these conceptual relations. The operands for all relations are restricted to a single part of speech. For example, activities found in RNLS(s) are represented using an activity model that defines the following associated relations: α(*activity*, *actor*), α(*activity*, *action*), α(*activity*, *object*). Using this model, we can decompose RNLS #2 for example into the following declarative relations: δ(*actor*, *health-care-provider*), δ(*action*, *copy*), δ(*object*, *patient-record*). Finally, the root relation is used to distinguish the activity as the main idea σ(*activity*). The activity model with a *actor*, *action* and *object* is a simple example with more sophisticated examples found in earlier work [BA05a, BA05b].

The Knowledge Transformation Language (KTL) provides semantics for combining the set of formal relations that describe a semantic model into a single expression. The operators in KTL were designed to maintain a correspondence with simple natural language sentences. For example, the associative relations $\alpha(C, P_i)$ for an arbitrary concept C and parameters P_i for $1 \leq i \leq n$ is expressed in two ways: 1) using set-associative operators (curly brackets) for a set of parameters such as C { P_1 P_2 … P_n } which reads "C has P_i" or 2) for a single parameter using the reverse-associative operator (a colon) such as P_i : C which reads "P_i of C". The declarative relation $\delta(P_i, C_j)$ for an arbitrary concept C_j for $1 \leq j \leq m$ and parameters P_i is also expressed in two ways: 1) using the declarative operator (equals) for a logical set of concepts such as $P_i = C_1$ & (C_2 | … C_m) which reads "C_j is P_i" or 2) using the reverse-declarative operators (square brackets) for a logical set of parameters such as C_j [P_1 & (P_2 | … P_n)] which reads "C_j that is P_i". Valid semantic models always maintain these correspondences with simple natural language. For the complete context-free grammar describing full KTL see Appendix A. As a more sophisticated example, the KTL expression describing the sum of relations for RNLS #1, #2 and #3 are provided below:

```
activity [ right : health-care-provider ] {
     actor = health-care-provider
     action = charge
     object = patient
     target = (cost : activity {
          actor = health-care-provider
          action = copy
```

```
                  object = patient-record
            } & cost : activity {
                  actor = health-care-provider
                  action = send
                  object = patient-record
            })
      }
```

Semantic models support queries that ask *what*, *where*, *how* and *why* questions. Queries are either Boolean questions in that they ask very specific questions yielding a *yes* or *no* response or they are *wh*-questions which are open-ended queries yielding values from semantic models. The query algorithm matches values in a semantic model against query variables that are normal identifiers prefixed by a question mark. For example, the query that asks the question “Who may charge patients and for what costs?” is represented in KTL as follows:

```
      activity [ right : ?whom ] {
            action = charge
            object = patient
            target = cost : ?what
      }
```

Queries provide the foundation for building more advanced applications. The template method for generating natural language policy statements is one such application [BA05b]. As we discuss in Section 5, we foresee queries playing a key role in constraints and the pre-conditions of rules in a machine-enforceable policy language. Finally, throughout our analysis, the incremental and structured representations provided by RNLS(s) and semantic models made it easier for us to recognize the necessary semantics to formalize rules and constraints from natural language regulatory text.

3. ANALYZING PRIVACY LEGISLATION

In the United States, the “law” is fairly complex since it is distributed across many documents that can both supersede and extend each other. In addition, the law is constantly evolving through court decisions based on real-world events. The legislative lifecycle begins with a congressional document such as the Health Insurance Portability and Accountability Act passed by the U.S. Congress in April 2000. Federal regulators, directed by such legislation as HIPAA, then develop regulations for governing the behavior of individuals and organizations. In the case of HIPAA, two important regulations were adopted by the U.S. Department of Health and Human Services (HHS) including the *HIPAA Security Rule* [HSR] and the *HIPAA Privacy Rule* [HPR]. The Security Rule specifies a number of physical and electronic safeguards that in general will increase the security of protected health information systems. In contrast, the Privacy Rule describes context-sensitive rights and obligations affecting information sharing practices that are required to protect individually identifiable health information. Finally, to assist with public consumption of the law, regulators (e.g., HHS) also provide various Fact Sheets targeted at specific interest groups (e.g., patients, hospitals, insurers) that attempt to address the most pertinent questions from each group’s unique perspective.

