ABSTRACT

BREAUX, TRAVIS DURAND. Legal Requirements Acquisition for the Specification of Legally Compliant Information Systems. (Under the direction of Ana Isabel Antón.)

U.S. federal and state regulations impose mandatory and discretionary requirements on industry-wide business practices to achieve non-functional, societal goals such as improved accessibility, privacy and safety. The structure and syntax of regulations affects how well software engineers identify and interpret legal requirements. Inconsistent interpretations can lead to non-compliance and violations of the law. To support software engineers who must comply with these regulations, I propose a Frame-Based Requirements Analysis Method (FBRAM) to acquire and specify legal requirements from U.S. federal regulatory documents. The legal requirements are systematically specified using a reusable, domain-independent upper ontology, natural language phrase heuristics, a regulatory document model and a frame-based markup language. The methodology maintains traceability from regulatory statements and phrases to formal properties in a frame-based model and supports the resolution of multiple types of legal ambiguity. The methodology is supported by a software prototype to assist engineers with applying the model and with analyzing legal requirements. This work is validated in three domains, information privacy, information accessibility and aviation safety, which are governed by the Health Insurance Portability and Accountability Act of 1996, the Rehabilitation Act Amendments of 1998, and the Federal Aviation Act of 1958, respectively.

Legal Requirements Acquisition for the Specification of Legally Compliant Information Systems

by
Travis Durand Breaux

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2009

APPROVED BY:

Annie I. Antón Chair of the Advisory Committee	David L. Baumer
Jon Doyle	Eugene H. Spafford
Mladen A. Vouk	

BIOGRAPHY

Travis D. Breaux received the Bachelors of Arts in Anthropology from the University of Houston in May 1999 and subsequently served as a volunteer in the United States Peace Corps in Mongolia. Upon his return, he transitioned from anthropology to computer science when he received the Bachelors of Science in Computer and Information Science (with Honors) from the University of Oregon in December 2003 before enrolling in the Computer Science doctoral program at North Carolina State University the following year.

Dr. Breaux is a 2006-09 IBM Ph.D. Fellowship recipient, 2008-09 NCSU Preparing the Professoriate Fellowship recipient, the 2006-07 Walker H. Wilkinson Research Ethics Fellowship recipient, a 2005-06 CISCO Information Assurance Scholarship recipient and he has received undergraduate awards for academic excellence from programs funded by the U.S. Department of Energy and National Science Foundation. Dr. Breaux was a Visiting Scholar at CERIAS at Purdue University in summer 2006, an intern at the IBM T.J. Watson Research Center in summer 2005 and an intern at the Oak Ridge National Laboratory in summers 2003, 2004. Dr. Breaux has several publications in ACM and IEEE-sponsored journals and proceedings of conferences, symposia and workshops.

Dr. Breaux traces his passion for requirements engineering back to teachings on culture cosmology and representations of the world by Dr. Susan Rasmussen and Dr. Quetzil Casteñeda at the University of Houston. Dr. Breaux was first introduced to the field of Requirements Engineering by his undergraduate adviser, Dr. Stephen Fickas, at the University of Oregon whose influence includes requirements monitoring, requirements negotiation and ephemeral requirements. Under the guidance of Dr. Annie Antón, Dr. Breaux has extended his interests to include the societal impact of system requirements on privacy and security in their "ground-breaking" work to acquire software requirements from policies and U.S. federal and state regulations.

Appointments and Tenure

- (2005-2009) Research Assistant at North Carolina State University under Dr. Annie Antón
- (2006, Summer) Visiting Scholar at the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University under Drs. Annie Antón and Eugene Spafford
- (Summer 2005) Research Assistant at IBM Thomas J. Watson Research Laboratory under Drs. Clare-Marie Karat and John Karat
- (2004, Summer) Research Assistant at Oak Ridge National Laboratory under Dr. Robert Patton

- (2004-2005) President of the Computer Science Graduate Student Association: Elected President for the academic year 2004-2005; in that year, the association represented the interests of 384 graduates students (127 Doctoral, 257 Masters) in the Department of Computer Science at NCSU.
- (2003, Summer) Research Assistant at Oak Ridge National Laboratory under Dr. Thomas Potok
- (2000-2001) Peace Corps Volunteer Representative: Elected in August 2000; in that year, eight representatives represented the interests of 81 volunteers from across Mongolia to the local Peace Corps administration.

Supporting Publications (Chronologically Ordered)

[1] T.D. Breaux, "Exercising Due Diligence in Legal Requirements Acquisition: A Tool-supported, Frame-based Approach", *17th International Requirements Engineering Conference*, Atlanta, Georgia, September 2009.

[2] T.D. Breaux, C. Powers, "Early Studies in Acquiring Evidentiary, Reusable Business Process Models for Legal Compliance", *6th International Conference on Information Technology - New Generations*, Las Vegas, Nevada, April 2009.

