# Compliance Engineering: Aligning Software Requirements with Policies and Government Regulations

Travis D. Breaux
Department of Computer Science
North Carolina State University
Raleigh, North Carolina, USA
tdbreaux@ncsu.edu

## ABSTRACT

As information is increasingly managed electronically, policies and government regulations intended to protect personal privacy are increasing the requirements complexity of software systems. These regulations and policies are frequently developed by lawyers and domain experts – not engineers – resulting in complex and ambiguous legal language. To ensure software complies with the law, software developers face the perilous challenge of distilling regulations into implementable software requirements. Furthermore, because regulations describe business processes and not individual software systems, auditors, managers and developers are faced with a daunting traceability quagmire when aligning regulations, business practices and requirements across an organization. To address these two challenges, I propose a framework that includes a methodology to distill regulations into stakeholder rights and obligations and a formal model to align rights and obligations with requirements. The methodology includes techniques to systematically reduce complexity, identify ambiguities and infer implied rights and obligations to improve requirements coverage. The model employs delegation and ownership to track the refinement of rights and obligations into implementable requirements across an organization. The framework will enable auditors to certify that delegation and refinement decisions that result in requirements comply with the intent of the law; thus transferring liability from software validation to software verification.

## Categories and Subject Descriptors

D.2.1 [**Requirements/ Specifications**]: Languages and Methodologies – *ambiguity,elicitation, refinement, traceability.*

## General Terms

Security, Human Factors, Standardization, Legal Aspects.

## 1. INTRODUCTION

Policies and regulations govern information practices such as those developed to comply with the U.S. Health Insurance Portability and Accountability Act[1] (HIPAA) in the health care industry. Increasingly, public and private organizations are using software to electronically manage information under the provenance of these regulations. To make matters more complex, these information practices span the scope of multiple software systems; each potentially owned or managed by a different stakeholder. This challenges policymakers, software engineers and system administrators to certify that, at a minimum, software systems do not violate the law and, moreover, that such systems promote overall compliance throughout an organization.

To understand the scope of this problem, consider recent events regarding the HIPAA Privacy Rule [18]: a set of regulations governing how companies and patients can access, share, and amend relevant medical information. The Office of Civil Rights (OCR) in the U.S. Department of Health and Human Services is responsible for enforcing the HIPAA Privacy Rule. By March of 2006, OCR had received over 18,900 HIPAA-related complaints including the improper use and disclosure of health information [19]. That same month, the HIPAA Enforcement Rule became effective, which imposes up to $25,000 in civil penalties for a single violation [11]. The HIPAA requires companies to certify they have policies and procedures in place to prevent improper use and disclosure of electronic health information. Without proper assurance that these policies are aligned with information systems, companies will be vulnerable to violations.

To this end, I propose a requirements engineering framework to align regulations with software requirements to help certify that software systems comply with the law. The remainder of this paper is organized as follows: terminology in Section 2; related work and approaches in Section 3; research methodology and proposed framework in Section 4; hypothesis and validation in Section 5; with contributions and summary in Section 6.

## 2. TERMINOLOGY

The following terms and definitions establish the key vocabulary used in the proposed framework [5, 6, 7]:

- *Rights* are statements that permit stakeholders to perform specific actions; possibly using software systems.
- *Obligations* are statements that require stakeholders to perform specific actions; possibly using software systems.
- *Conditions* are statements about the properties of individuals, ordering of events or the roles of individuals in activities. Conditions may also be rights or obligations.
- *Definitions* are statements that define the meaning of a noun phrase through a set of conditions. For example, the definition "health care providers are entities who provide health services" defines the noun phrase "health care providers" using the condition "entities who provide health services."
- *Delegation* is the act of one stakeholder to assign a right or obligation to another stakeholder.
- *Ownership* is the state of one stakeholder to be held solely accountable for satisfying an obligation, regardless if they delegated that responsibility to another stakeholder.