For this study, we analyzed the following HIPAA-related documents:

- Fact Sheet: Protecting the Privacy of Patients’ Health Information” [HHS04]
- HIPAA Privacy Rule: Section 160, Subparts C, E and Section 162, Subparts E [HPR]

In order to address the diversity of documents and their relative importance, we designed our case study to be formative with the purpose of identifying additional semantics for representing formal policy rules. The Fact Sheet was prepared by the HHS to define rights and obligations established in the HIPAA Privacy Rule. The Fact Sheet is more amenable to analysis than the rule because it results from an effort that includes “answers to hundreds of common questions about the rule as well as explanations and descriptions about key elements of the rule” [HHS04]. The Fact Sheet was written to comply with law while highlighting information most relevant to consumers and patients in a reader-friendly document. An important difference is the Fact Sheet excludes a complex matrix of cross-references distributed through the original Privacy Rule. Our analysis of the Privacy Rule shows a total of 439 cross-references across 22 sections of the rule with a maximum 71, mean 19.9, and median 17 cross-references. In addition, 38 cross-references referred to non-HIPAA documents. Each cross-reference qualifies the meaning of the containing statement by referring to a definition or statement in another section or more rarely in another document. Finally, the HIPAA Privacy Rule has eighteen times more words than the Fact Sheet, making the fact sheet a reasonable introduction for a formative study.

The analysis procedure that we applied to the Fact Sheet is described in three steps which were repeated throughout that document: 1) identify a natural language statement that expresses rights, permissions, or obligations; next apply semantic parameterization to the statement to 2) derive semantic models for the actors, actions, and objects of each statement and 3) derive rules with pre-conditions and effects built from temporal constraints that interrelate individual semantic models. The two applications of semantic parameterization both produce reusable natural language patterns that make the process more consistent and hence repeatable.

Applying the semantic parameterization process to the entire Fact Sheet yielded encodings for 15 rights, 19 obligations and 12 rules. In addition, several reusable patterns were identified including seven patterns for rights, seven for obligations and nine for rules. These patterns are presented in Section 4. The process required 11 person hours; the first author spent only 4 hours initially with an additional 7 hours spent by both the first and second authors collaborating. Finally, we indexed the original Privacy Rule using the twenty-three patterns to validate our observations of these patterns in the original regulation text.

4. PATTERNS FOR RIGHTS, OBLIGATIONS AND RULES

The natural language patterns correlate unique word sequences with specific parts of speech (e.g., modals, verbs, prepositions) to rights, obligations and rules. The patterns for encoding rights and obligations are similar because they all identify a primary actor, action, and some relationship to other objects or activities. The patterns for encoding rules more frequently coincide with the patterns for encoding obligations than those for encoding rights. In the following examples, an actor is either a provider of health-related services or products or a consumer or patient.

4.1. Patterns for Rights *(R)*

Rights define what an actor is allowed to do in terms of their capabilities. For example, an actor, such as a patient, may be capable of “seeing” their medical records however they may not have the expressed right to perform this activity. The following seven natural language patterns were identified that consistently encode rights:

R_1: <actor> should/may be able to <verb> …
R_2: <actor> may <verb> …
R_3: <actor> can/could <verb> …
R_4: <policy> permits <actor> to <verb> …
R_5: <actor> would not have to <verb> …
R_6: <policy> does not restrict… <actor> …
R_7: <policy> does not require <actor> …

Among the seven patterns, three cases are highlighted. The most general case includes patterns R_1, R_2 and R_3 where a right is expressly stated for a particular actor or group of actors. In this case, the modalities “should,” “may,” and “can” suggest that the actor has both the capability and the right to perform the action (a verb). In the Fact Sheet, the first case represented by pattern R_1 appears in the following statement S_1:

> *S_1:* Patients generally **should be able to** see and obtain copies of their medical records…

In *S_1*, the "patient" is the actor and their rights include the actions (both verbs) to "see" and "obtain" copies of their medical records. The patterns for *R_2* and *R_3* work similarly and in all cases the rule is the implied authority granting these rights to actors.