[3] T.D. Breaux, J.D. Lewis, P.N.H. Otto, A.I. Antón, "Identifying Legal Vulnerabilities and Critical Requirements Using Criminal Court Proceedings", *24th ACM/SIGAPP Symposium on Applied Computing (ACM SAC'09)*, Honolulu, Hawaii, April 2009, pp. 355-359.

[4] T.D. Breaux, A.I. Antón, E.H. Spafford, "A Distributed Requirements Management Framework for Compliance and Accountability", *Elsevier Computers and Security (COSE)*, 28(1-2): 8-17, February/March 2009.

[5] T.D. Breaux, A.I. Antón, J. Doyle, "Semantic Parameterization: A Conceptual Modeling Process for Domain Descriptions," *ACM Transactions on Software Engineering Methodology*, 18(2): Article 5, November 2008.

[6] N. Kiyavitskaya, N. Zeni, T.D. Breaux, A.I. Antón, J. Gordy, L. Mich, J. Mylopoulos, "Automating the Extraction of Rights and Obligations from Regulations," *International Conference on Conceptual Modeling*, October 2008, pp. 154-168.

[7] T.D. Breaux, A.I. Antón, K. Boucher, M. Dorfman, "Legal Requirements, Compliance and Practice: An Industry Case Study in Accessibility," *IEEE 16th International Requirements Engineering Conference*, Barcelona, Spain, September 2008, pp. 42-52.

[8] T.D. Breaux, A.I. Antón, "Analyzing Regulatory Rules for Privacy and Security Requirements," *IEEE Transactions on Software Engineering*, 34(1): 5-20, January 2008.

[9] N. Kiyavitskaya, N. Zeni, L. Mich, T.D. Breaux, A.I. Antón, J. Mylopoulos, "Extracting Rights and Obligations from Regulations: Towards a Tool-Supported Process," *IEEE/ACM 22nd International Conference Automated Software Engineering*, November 2007, pp. 429-432.

[10] T.D. Breaux, A.I. Antón, "A Systematic Method for Acquiring Regulatory Requirements: A Frame-based Approach," *6th International Workshop on Requirements for High Assurance Systems*, New Delhi, India, September 2007.

[11] T.D. Breaux, M.W. Vail, A.I. Antón, "Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations," *IEEE 14th International Requirements Engineering Conference*, Minneapolis, MN, September 2006, pp. 49-58.

[12] T.D. Breaux, A.I. Antón, Clare-Marie Karat and John Karat, "Enforceabiltiy vs. Accountability in Electronic Policies," *IEEE 7th International Workshop on Policies for Distributed Systems and Networks*, London, Ontario, June 2006, pp. 227-230.

[13] T.D. Breaux, A.I. Antón, "Mining Rule Semantics to Understand Legislative Compliance," *ACM Workshop on Privacy in the Electronic Society*, Alexandria, VA, November 2005, pp. 51-54.

[14] T.D. Breaux, A.I. Antón, "Analyzing Goal Semantics for Rights, Permissions and Obligations," *IEEE 13th International Requirements Engineering Conference*, Paris, France, September 2005, pp. 177-188.

[15] T.D. Breaux, A.I. Antón, "Deriving Semantic Models from Privacy Policies," *IEEE 6th International Workshop on Policies for Distributed Systems and Networks*, Stockholm, Sweden, June 2005, pp. 67-76.

ACKNOWLEDGMENTS

Special thanks to Mr. Calvin Powers for his mentorship under the IBM PhD Fellowship.
Mr. Powers opened a window in a tower to a world of practice.

Special thanks to Dr. Annie Antón, Dr. Jon Doyle, Dr. John Mylopoulos
and Dr. Eugene Spafford
for their supportive research and career development advice.

This work was supported, in part, by: the IBM PhD Fellowship funded by the IBM Center for Advanced Studies, Research Triangle Park, North Carolina; National Science Foundation (NSF) Information Technology Research (ITR) Grant No. 032-5269, "Encoding Rights, Permissions and Obligations: Privacy Policy Specification and Compliance;" NSF CyberTrust Grant No. 043-0166, "Policy-Driven Framework for Online Privacy Protection;" and the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University, West Lafayette, Indiana.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

To know the laws is not to memorize their letter but to grasp their full force and meaning.
Marcus Tullius Cicero (106–43 B.C.)

In a society based on the rule of law, individual and organizational actions are governed by laws that serve to achieve societal goals. As law-based societies shift emphasis from industrial and manufacturing-driven economies to information and knowledge-driven economies, software-intensive information systems will increasingly support these actions in the production and delivery of goods and services. Laws that govern information systems pose significant compliance challenges to relevant stakeholders, including corporate executives, lawyers and software engineers. *This dissertation addresses the challenge of identifying relevant legal requirements from policies, laws and guidance documents and aligning these requirements with software specifications to maintain a defensible position in a court of law.*

Corporate executives and lawyers are steadfastly concerned about the seriousness of this compliance problem. During 2005-2008, an annual Ernst and Young survey of over 1,100 organizations revealed that the top two drivers of information security practice are compliance with regulations and data privacy protection [54, 55, 56]. A 2006 CSO Magazine survey [49] and a 2005 CIO Insight Magazine survey [43] both found that significant software expenditures are being directed at improving security to reduce non-compliance with regulations. However, improved security will not address other important non-functional requirements, such as safety, accessibility and transparency requirements, which are also required to comply with government laws and regulations.