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

- *Refinement* is the act of one stakeholder to create a new right or obligation as a more specific interpretation of an existing right obligation; usually by including context-sensitive information from a specific business practice.

# 3. RELATED WORK

In requirements engineering, the actions of stakeholders have been modeled as goals [9] with methods to derive goals from natural language documents [1] and policies [2, 3]. Dardenne et al. propose the KAOS framework for acquiring conceptual models of stakeholder actions, called goals [9]. KOAS introduces several semantic primitives necessary to refine high-level goals that describe organizational objectives into low-level goals that describe specific processes or actions. Antón proposed the GBRAM methodology to extract goals from natural language documents, including process descriptions and stakeholder interviews [1]. GBRAM provides heuristics to identify actors, actions and constraints in natural language necessary to specify goals. Antón et al. have since applied techniques similar to GBRAM to extract goals from Internet privacy policies [2] and organize these goals into a privacy taxonomy [3].

At minimum, rights and obligations are Deontic extensions to the goal model, accounting for "what ought to be" and "what is permissible" [12]. However, because subtle misinterpretations of policies and regulations result in catastrophic legal violations, we developed a more precise methodology to trace the exact words in regulatory statements to finer semantic primitives than rights and obligations [4, 5]. These finer semantics are used to identify ambiguities in the natural language text and infer implied rights and obligations [7] to improve requirements coverage.

Giorgini et al. describe Secure-Tropos, a visual framework with a formal semantics in Datalog that enables modeling relationships among actors and goals using refinement, permission, delegation and ownership [13]. Massaci et al. have applied Secure-Tropos to the Italian Data Protection legislation [15]. Because policies and regulations span business practices, multiple stakeholders must align their rights and obligations with software requirements. Auditors evaluate these alignment decisions to determine their efficacy in achieving compliance. The model in Section 4 extends Secure-Tropos by encoding these stakeholder decisions in a distributed environment with stakeholders held accountable.

In software engineering, May et al. describe a methodology to extract formal models from regulations that they applied to one section in the HIPAA Privacy Rule [16]. Our work to extract formal models from four sections in the Privacy Rule [7] contradicts several of their basic assumptions, including: 1) each paragraph has exactly one rule; and 2) external and ambiguous references are satisfied by default [16]. Since their methodology lacks techniques to handle ambiguities and constraints acquired from cross-references, their models will be inconsistent with regulations like the HIPAA. Techniques to address this problem are described in Section 4.

# 4. RESEARCH PROPOSAL

I first describe the research methodology used to develop the proposed solution, before illustrating the framework methodology and formal model from the perspective of the user (a requirements or software engineer).

The research methodology used to investigate this problem is called Grounded Theory, proposed by Glaser and Strauss [10], which states theory developed from a data set is valid for that data set. When using Grounded Theory, one must scale existing theory to accommodate new datasets. Scaling existing theory requires incrementally identifying limitations in that theory, proposing new extensions to that theory, which address those limitations, and ensuring the new theory continues to support observations from previous data sets. To date, the framework has been developed by applying Grounded Theory to three datasets: the most frequent 100 goals in over 100 privacy policies [4, 5], a HIPAA Fact Sheet [6], and the HIPAA Privacy Rule [7].

## 4.1 The Methodology

The methodology to extract rights and obligations from regulation text is summarized before illustrating by example.

1) For each sentence in the regulation text, the user systematically applies phrase heuristics to classify the statement as a *definition*, *right*, *obligation* or *condition*. Heuristics include modality (can, may, must), condition key words (if, unless, except) and English conjunctions (and, or, not). Because modals and English conjunctions are ambiguous, the user must document their interpretation (e.g., "may" indicates a right) and assign logical meanings to each conjunction. Due to logical disjunctions, each sentence may have multiple rights and obligations.

2) The user must follow cross-references to relevant statements in other paragraphs, incorporating relevant conditions. Cross-references typically appear in conditions, for example, "the entity must receive consent consistent with paragraph (e)," refers to statements in paragraph (e) that describe conditions for consent.