In most circumstances, the policy is the *implied* authority transferring rights to actors and is not expressively stated in the natural language statements. The pattern *R_4* is different, however, since it explicitly identifies the policy, or in this case the rule, as the authority transferring rights to actors. In the following example *S_2*, the pattern *R_4* distinguishes this statement as a right of the provider:

> *S_2:* …the rule **permits** doctors and other covered entities to communicate freely with patients about treatment options…

The last three cases demonstrate how the language used for obligations (see Section 3.2), such as "would have to," "restrict," or "require," is negated to establish a right. In other words, if an actor is not obligated to perform some action then, unless otherwise stated, they have the *implied* right to perform or not perform the action at their discretion. In the following example *S_3*, the pattern *R_6* identifies this statement as a right of the provider:

> *S_3:* …the rule **does not restrict** the ability of doctors, nurses and other providers to share information needed to treat their patients…

As we will see in Section 3.2, negating specific keywords for rights also establishes symmetric obligations for the associated activities.

Each of these patterns was indexed in the Privacy Rule text to validate the consistent usage of these patterns to specify rights in the regulation text. In Table 1, the number of times a pattern was conferring a right to an actor is documented with correct and incorrect instances in the text. The patterns including the modal "may" were generally consistent despite the fact that "may" can be used to mean a general possibility as opposed to a specific right of an actor.

Pattern	***Rights***	***Other***
<actor> should/may be able to…	0	0
<actor> may…	190	17
<actor> can/could…	0	9
<policy> permits…	3	1
<actor> would not have to…	0	0
<policy> does not restrict…	0	0
<policy> does not require…	0	0

Table 1: Patterns for Rights in the Privacy Rule

4.2. Patterns for Obligations *(O)*

Obligations define the required behavior of an actor in one or more activities. The following seven patterns were identified in the Fact Sheet that consistently encode obligations.

```
O1: <actor> should <verb> …
O2: <actor> should be <verb'ed> …
O3: <actor> will/would <verb> …
```

```
O4: <actor> must/must be <verb’ed> …
O5: <actor> which is charged with <verb’ing> …
O6: <policy> requires <actor> to <verb> …
O7: <actor> may not <verb> …
```

The patterns for obligations have several notable characteristics. First, pattern *O_1* and *O_2* are similar except that the verb in *O_2* is in the past-tense form and is preceded by the verb “be”. It is foreseeable that pattern *O_4* could have a similar relation with the modal “must” accompanied by a present-tense verb. The following statements *S_4* and *S_5* show the original context for *O_1* and *O_4*, respectively.

```
S4: …covered entities generally should provide access to these records
    within 30 days…
S5: …personal health information generally must be used only for
    purposes related to health care…
```

Similar to the patterns for rights, the patterns for obligations include pattern *O_6* that explicitly identifies the policy as the authority transferring obligations to actors. Statement *S_6* provides an example where pattern *O_6* occurs in the Fact Sheet. In addition, pattern *O_7* uses the language seen in rights with negation to establish an obligation for the actor. Statement *S_7* provides and example for pattern *O_7*.

```
S6: …the rule requires covered entities to have written privacy
    procedures…
S7: …personal health information generally may not be used for purposes
    not related to health care…
```

The patterns for obligations were also indexed in the Privacy Rule to validate semantic consistency. Table 2 shows each of the patterns and the number of times they were used for assigning an obligation to an actor in the rule.

Pattern	***Obligations***	***Other***
<actor> should…	0	1
<actor> will/would…	18	31
<actor> must/must be…	189	0
<actor> which is charged with…	3	1
<policy> requires…	1	0
<actor> may not…	30	0

Table 2: Patterns for Obligations in the Privacy Rule

4.3. Patterns for Constraints *(C)*

For our purposes, rules associate pre-conditions with effects. Both pre-conditions and effects contain constraints that may describe activities, such as “a patient makes a request to a provider,” or state such as “information is classified protected”. If a pre-condition is true then a set of corresponding effects are also true. In the results of our analysis, pre-conditions and effects often included temporal constraints between the times of activities and other activities or calendar times. Temporal constraints serve to create events from activities and/ or states by relating them to explicit times or sequences of events. In a rule, the pre-conditions will contain one or more conditions some of which describe activities, states and/ or have temporal constraints.