The penalties for legal non-compliance can be severe. ChoicePoint, an "information broker", acknowledged that records on more than 163,000 consumers were acquired by identity thieves [57]. ChoicePoint's products span multiple personal information sources, including U.S. federal, state and local license, court, insurance and utility documents [133]. A prior review of ChoicePoint's business products suggests that the company had, with or without intent, developed these products without proper controls mandated by the Fair Credit Reporting Act (FCRA), 15 U.S.C. §1681 et seq. [82]. Under the FCRA, the U.S. Federal Trade Commission fined ChoicePoint $15 million in civil penalties and consumer redress and requires ChoicePoint to undergo biennial security audits

for 20 years from the date of the enforcement action [57, 156]. Hoofnagle and Solove believe that violations similar to the one by ChoicePoint result from how regulations are interpreted by companies in the context of their software specifications and designs [82]. In addition to civil fines and consumer redress, penalties may include legal fees, reengineering costs, public harms, consumer churn and loss of public trust.

To avoid penalties from regulatory enforcement actions, relevant stakeholders must demonstrate *legal compliance*, which consists of "efforts to maintain a defensible position in a court of law" [34, 140]. Proof of legal compliance is prepared, in part, by demonstrating through evidence-based mechanisms three principled stakeholder behaviors:

- *Due diligence*, which means "reasonable efforts that persons make to satisfy legal requirements or discharge their legal obligations";
- *Good faith*, which means "honesty in belief or purpose, faithfulness to one's duty or obligation, observance of reasonable commercial standards of fair dealing in a given trade or business, or absence of intent to defraud or to seek unconscionable advantage"; and
- *Standard of care*, which 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" [68].

Due diligence, good faith and a reasonable standard of care require that software engineers maintain evidence that unequivocally represents that: (1) software requirements correctly align with relevant laws; (2) software specifications correctly implement aligned software requirements; and (3) aligned software requirements are verifiable through testing at compile time or monitoring at runtime. This dissertation is limited to the first evidentiary challenge with a three-part thesis that *a methodology exists to acquire legal requirements from U.S. federal regulations, which is*:

1. *Consistent*, meaning there are no false-positives by showing that the method does not produce indecision about which legal statements map to which kind of legal requirements;
2. *Complete*, meaning there are no false-negatives by showing that the method includes all legal requirements based on a finite classification of legal statements; and
3. *A partial solution to the alignment of software requirements with relevant laws*, by showing that eight legal requirement refinement patterns provide reasonable explanations for the kinds of gaps that exist between legal and software requirements.

Unlike conventional software requirements, legal requirements have three fundamentally distinct characteristics that are supported by this dissertation:

1. Legal requirements govern multiple industries, goods and services, whereas traditional practice focuses on software requirements target specific systems;

2. Legal requirements are not elicited by engineers from stakeholders, they are codified in legal language and interpreted therefrom; and

3. Ambiguity cannot be removed from legal requirements by software engineers, it can only be classified and interpreted in the context of organizational practices, goods and services.

The remainder of this chapter is organized as follows: background on the lifecycle of the U.S. federal rulemaking process and terminology are presented in Section 1.1; the theoretical foundations upon which this dissertation is based are presented in Section 1.2; and related work describing other methods, frameworks and models that cover related aspects of the problem addressed by this dissertation are presented in Section 1.3.

1.1 Legal Background

In the United States, federal laws exist and evolve through a complex legal lifecycle that involves numerous parties, professions and viewpoints. This section reviews the relevant part of this lifecycle that is necessary for understanding federal regulations for the purposes of this dissertation.

1.1.1 U.S. Federal Legislation and Rulemaking

The relevant lifecycle begins with an *act* promulgated by the U.S. Legislature. Each act consists of: (1) the societal and economic goals that the federal government seeks to achieve; (2) the responsible Executive branch and independent agencies charged with achieving those goals; and (3) the powers of enforcement, if any, that are granted to these agencies. Executive branch agencies include the Food and Drug Administration, under the Department of Health and Human Services, which is responsible for regulating medical devices. Independent agencies include the Nuclear Regulatory Commission, which is responsible for regulating nuclear reactor safety. All public acts of law are codified and maintained in the United States Code (U.S.C.). Because a single act of law can create and revise multiple sections of the U.S.C., it is useful to cite acts using their public law number, such as "Public Law No. 104-191, 110 Stat. 1936," which refers to the 104th session of U.S. Congress, the 191st act ratified in that legislative year, and published in volume 110, page 1936 of the Statues at Large, the official compilation of laws and resolutions passed by Congress.