3) The user separates each definition, right, obligation and condition and assigns a unique index and a reference to the originating paragraph for traceability. Each definition, right and obligation is assigned a logical formula comprised of the unique condition indices; each logical operator is mapped to the logical meaning assigned to the corresponding English conjunction.

Consider the regulation text summarized from the Privacy Rule §164.520(c)(2) and (c)(3) re-indexed as paragraph (a), below; this example appeared previously in [7]. The normative phrase (must) and condition words (if, unless) are **bold**, the condition phrases are underlined and the obligation phrases are *italicized*.

(a) Standard: *The covered entity **must** provide the individual notice.*
  (1) *A covered entity* who has a direct treatment relationship with an individual ***must*** …
    (A) *Provide notice* no later than the first service delivery;
    (B) **If** the covered entity maintains a physical delivery site:
      *i. Have the notice available for individuals to take.*
      *ii. Post the notice in a clear and prominent location.*
  (2) For the purposes of paragraph (a)(1), *a covered entity* who delivers service electronically, ***must*** *provide electronic notice* **unless** the individual requests to receive a paper notice.

Applying the methodology, the user derives constraints $C_1$–$C_5$ and obligations $O_1$–$O_5$, below. The covered entity is abbreviated to (CE). Because each phrase in the regulation is attributed to an extracted statement, at minimum this systematic approach provides statement-level requirements coverage for an entire document. The *italicized* phrases in obligations $O_2$, $O_4$, and $O_5$ are ambiguities resolved by inference using the regulation text. The original paragraph numbers follow condition and obligation statements; the condition indices ($C_x$) appear in logical formulas in square brackets after the obligation statements.

**Conditions:**

**$C_1$:** The CE has a direct treatment relationship with the individual. (a)(1)
**$C_2$:** The notice is provided no later than the first service delivery. (a)(1)(A)
**$C_3$:** The CE maintains a physical delivery site. (a)(1)(B)
**$C_4$:** The CE delivers health service electronically. (a)(2)
**$C_5$:** The individual requests to receive a paper notice. (a)(2)

**Obligations:**

**$O_1$:** The CE must provide notice to the individual (a).
**$O_2$:** The CE must provide notice *to the individual.* (a)(1)(A) [$C_1 \wedge C_2$]
**$O_3$:** The CE must have the notice available for individuals to take. (a)(1)(B)(i) [$C_1 \wedge C_3$]
**$O_4$:** The CE must post the notice in a clear and prominent location *for the individual to read*. (a)(1)(B)(ii) [$C_1 \wedge C_3$]
**$O_5$:** The CE must provide electronic notice *to the individual.* (a)(2) [$C_1 \wedge C_2 \wedge C_4 \wedge \neg C_5$]

Right, obligation and condition statements are often distributed across several different sections in the Rule. For example, the subject CE is specified in paragraph (a)(1); however, the obligation phrases *provide notice*, *have notice available*, and *post the notice* each appear in paragraphs (a)(1)(A), (a)(1)(B)(*i*), and (a)(1)(B)(*ii*), respectively. The constraint $C_1$ appearing in sub-section (a)(1) is applied across each of these obligations and the obligation in paragraph (a)(2), due to a cross-reference back to (a)(1). Because each section is written from a different viewpoint, cross-references to other sections require the user to interpret the referenced statements across contexts.

Techniques to specify rights and obligations at the word-level using logical models have also been developed [4, 5]. This degree of specification is used to: algorithmically identify ambiguities; infer implied rights and obligations; and organize statements into abstraction hierarchies [7].

## 4.2 The Model

Effective policies and regulations assign individual stakeholders the responsibility to create policies and procedures for high-level goals – a process called refinement. To align policies with system requirements, the following model traces rights and obligations from policies to system requirements. As an overview, I present a subset of the model consisting of the set A of actors, the set S of systems and the set O of obligations. This subset contains the following sets of relations:

The *assignment set* $AS \subseteq (A \cup S) \times O$ consists of pairs (*subject*, *goal*) such that each actor must satisfy or delegate the *goal* to other actors or systems; and systems must satisfy the goal.