The following nine patterns *C_1* through *C_9* were extracted to identify rules in the Fact Sheet.

```
C1: <actor> should be able to <action>… if <actor/ object>… <verb>
```

C_2: <actor> may <verb>… but <actor> would not have to <verb>…
C_3: <actor> will <verb>… on/upon <event>…
C_4: <actor> may <verb>… for/for each <event>…
C_5: <actor> must <verb>… to ensure that <actor>… will <verb>…
C_6: <actor> would have to <verb>… before <verb>…
C_7: <actor> must first <verb>… before <verb>…
C_8: <actor> must <verb>… by <date>…
C_9: <actor> should <verb>… within <timeframe>…

There are two classes of rules distinguishable by how the rule semantics are associated with temporal constraints. The first class of rules have the event *A* in the pre-conditions all occurring before the event *B* in the effects as evidenced by patterns C_1 through C_7. In this case, the rule is equivalent to a temporal constraint between two activities. In patterns C_1 and C_2 the first activity is the effect for the second activity, the pre-condition. In patterns C_3 and C_4, the temporal constraints associated with the terms "on," "upon," "for," and "for each" establish the first activity as the effect of the latter activity, the pre-condition. In patterns C_5 through C_7, however, the first activity is the pre-condition for the latter activity, the effect. The rule semantics for patterns C_1 through C_7 are shown in Expression E_1. The less-than sign signifies a comparison between the finish time T_f of event *A* and the start time T_s of event B.

$$E_1: \mathit{if}\ \{\ A\ \}\ \mathit{then}\ \{\ T_f : A < T_s : B\ \}$$

The second class of rules shows that an activity that finishes at time *T* must occur before a deadline T_1 or within a timeframe (T_2, T_3). In pattern C_8, the activity must occur before the deadline while in pattern C_9 the activity should occur within the timeframe. Similar to the patterns C_1 through C_7, the rules corresponding to C_8 and C_9 also have an equivalent set of temporal constraints. For sometime T_0, the rule semantics for patterns C_8 and C_9 are shown in Expressions E_2 and E_3, respectively.

$$E_2: \mathit{if}\ \{\ T_0 > T_1\ \}\ \mathit{then}\ \{\ T : G < T_1\ \}$$
$$E_3: \mathit{if}\ \{\ T_2 < T_0\ \&\ T_0 < T_3\ \}\ \mathit{then}\ \{\ T_2 < T : G < T_3\ \}$$

In the above two cases, the activities were described in natural language without explicit reference to the start or finish time of these activities. Because the formalism of our approach affords (and requires) a higher degree of formality and specification, these statements would require further elaboration by the regulator. In other words, transcribing natural language policy statements into KTL exposes ambiguities in the original language and causes the elicitation of specific details necessary to remove such ambiguity. Finally, in both cases the rules and temporal constraints define behavior that is expected to occur, and if such rules were violated then the regulations from which they were derived would have been violated as well.

5. ANALYSIS OF CONSTRAINTS

Applying Semantic Parameterization to the Fact Sheet yielded insights into natural language dynamics in policy statements that are required to specify rules with constraints in both pre-conditions and effects. We seek to generalize our observations and in particular we identify the need to represent cardinality, arithmetic operators, comparison relations, and ordinality as observed first in the Fact Sheet and later in the HIPAA Privacy Rule.

5.1 Cardinality

Numbers in policy statements can be divided into two categories: symbolic and cardinal numbers. Symbolic numbers such as zip codes or social security numbers are strictly representational in nature and may be treated as unique identifiers for a concept such as region or person, respectively. Cardinal numbers, however, signify a quantity of some concept. For example, the HIPAA Fact Sheet states several *penalties* (sanctions) for not complying with an obligation including fines from 100 dollars to 100,000 dollars and time in prison less than 10 years. In these cases, the concept is a penalty such as a *fine* in a number of *dollars* or a *prison sentence* in a number of *years*. In general, cardinal numbers always pair a *numerical quantity*, such as 100,000, with a conceptual unit, such as dollars, and typically imply some *named quantity* such as a fine. Cardinal numbers serve as the operands to arithmetic operators and comparison relations and may be used to establish an ordinal relation across a set of entities. In the Privacy Rule, we identified 64 different instances of cardinal numbers.