The Administrative Procedures Act (APA) of 1941, which is codified in 5 U.S.C. Subchapter II, establishes the *rulemaking* procedure for formulating, amending and appealing U.S. federal regulations. Section §553 of the APA defines rulemaking generally in three steps: (1) the agency publishes a Notice of Proposed Rulemaking (NPRM) in the Federal Register, the official publication for chronicling changes to the Code of Federal Regulations (CFR); (2) interested persons are then granted an opportunity to participate by submitting public comments, including written data, views, and arguments, to be considered by the rulemaking agency; and (3) the agency publishes the final rule in the Federal Register with a general, corresponding statement of the basis and purpose. The publication of the final rule may also include printed responses from the agency to specific

public comments, sometimes in the form of changes to the proposed rule. The legal lifecycle of a regulation may continue, years after the publication of the final rule, by agencies issuing guidance documents to clarify ambiguity in final rules and to answer frequently asked questions from the public. The rulemaking process applies to *substantiative regulations* that comprise new law, whereas *interpretative* regulations consist of guidance documents and agency responses to industry letters that are not required to meet the rulemaking process requirements [13].

The National Archives and Records Administration publishes the Federal Register that records the official chronology of changes to the Code of Federal Regulations (CFR). The National Archives also maintains the drafting guidelines for writing proposed and final rules [131]. These guidelines describe the CFR structure as a descending hierarchy with at least five levels: Title, Chapter, Part, Section and Paragraph. Paragraphs can be sub-divided into no more than six levels indicated by lowercase letters, Arabic numerals, Roman numerals, uppercase letters, Arabic numerals and Roman numerals, each level in parenthesis and in that order. For example, citations to U.S. federal regulations are of the form §164.520(a)(2)(i)(A), which refers to Part 164, Section 520, paragraph (a) and three additional levels (2), (i), (A), respectively. The Title and Chapter numbers, both uppercase Roman numerals, do not usually appear in citations.

The sections of a proposed or final rule often include: the Purpose, describing the societal and economic goals that the rule seeks to achieve; the Applicability and Scope, describing who and what actions are affected; and Definitions, wherein each definition describes a *term of art* that is a precise phrase with a specialized meaning in a specific subject area [68].

In addition to federal regulations, non-government organizations (NGOs) create guidance documents and standards that are recognized or adopted by government agencies. For example, the Radio Technical Commission for Aeronautics, an NGO, develops and maintains a series of guidance documents including DO-178b, titled "Software Considerations for Airborne Systems and Equipment Certification" [141, 142]. The Federal Aviation Administration, an agency under the Department of Transportation, accepts DO-178b as a guidance document by which manufacturers certify their airborne software systems.

1.1.2 U.S. Federal Regulations Studied

Three U.S. federal regulations are studied and discussed throughout this dissertation:

1. Part 164, Subpart E, titled "Privacy of Individually Identifiable Health Information" (the Privacy Rule) [125, 126, 127, 128]. The Privacy Rule governs the use and disclosure of patient medical information and is one of three regulatory rules, in addition to the Security Rule and Enforcement Rule. The Privacy Rule is maintained by the Office of Civil Rights in the U.S. Department of Health and Human Services to comply with the Health Insurance Accountability and Portability Act (HIPAA) of 1996[1] that affects numerous sections of the U.S.C.

[1] Public Law No. 104-191, 110 Stat. 1936 (1996)

2. Part 1194, titled "Electronic and Information Technology Accessibility Standards" (Accessibility Standards) [157]. The Accessibility Standards govern access to information by individuals with disabilities and is developed by the U.S. Access Board, an independent Federal agency, to comply with Section 508 of the Rehabilitation Act Amendments of 1986[2] and 1998[3] (29 U.S.C. §794d).

3. Parts 1, 21, 25, 33, 121, and 135, titled "Extended Operations (ETOPS) of Multi-Engine Airplanes" [58, 59, 60, 61, 62]. The ETOPS governs the certification of maintenance programs for multi-engine aircraft that fly long distance routes and is developed by the U.S. Federal Aviation Administration (FAA) of the Department of Transportation. Unlike the Privacy Rule and the Accessibility Standards, ETOPS is developed by the FAA through their general authority to regulate airplane safety established under the Federal Aviation Act of 1958[4] (49 U.S.C. §44701, etc.). ETOPS is intended to harmonize U.S. federal aviation regulations with the International Civil Aviation Organization (ICAO), an international aviation standards body, created under the Convention on International Civil Aviation, an international treaty to which the United States is a signatory state.