The *ownership set* $OS \subseteq (A \times O)$ consists of pairs (*owner*, *goal*) such that each *goal* has exactly one *owner* who is solely responsible for ensuring that the *goal* is satisfied.

The *delegation set* $DS \subseteq (A \times O \times (A \cup S))$ consists of triples $\langle$*subject*, *goal*, *target*$\rangle$ in which a *subject* delegates their assigned *goal* to a *target* (actor or system), resulting in (*target*, *goal*) $\in$ AS. We assume a permission framework exists to ensure authorized delegation.

The *refinements set* $RS \subseteq (A \times O \times O)$ consists of triples $\langle$*actor*, *goal*, *sub-goal*$\rangle$ in which an *actor* refines an assigned *goal* into a *sub-goal*, resulting in (*actor*, *sub-goal*) $\in$ AS. For an (*actor*, *goal*) pair, the possibly empty set of all *sub-goals* that refine *goal* is called a *refinement strategy*.

Several properties determine if the model is well-formed, including: consistency among assignment, delegation and refinement sets with no cycles; the verifiability of "satisfied" obligations; and the existence of assigned rights necessary to satisfy assigned obligations. The model has been validated using the little language Alloy [14].

As an example, consider obligation $O_5$ from Section 4.1. The HIPAA Privacy Rule assigns $O_5$ to an actor, such as a Chief Privacy Officer (CPO), who then delegates this obligation to a system administrator (SA). For a particular system T, in which constraint $C_4$ maps to $C_6$, and the system administrator refines $O_5$ into requirements $O_6$, $O_7$, below:

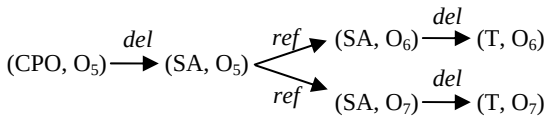
**Conditions:**

**$C_6$:** The system delivers health services to the individual.

**Obligations:**

**$O_6$:** The system shall maintain an electronic copy of the notice.
**$O_7$:** The system shall provide electronic notice to the individual who receives services. [$C_1 \wedge C_2 \wedge \neg C_5 \wedge C_6$]

The decision sequences for implementing obligation $O_5$ are traced in the following directed graph. Arrows, labeled *del* and *ref* for delegation and refinement, respectively, show the flow of decisions resulting in new obligation assignment pairs.




Auditors can certify (by digital signature) that these decision sequences comply with the intent of the law; the certified decision sequences are compliance artifacts. Decision sequences may periodically require re-certification such as when they change due to policy evolution or organization restructuring.

In the example, unresolved questions include: must the CE electronically notify individuals of changes to the notice; if so, who notifies the system administrator of such changes; does the administrator have sufficient rights to delegate requirements to all systems that deliver electronic notices? The model provides an extensible foundation to track unresolved issues like these as model inconsistencies.

# 5. VALIDATION

Two hypotheses stand to be tested: 1) The methodology applied to the same regulatory text produces the same set of rights and obligations, independent of the user; and 2) for auditors and organizations, the model improves precision and reduces effort in the identification and certification of systems that must comply

with regulations. The validation must also establish the relevance of this framework to practitioners in government and industry.

The first hypothesis has received preliminary validation in a pilot [6] and case study [7]. A larger study is planned with between 20-30 participants that will employ modern experimental design practices [8] to establish internal validity, including control and methodology groups, randomization across groups to distribute uncontrolled co-factors, etc. Special consideration will be given to avoid Simpson's Paradox [17] and the Hawthorne effect [8].