5.2 Arithmetic Operators, Comparison Relations

Natural language policy statements include adjectives (in inflected form) and prepositions that, used in conjunction with cardinal numbers or named quantities, indicate an *increase* or *decrease* (arithmetic) operation or a *greater than* or *less than* (comparison) relation. While a few keywords are generic such as *more* and *less*, most are relevant only to a specific named quantity. For example, the keywords *before*, *during*, and *after* are relevant to the named quantity time; *younger* and *older* are relevant to age; *shorter*, *longer*, *wider*, and *taller* are relevant to width, height, length, etc. In general, if the keyword is preceded by a numerical quantity and followed by a named entity with an implied reference to an appropriate named quantity in natural language, the keyword refers to an arithmetic operation. For example, a deadline described by the statement "30 days after the request" signals an arithmetic operation where "30 days" is added to "the time of request" to establish the deadline.

Alternatively, the keywords may appear in a comparison relation between two entities. For example, in the Fact Sheet the statement "patients would have to sign a specific authorization *before* a covered entity could release their medical information" compares the time of two events establishing that one event occurs before another. Evidence for the use of these and similar keywords has been indexed in the HIPAA Privacy Rule and the results appear in Table 3 with totals for the number of instances that were arithmetic (***A***), comparative (***C***), and neither (***N***).

Keyword	***A***	***C***	***N***	***Example from HIPAA Privacy Rule***
less	5	1	0	• not ***less*** than 30 days before… • ***less*** that 6 years from…
more	27	10	0	• no ***more*** frequently than once every… • contains ***more*** than 20,000 people…
before	1	9	9	• at least 15 days ***before*** the… • not less than 30 days ***before***…
after	20	8	2	• 180 days ***after*** the effective date… • ***after*** the compliance date…
older	0	1	0	• age 90 or ***older***…
smaller	0	1	0	• geographic subdivisions ***smaller*** than a state…
longer	2	7	0	• no ***longer*** than 30 days from the date… • no ***longer*** needed for the purpose…
during	12	4	0	• ***during*** the first year after… • ***during*** normal business hours…
within	25	0	5	• ***within*** 180 days of when… • ***within*** the time limit set…

Table 3: Inflected-form adjectives used in arithmetic, comparative operations.

In KTL, an explicit arithmetic operator exists that will add or subtract two quantities of the same type of entity (see Expression E_4, where ***a*** and ***b*** are actual numbers). In addition, context-sensitive or implicit arithmetic operators can be defined that add or subtract quantities by allowing an implicit reference to a numerical quantity. In this case, a path from the root concept to a named quantity in a semantic model is used to identify the implicit numerical quantity. For example, for some number ***b***, the statement "***b*** minutes after an event" has the root concept *event* with a path to the *time* of the event (see Expression E_5). Since time is a continuous yet segmented quantity (e.g., segmented into seconds, minutes, etc.), it is handled as a separate type from other concepts. Finally, comparative relations are defined in the same fashion with both explicit (see Expression E_6) and implicit operators. Where the evaluation of an arithmetic operation is a numerical quantity, the evaluation of a comparative relation is a Boolean.