The Privacy Rule and the Accessibility Standards bear significant cost and impact to society in the United States. Among other regulations, the U.S. Office of Management and Budget identified these two regulations as "economically significant" or "major" rules that have a net present or annual cost to society in excess of $100 million [130]. As of September 2001, the Privacy Rule had a net present cost of $11.8 billion while the Accessibility Standards had an annual cost of between $177-1,068 million, over half of which was estimated as a Federal government cost burden [130]. In addition to cost, these two regulations have a significant impact to society. The Privacy Rule affects the healthcare industry, which employs over 14 million individuals across 538,000 establishments, including hospitals and private practices [37]. By March 2006, the Office of Civil Rights had received over 18,900 complaints regarding potential violations of the Privacy Rule, only 72% of which were fully resolved and nearly 300 of which were forwarded to the U.S. Department of Justice for criminal investigation [163].

The Accessibility Standards affect companies that sell information access-related products, such as computer and software systems, to U.S. federal agencies. Because the public purchases the same products as federal agencies do from companies such as Cisco, IBM, Oracle and Microsoft, the societal benefit of the Accessibility Standards is extended to the general population. In the United States, this includes 36.4 million adults who experience some hearing difficulty without a hearing aid, 20.2 million adults who report trouble seeing even with glasses or contact lenses, and 32.4 million adults who have limited reach or mobility [137]. The ETOPS governs over 10,000 domestic U.S. flights per year and affects over 3 million passengers on commercial aircraft, annually [38]. Methods that improve a company's ability to comply with these regulations will have a direct and

[2]Public Law No. 99-506, 100 Stat. 1807 (1986)
[3]Public Law No. 105-220, 112 Stat. 936 (1998)
[4]Public Law No. 85-726, 72 Stat. 731

positive effect on these individuals.

1.1.3 Legal Language and Engineering Challenges

This section discusses the three fundamental challenges faced by engineers who extract legal requirements from legal text: ambiguity, traceability and accountability.

Ambiguity

U.S. federal regulations contain ambiguities that are intended by law makers to be re-interpreted as business practices evolve and as capabilities to comply with regulations change over time. For example, HIPAA §164.512(e)(1)(iv) states that an entity must make "reasonable" efforts to notify individuals of certain requests for their protected health information. The word "reasonable" is an intended ambiguity: exactly which mechanisms are considered reasonable, (e.g., postal mail, secure electronic mail or websites, etc.) varies depending on the type of communities served and the prevalence of relevant, existing technologies.

Lawmakers also define governed entities using terms that are open to interpretation. For example, in HIPAA §164.304, the term "workstation" is exemplified by "a laptop or desktop computer, or any other device that performs similar functions." Compliance officers must decide if this definition is intended to cover handheld Personal Digital Assistants (PDAs). As PDAs increase in computational power and become better integrated into routine business practices, organizations may need to re-interpret this ambiguity to achieve compliance.

Regulations also contain unintended ambiguities that are inherent to natural language syntax and semantics. Four types of ambiguity are distinguished and addressed by this dissertation: logical ambiguity, attributive ambiguity, referential ambiguity and under-specification. Because these ambiguities can affect the interpretations of multiple, related requirements in a requirements document, Kamsties classifies these ambiguities as requirements document ambiguity [99].

Logical ambiguity refers to English words that can be mapped to different logical interpretations. Herein, we only consider how English conjunctions (and, or) can be assigned conflicting logical connectives; see Berry and Kamsties for a separate discussion of universal and existential qualification-related ambiguity [16]. For example, in HIPAA §164.524(a)(1), individuals have "a right of access to inspect and obtain" a copy of their protected health information. While this statement uses the English conjunction "and," presumably individuals can obtain a copy of their information without needing to inspect the information; e.g., the conjunction can be interpreted as a logical-or. In contrast, interpreting this conjunction as a logical-and may lead to systems that provide the information such that an inspection is required and confirmable.

Attributive ambiguity is found in phrases that may be reasonably ascribed to more than one other phrase within a sentence. For example, in HIPAA §164.520(b)(1)(vii), the statement "the [privacy] notice must contain the name or title and telephone number of a person or office" may be construed to mean the notice contains one of: (1) the name of the person or office; (2) the title

and telephone number of the person or office; or (3) the name and telephone number of the person or office. Because the phrase "and telephone number" can be attributed to the "name and title" or only the "title," the engineer may interpret either options (1) and (2) or options (2) and (3) as valid interpretations. The former interpretation permits the organization to withhold the telephone number from the privacy notice, making it more difficult for recipients of the notice to contact the person or office.

Referential ambiguity occurs when a word or phrase has multiple meanings; this includes intensional and extensional polysemy [33]. This dissertation considers a type of extensional polysemy in which words have an *anaphoric* (backward-referencing) or *cataphoric* (forward-referencing) function. These words include pronouns (this, that, they), noun phrases that use definite articles (the) and some adjectives (such). A statement that contains the phrase "must provide such notices" refers to notices that are elaborated upon in the broader context of this statement or paragraph. The engineer must identify additional implications or constraints on the "notices" that appear in the broader context before determining which notices must be provided. In addition, cross-references between paragraphs introduce referential ambiguity because a single paragraph can contain multiple statements, not all of which are relevant to a cross-reference. Gause and Weinberg consider a third kind of referential ambiguity that is quantitative (e.g., words such as small, inexpensive, etc.) [69], which is not addressed in this dissertation.