To validate the second hypothesis, I am designing a quasi-experiment for small health care providers (HCP) who use electronic information systems governed by the HIPAA. In the control group, the HCP will be given fictitious HIPAA violations and asked to identify software features that contributed to the violation and administrators responsible for that software. This experiment will seek to evaluate the time required and precision in identifying the responsible systems and personnel. In the model group, we will employ a multi-user, software prototype based on the proposed model and seeded with rights and obligations from HIPAA regulations. The HCP will align these regulations with their business practices and information systems using the prototype. Similar to the control group, the HCP will be asked to use the prototype to identify the software features that affected the fictitious HIPAA violations. Three factors complicate comparing the control and model group results: 1) the time and cost to recruit participants and conduct the experiment limits the number of participants in each group; 2) co-factors such as organizational structure, human factors, business practices and software systems can vary significantly between HCPs; and 3) the act of an HCP performing in the control group for a fictitious HIPAA violation will bias their performance in the model group for a related violation, and vice versa. To address these difficulties, we will explore experimental design practices in other disciplines to establish comparable and verifiable results.

## 6. CONTRIBUTION & SUMMARY

The contributions of this research will include:

1. A methodology that provides engineers a systematic approach to disambiguate natural language regulations to specify regulatory compliant stakeholder rights and obligations.
2. A model that provides an accounting of stakeholder decisions to delegate rights and obligations to other stakeholders and refine obligations into system requirements. Decision chains can be certified by auditors as compliance artifacts.
3. A software prototype based on the proposed model will enable codifying rights and obligations and tracking delegation and refinement decisions to acquire these compliance artifacts.

In this paper, I describe a framework consisting of a methodology to extract rights and obligations from policies and regulations and a model to align these artifacts with software requirements. The framework provides a foundation to engineer compliance from policy statements to software requirements.

## 7. ACKNOWLEDGMENTS

I thank my adviser, Dr. Annie Antón, for her guidance and support. This work was supported in part by CERIAS at Purdue, NSF ITR #032-5269; and ITR CyberTrust #NSF 043-0166.

## 8. REFERENCES

[1] A.I. Antón, "Goal-based requirements analysis," *2nd IEEE Int'l Conf. Requirements Engineering*, pp. 136-144, 1996.

[2] 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.

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

[4] 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.

[5] 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.

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

[7] 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.*, 2006.

[8] D.T. Campbell, J.C. Stanley, *Experimental and quasi-experimental designs for research*, Houghton-Mifflin Co., Boston, MA, 2005.

[9] A. Dardenne, A. van Lamsweerde, S. Fickas, "Goal-directed requirements acquisition," *Science of Computer Programming*, 20(1-2):3-50, 1993.

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

[11] HIPAA Administrative Simplification: Enforcement, 45 CFR Parts 160, 164, *U.S. Fed. Reg.*, 71(32):8389-8433, 2006.

[12] J.F. Horty, *Agency and Deontic Logic*, Oxford University Press, New York NY, 2001.

[13] P. Giorgini, F. Massacci, J. Mylopoulos, N. Zannone, Modeling security requirements through ownership, permission and delegation," *13th IEEE Int'l Conf. Req. Engr.*, pp. 167-176, 2005.

[14] D. Jackson, "Alloy: a lightweight object modeling notation," *ACM Trans. Soft. Engr. Meth.* 11(2): 256-290, 2002.

[15] F. Massaci, M. Prest, N. Zannone, "Using a security requirements engineering methodology in practice: the compliance with the Italian Data Protection legislation," *Computer Standards & Interfaces*, 27(5):445-455, 2000.

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

[17] E.H. Simpson, "The interpretation of interaction in contingency tables" Journal of the Royal Statistical Society, B(3):238-241.

[18] Standards for Privacy of Individually Identifiable Health Information, 45 CFR Parts 160, 164, *U.S. Fed. Reg.*, 67(157): 53181-53273, 2002.

[19] W. Wilkinson, "The office for civil rights and health care privacy," *12th National HIPAA Summit*, Washington, D.C., April 10, 2006.