E_4: *[**a**] dollar* + *[**b**] dollar* ↔ *[**a*** + ***b**] dollar*
E_5: *event* + *[**b**] minute* ↔ *[**a**] minute* [*time* : *event*] + *[**b**] minute*
E_6: *[**a**] dollar* < *[**b**] dollar* ↔ true if and only if ***a*** < ***b***

5.3 Ordinality

Ordinality refers to the order of an entity within a set of comparable entities. Ordinality appears in HIPAA with adjectives such as *first*, *second*, and *last* based on contextual criteria relevant to ordering a set of entities. For example, the *first* event always refers to the earliest event in a set of events ordered by time. Ordinality depends on the existence of comparison relations to create an order, and as a result, the inflected-form adjectives each have an ordinal form that describes the *first* or *last* entity in an order. For example, the forms *least* and *most* are generic while others refer to specific concepts such as *earliest* and *latest* for time, *oldest* and *youngest* for age, *shortest*, *longest*, *widest* and *tallest* for width, height, length, etc. In the Privacy Rule, the ordinals *first* and *last* were most common with 7 and 3 occurrences, respectively. Typical usage for ordinals in the Rule include "the *first* disclosure during the accounting period" and the "individual's *last* known address." In Expression E_7, a partial-order is established over a set of elements defined by a query (all event models), a comparative operator (less-than) and a conceptual reference (the time of the event). In this order, the first event is also the earliest event in the set of events.

E_7: *order* { *event* } *by* { < *time* : *event* }

6. DISCUSSION

The analysis of the HIPAA Fact Sheet and Privacy Rule provided three important insights: 1) rules may be more effectively expressed using temporal constraints, 2) constraints expressed in regulation text include cardinality, arithmetic operators, comparison relations, and ordinality, and 3) rights must be balanced with corresponding obligations otherwise they have no value to stakeholders.

Rules in the form of pre-conditions with effects are desirable in formal models to enable logical inference. In general, such rules establish a *correlation* between evidence that is a weaker relationship than *causation*. In other words, a rule may establish that "if we observe evidence *A* then we also expect to observe evidence *B*." The actual *justification* for observing *B* in the context of *A*, however, is not expressed by a rule. In some circumstances, this may be sufficient: for example, if the relationship is unknown or the explicit definition of the relationship is unnecessary. However, as observed in our analysis of HIPAA these relationships are often stated and deemed relevant to the interpretation of the regulations. For example, in Section 4.3 we observed that each rule, by separating the relevant information into pre-conditions and effects, conditionally exposes the corresponding temporal constraint. In fact, more important than establishing a correlation is that the temporal constraint is usable to detect policy violations without additional inference per se; whereas the rule only describes an indirect property of the desired environment.

Our analysis also revealed that in HIPAA constraints are often described using cardinal numbers, arithmetic operations, comparison relations and ordinals to distinguish entities in regulations. Understanding the relationship between constraints and the original regulation text is important in order to evaluate the ability of emerging policy languages to sufficiently express compliance requirements. EPAL 1.1 [EPAL] and Oasis XACML 2.0 [XACML] express numbers as attributes, however, they do not support the designation of units for these numbers — a source of potential ambiguity if one policy describes a time in minutes and another policy describes time in hours, for example. With regards to arithmetic operators and comparison relations, P3P [P3P] relies on the W3C APPEL 1.0 Working Draft [APPEL] for rules that do not include either, although support for comparison relations are declared as items for future work. Alternatively, the EPAL 1.1 standard defers to XACML for conditions that allow both arithmetic operators and comparison relations. Finally, P3P, APPEL, EPAL and XACML all lack semantics to express ordinality over a set of related elements. Since regulations clearly indicate the need for this functionality, a complete and effective policy language will need to be equally expressive.

Lastly, we observed that rights and obligations are complementary and that they must be balanced to ensure rights are both accountable and enforceable. For example, in the HIPAA Privacy Rule the patient may request that the healthcare provider restrict access to their protected health information, however, the provider is not obligated to honor that request [HPR]. Rights without complementary obligations are meaningless since governed parties are not required to respond to the invocation of such rights. In terms of designing and engineering software systems, these rights may be effectively ignored. On the other hand, obligations without an explicit and complementary right do have value and must be properly incorporated into system specifications.