Under-specification or omission occurs when an expected word or phrase is missing from a sentence. For example, §1194.21(f) in the Accessibility Standards states "Textual information shall be provided through operating system functions for displaying text." This statement describes *how* textual information is provided but does not state *who* or *what* provides this information. Unlike missing requirements [69], under-specifications include word and phrase omissions from documented requirements, some of which can be identified using the Inquiry-cycle model [139]. The engineer must identify these omissions and seek to disambiguate or at times make reasonable assumptions about which classes of entities the statement is intended to reflect.

Traceability

Regulations present traditional and novel traceability challenges to engineers. Similar to other requirements sources (e.g., interviews, scenarios and use cases), the loss of original context also affects requirements that are extracted from regulations. Unlike these other sources, the "context" of a regulatory statement is distributed across multiple sections, paragraphs and sub-paragraphs of the source document. Engineers must reconstruct this context by employing knowledge of the regulatory document structure and the cross-reference syntax. For example, a regulatory statement can start in one paragraph and end in a sub-paragraph; this break is called a *continuation* or a continuance by Wilson et al. [164]. Consider the following continuation in HIPAA §164.520(a)(2)(ii) that describes two requirements (obligations) to maintain and provide a privacy notice to patients:

(ii) A group health plan must:
(A) Maintain a notice under this section; and
(B) Provide such notice to any person

During requirements extraction, traceability must be maintained among unique paragraph indices and corresponding requirements to map paragraph cross-references back to those requirements [36, 31]. Because legal complaints often cite compliance failures using cross-references to paragraphs in a regulation, demonstrating which paragraphs map to which requirements is necessary to demonstrate due diligence. Therefore, the paragraph index (ii) in the example continuation above should trace to both requirements (to maintain and provide), whereas the paragraph index (ii)(A) should only trace to the maintenance requirement and paragraph index (ii)(B) should only trace to the provision requirement.

Accountability

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 assumed to be compliant. Separately, a software system is accountable under a regulation if, for every permissible, obligatory 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. Alternatively, an accountable system can demonstrate which regulatory rules apply to every transaction and can produce a corresponding audit trail [34, 25]. Improving accountability supports demonstrating due diligence and improves compliance, whereas a compliant system may not be accountable at all and thus fails to provide evidence of due diligence. 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 the methodology in this dissertation itemizes constraints and prioritizes requirements helps software engineers achieve accountability by this definition.

1.2 Theoretical Foundations

The theoretical foundations of this dissertation include fundamental theory to understand: (1) the relationship between requirements and specifications; (2) the relationship between formalism and phenomenology in requirements engineering; (3) the types of knowledge that appear in requirements; and (4) the relevance of traceability in requirements engineering. What follows is a brief overview of this work with respect to the field of requirements engineering.

1.2.1 The Environment and the Machine

Zave and Jackson champion the need to understand and describe the environment of a machine (e.g., software and hardware), called the *domain*, as an equally important engineering task as developing requirements and design specifications [93, 167]. In their view, it is not necessary to describe the machine to be built, however abstractly, within the scope of requirements. However, they believe it is necessary to distinguish which actions are controlled by the environment from those actions controlled by the machine [93, 167]. For example, the Accessibility Standards have a "product focus" in which most regulatory statements describe system constraints or actions that are shared with the machine (e.g., telephony equipment, web browsers, etc.). However, the HIPAA Privacy Rule has a "stakeholder focus" in which most statements describe stakeholder actions that may or may not share phenomena with software systems (e.g., patient notification may be done by placing telephone calls, sending electronic mails or speaking directly with patients). Because regulations do not presume the existence of machines and instead only describe stakeholder actions, it is imperative that lawyers and engineers consider all stakeholder actions in addition to the more obvious system constraints when developing compliant software. The decisions about which actions will be refined into functional requirements can then be based on the most relevant factors affecting long-term organizational goals at the time of system design. This approach is especially important as new and emerging technologies will enable more effective ways to achieve cost-effective and verifiable compliance in the future.

1.2.2 Phenomenology and Formalism

Jackson and Zave assert that the semantics of formal specifications adhere to an accepted phenomenology, in which the specifications express phenomena within the environment [94]. For example, the phenomenology of Deontic Logic consists of permissions and obligations, which distinguish "what is permissible" from "what ought to be" [84], whereas the phenomenology of Temporal Logic describes the order of events in linear or real-time. Several phenomenologies can be expressed in hybrid logics that are undecidable [20]. In legal reasoning, Hohfeld distinguishes fundamental legal concepts, including *rights* and *duties*, which are similar to permissions and obligations, and their respective opposites, which he calls *no-rights* and *privileges* [80, 81]. The Hohfeld opposites are similar to how permissions and obligations are negated under the axioms of Deontic Logic: not permissible implies it ought not to be, sometimes called a *prohibition* or *refrainment*, which agrees with the Hohfeld concepts. In Hohfeld, not obligatory is called a "privilege", which is distinct and separate from a "right" or permission. In requirements engineering, *anti-rights*, which are statements that do not confer a right, and *anti-obligations*, which, similar to privileges, are statements that do not confer an obligation [36].