7. SUMMARY & PLANS FOR FUTURE WORK

The process of ensuring that software systems (and related business processes) comply with privacy law includes understanding the formal relationships between legislation, regulations, organizational policies and system specifications. In this paper, we analyze privacy regulations to identify formal semantics that govern both business processes and software systems. These semantics specify rights and obligations that define what people and systems are permitted to do and what they must do, respectively. Furthermore, we identified natural language constraints including cardinal relations, arithmetic operators, comparison relations and ordinal relations to characterize how rules describe measurable expectations of people and systems. From here, we foresee several open problems inviting future work including the development of methods to improve the machine-enforceable quality of policy texts in addition to methods that assist policy analysts and software developers ensure their systems comply with policy. Two problems in this area include the role of ambiguity in policy texts and the conceptual gap between policy texts and system specifications.

Legislation and regulation text is often considered too ambiguous for enforcement and compliance purposes. In some cases, the ambiguity is desirable because it is impossible to enumerate all intended interpretations of law when developing regulations. However, there exist different types of undesirable ambiguity that require unique approaches to disambiguate and clarify the intended meaning. For example, using a general word such as “covered entities” may be ambiguous in context if the only intention is to refer to “those covered entities that exclusively provide financial services.” Alternatively, using adjectives such as “to respond quickly” or “to respond within reasonable time” are subjective and therefore prone to ambiguous interpretations. In the future, we seek to build upon our observations from this work to enable categorizing types of ambiguity and developing methods to automatically detect and interactively clarify ambiguous language. One approach may be to detect and define non-functional qualities in terms of functional constraints over quantities, such as the functional constraints discussed in Section 5.

Requirements in software engineering are functionally equivalent to machine-enforceable obligations, in that they describe what systems must do to satisfy stakeholder needs. However, legislation and regulations rarely reach the detail required to specify complete system requirements. The HIPAA Security Rule is fairly unique in that it includes the requirement for Role-based Access Control (RBAC) in healthcare systems that store patients’ protected health information [HSR]. On the other hand, legislation and regulations do specify non-functional requirements that may be used to develop sound strategies or industry practices in the form of functional system requirements. For example, the HIPAA Privacy Rule states that covered entities must limit employee access to protected health information to only certain purposes. Compliance requires an approach (such as a methodology and/ or framework) that enables regulators, compliance officers and system administrators to trace from the obligation to a set of system requirements and evaluate the solution's ability to sufficiently offset the risks of associated vulnerabilities.

As previously mentioned, our research was conducted using grounded-theory [GS67]; that is, we developed our language semantics by analyzing the semantics of domain-relevant legislation and regulations. We believe the work presented in this paper can be extended towards addressing such challenges as bridging the gap from privacy regulations to system requirements. To date, we have shown that KTL catalogs the requirements for a privacy expression language; however, to further validate this work, we intend to compare KTL to other business rule languages that allow for expression of deontic logic modes.

REFERENCES

[ABM98] G. Antoniou, D. Billington and M. Maher. “On the Analysis of Regulations Using Defeasible Rules.” In *Proc. of the AAAI-98 Workshop on Knowledge Management and Business Process Reengineering*, Madison, Wisconsin, pp. 46-50, July 1998.

[AEB04] A.I. Antón, J.B. Earp, D. Bolchini, Q. He, C. Jensen and W. Stufflebeam, “The Lack of Clarity in Financial Privacy Policies and the Need for Standardization,” *IEEE Security & Privacy*, vol. 2 no. 2, pp. 36-45, 2004.

[AKS03] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu. “An XPath-based Preference Language for P3P” In *Proc. of the 12th Int'l. Conf. on World-Wide Web (WWW'03)*, Budapest, Hungary, pp. 629-639, May 2003.

[APPEL] L. Cranor, M. Langheinrich and M. Marchiori. A P3P Preference Exchange Language (APPEL), version 1.0. W3C Working Draft, http://www.w3.org/TR/P3P-preferences/

[BA05a] T.D. Breaux and A.I. Antón. “Deriving Semantic Models from Privacy Policies.” In *Proc. of the 6th Int’l Workshop on Distributed Systems and Networks (POLICY’05)*, Stockholm, Sweden, 6-8 June 2005, pp. 67 – 76.

[BA05b] T.D. Breaux and A.I. Antón. “Analyzing Goal Semantics for Rights, Obligations, and Permissions.” In *Proc. of the 13th Int’l Conf. on Requirements Eng. (RE’05)*, Paris, France, Aug. 29-Sept. 2, 2005.