Goal-oriented requirements engineering frameworks and methods are used to model the phenomenology of stakeholder *goals* that describe states or actions to be achieved, maintained or avoided within the environment. These include the Knowledge Acquisition in autOmated Spec-

ification (KAOS) framework [51], the Goal-Based Requirements Analysis Method (GBRAM) [5], i* (pronounced "ay-star") [166] and Tropos [67, 70]. Goals formally distinguish an actor (a noun) from an action (a verb) and can be classified by their action word: the first action words proposed by Dardenne et al. include *achieve, avoid, cease, maintain* and *optimize* [51]. These action words have a formal semantics in Typed, First-Order Temporal Logic [51]. The action words differ from legal concepts: whereas *avoid* and *cease* can be viewed as refrainments, each of these words, including *achieve* and *maintain*, can be permissible, obligatory or forbidden depending on the legal context. Others have since extended the goal taxonomy with additional action words from the privacy and security domain, including *allow, deny, inform, limit, monitor and require* [6]. Several of these action words (e.g., allow, require, deny) are closer approximations of Deontic concepts for permissions, obligations and refrainments. In general, the direct object of the action word in a goal description is an unstructured natural language phrase [4, 51, 67, 166].

In requirements engineering, the term refinement has been used to describe goal *decomposition*, in which goals are decomposed into sub-goals using logical AND/ OR relationships [51, 123]. In this dissertation, *refinement* means: (1) to extend a requirements model by adding new information, including the act of goal decomposition; or (2) to reduce an abstract concept in a model to a specialization of that concept (e.g., to reduce an abstract term "software application" to a specialization of that term, "web browser"). This broader definition of refinement describes a process that increases specificity and reduces ambiguity by removing unintended interpretations of requirements. Semantic Parameterization is a process that has been applied to goal phrases to formalize this second meaning of refinement in goal models using Description Logic [33]. To help focus the requirements engineering effort, a requirements model should allow engineers to restrict the scope of phenomenology to as few concerns as necessary to complete a domain-specific engineering task while providing the flexibility to select which concerns are appropriate for the task at hand.

1.2.3 Belief and Intensional and Extensional Knowledge

Terminology in requirements engineering should be grounded "in the reality of the environment for which a machine is to be built" [93, 167]. This requires separating domain knowledge into terminology and assertions about the domain. Zave and Jackson note that definitions or assumptions are stated in the *indicative* mood and describe knowledge about the domain; requirements are stated in the *optative* mood and describe how the environment ought to be [92, 167] (see Table 1.1). In knowledge engineering, knowledge about the domain and expressed in the indicative mood is also called *intensional knowledge*, whereas assertions about instances in the domain are called *extensional knowledge* [11]. Separating intensional from extensional knowledge allows engineers to separate the task of refining which classes of actors and objects are affected by regulatory rules from the task of describing which states and actions of software systems will satisfy these rules. Assertions about possibility and necessity, which include statements in the optative mood, comprise modal knowledge. In this dissertation, we distinguish between intensional and extensional knowledge and between upper and lower ontology in the analysis of legal knowledge described in

regulations. In addition, legal statements are codified using Deontic concepts, a form of modal knowledge that distinguishes what is "permissible" and from what "ought to be" [84].

Table 1.1: Categories of requirements engineering knowledge

Definition	**Mood**	**Category**	**KAOS Level**	**Ontology**
Domain-independent knowledge	Indicative	Intensional	Meta	Upper
Domain-dependent knowledge			Domain	Lower
Assertions about instances in the domain		Extensional	Instance	
Assertions about possibilities in the domain	Optative	Modal		

Goal-oriented requirements engineering is a popular approach to modeling the environment because goals broadly describe environmental states [4, 51, 67]. Dardenne et al. founded this approach by introducing KAOS [51]. KAOS defines three separate levels of knowledge: the *meta-level* includes domain-independent concepts such as agent, action and entity; the *domain-level* includes domain dependent concepts, which specialize meta-level concepts such as borrower, borrow and book; and the *instance-level* includes the instances of things in specific states or events [51]. Meta-level knowledge describes a phenomenology such as permissibility or temporality of actions without describing what those actions are in a given domain. Meta-level knowledge is often grouped with domain-level knowledge in an ontology (see Table 1.1). Ontologies are often viewed hierarchically, wherein the most general concepts (e.g., agent, action) appear above specialized concepts (e.g., borrower, borrow) and are connected by arrows representing specialization relationships. Accordingly, the meta-level concepts are called the *upper ontology* whereas the domain-level concepts are called the *lower ontology*. Ontologies do not include extensional knowledge or assertions about the domain. In requirements engineering, the upper ontology should be limited to primitives that are necessary to describe a specific phenomenology and the lower ontology should represent terminology that is grounded in the environment of the machine.