[BCK03] S. Byers, L.F. Cranor and D. Kormann. “Automated Analysis of P3P-enabled Web Sites” In *Proc. ACM 5th Int'l. Conf. on Electronic Commerce*, Pittsburgh, Pennsylvania, pp. 326-338, October 2003.

[BEP00] D. Baumer, J.B. Earp, F.C. Payton. “Privacy of Medical Records: IT Implications of HIPAA.” *ACM Computers and Society*, 30(4), pp. 40-47, 2000.

[EGB01] T. van Engers, R. Gerrits, M. Boekenoogen, E. Glassée, P. Kordelarr. “POWER: Using UML/OCL for Modeling Legislation” In Pro*c. of the 8th Int'l Conf. on Artificial Intelligence and Law*, St. Louis, Missouri, pp. 157-167, May 2001.

[EPAL] P. Ashley, S. Hada, G. Karjoth, C. Powers and M. Schunter. Enterprise Privacy Authoring Language (EPAL), version 1.1, http://www.zurich.ibm.com/security/enterprise-privacy/epal/Specification/

[GS67] B.G. Glaser and A.L. Strauss. *The Discovery of Grounded Theory*. Aldine Publishing Company, Chicago, Illinois, 1967.

[HHS04] “Fact Sheet: Protecting the Privacy of Patients’ Health Information,” published by the U.S. Department of Health and Human Services, Washington D.C., April 14, 2003.

[HIPAA] “Health Insurance Portability and Accountability Act,” USC H.R. 3103-168, April 2000.

[HPR] “Standards for Privacy of Individually Identifiable Health Information.” 45 CFR Part 160, Part 164 Subpart E. In Federal Register, vol. 68, no. 34, February 20, 2003, pp. 8334 – 8381.

[HSR] "Standards for the Protection of Electronic Protected Health Information" 45 CFR Part 164, Subpart C. In Federal Register, 68(34), February 20, 2003, pp. 8334 – 8381.

[ISOCC] ISO/IEC 15408:1999, "Evaluation Criteria for Information Technology Security", International Organization for Standards (ISO), 1999.

[JS92] A.J.I. Jones and M. Sergot. "Deontic Logic in the Representation of Law: Towards a Methodology." Artificial Intelligence and Law, Kluwer Academic Publishers, 1(1), pp. 45-64, March 1992.

[KL03] S. Kerrigan, K.H. Law. "Logic-based Regulation Compliance-assistance," In *Proc. of the Int'l. Conf. of the 9th Artificial Intelligence in Law (ICAIL'03)*, Edinburgh, Scotland, UK. pp. 126-135, June 2003.

[Mer04] R. Mercuri. "The HIPAA-potamus in Health Care Data Security." *Communications of the ACM*, 47(7), pp. 25-28, 2004.

[P3P] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall and J. Reagle. The Platform for Privacy Preferences (P3P), version 1.0, W3C Recommendation, http://www.w3.org/TR/P3P/

[SAH04] W. Stufflebeam, A. I. Antón, Q. He and N. Jain. "Specifying Privacy Policies with P3P and EPAL: Lessons Learned." In *Proc. 2004 ACM Workshop on Privacy in Electronic Society (WPES'04),* Washington, D.C., p. 35, October 2004.

[She87] D. M. Sherman. "A Prolog model of the income tax act of Canada" In Proc. of the 1st Int'l Conf. on Artificial Intelligence and Law (ICAIL-87), Boston, MA, USA, pp. 127-136, 1987.

[SH72] R.C. Shank. "Conceptual Dependency: A Theory of Natural Language Understanding," Cognitive Psychology, v. 3, no. 4, 1972, pp. 532 – 631.

[SO84] J.F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA, 1984.

[SSK86] M.J. Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond and H.T. Cory. "The British Nationality Act as a Logic Program" In *Communications of the ACM*, 29(5), pp. 370-386, May 1986.

[XACML] T. Moses (ed.) eXtensible Access Control Markup Language (XACML), version 2.0, Oasis Standard. http://xml.coverpages.org/xacml.html