Statements in regulatory documents similar to the HIPAA Privacy Rule can be partitioned by whether they are stated in the indicative or optative mood [36]. In these documents, indicative statements are called *definitions* [36]. Among optative statements, three separate modalities are distinguished in this dissertation: (1) actions that are permitted are called *rights* or *permissions*; (2) actions that are strongly encouraged are called *recommendations*; and (3) actions that are required are called *obligations* [36]. Because rights are generally ascribed to people but not processes, the encapsulating term "permission" is used instead throughout this dissertation. Permissions and obligations are similar to the notions of "what is permissible" and "what ought to be" that are characterized by the Deontic Logic [84]. Recommendations have been observed more frequently in organizational security policies than in regulations [34]. Requirements engineering terminology distinguishes *mandatory* and *desirable* requirements [153, 69, 78, 85], which align with legal obligations

and recommendations, respectively. This dissertation introduces the new category of *discretionary* requirements that align with legal permissions.

1.2.4 Frame-based Knowledge Representation

Frames were proposed in the late 1960s and mid-70s as a linguistic and conceptual structure to model knowledge about the world [63, 119, 145]. In general, a *frame* corresponds to a concept that has one or more slots; each *slot* describes a stereotypical property of that concept. Slots are assigned an atomic value or another frame, called a *sub-frame*. Sometimes, slots have default values. In this dissertation, frames are extended by allowing slots to be assigned a first-order logic expression in which the propositions in these expressions are frames. This extension provides a rich frame-based language with which to express the variety of situations and contexts described by policies and regulations. Finally, frames correspond naturally to objects in object-oriented programming and patterns or templates in requirements engineering. A *pattern* is a frame with zero or more sub-frames and a *template* is a kind of pattern that is "shallow," wherein slots are only assigned atomic values; thus templates never contain sub-frames.

In requirements engineering, frames, patterns and templates have been employed to formalize constraints on requirements. Fillmore's *case frames* or *case roles* [63] and Gruber's *thematic roles* [77] model the properties of actions, such as the *actor* who performs the action or the *object* upon which the action is performed. Table 1.2 presents the correspondence among concepts in the upper ontology presented in Section 2.2, case roles and thematic roles. Case frames have been used to model scenarios [132] and goals [26, 108], the latter of which are limited to simple action statements. In addition to case frames, other approaches have employed patterns or templates to model Deontic [27, 26, 105] and Temporal constraints [44, 106, 152]. Koch et al. employ a template to model normative goals in a goal-refinement hiearchy [105]. Breaux et al. identified patterns to formalize permissions and obligations [26, 27] and proposed a template-based method [26] and a pattern-based method [30] to generate a controlled subset of natural language. The Frame-Based Requirements Analysis Method (FBRAM), presented herein, extends that work and is used to demarcate phrases in regulatory documents, in which some phrases represent concepts (frames) and other phrases link these concepts together as roles (slots). Konrad et al. [106] and Smith et al. [44, 152] employ patterns and templates, respectively, to align temporal constraints with natural language. Similar to Reubenstein et. al [143], FBRAM aligns the demarcated frames with a denotational semantics for generating expressions in first-order predicate logic. In Section 2.5.1, we describe a method based on this semantics for generating requirements specifications using a standard template.

1.2.5 Traceability between Artifacts

Domain descriptions exist as thoughts in the minds of stakeholders and as informal natural language transcriptions of those thoughts in the form of interviews, scenarios, policies and regulations, to

Table 1.2: Correspondence between upper ontology, case frames and thematic roles

Upper Ontology	**Case Role**	**Thematic Role**
subject	agent	agent
action		
object	objective	theme/ patient
source	source	source
target	dative	beneficiary/ recipient
purpose	goal	goal
instrument	instrumental	instrument
location	locative	location
condition		time
exception		time

name a few. Engineers must consistently maintain the alignment between informal descriptions and formal specifications, because the user of formal specifications must often recall the relevance and original context of formal expressions in the actual domain [94]. Gotel and Finklestein proposed contribution structures as a formalization of important relationships (e.g., authority, approval and commitment) between stakeholders and requirements [74, 75]. Because legal requirements are extracted from legal texts and not elicited from stakeholders, a different model is needed to maintain traceability between legal requirements and legal texts. The FBRAM maintains such traceability for multiple legal concepts described in Chapter 2; however, extensions are needed to formalize *jurisdiction*, which means the authority to interpret and apply the law to a specific context.

Goal-oriented frameworks such as KAOS [51] and Tropos [67] assume that a process exists to acquire goals from stakeholders. If the formal semantics of KAOS and Tropos cannot describe the full phenomenology of the domain, the goal models will miss information that is elicited from stakeholders. Alternatively, if the stakeholders relay descriptions of the environment through interviews, scenarios and policies, then requirements engineers can use the GBRAM to acquire goals from these sources [4, 7]. The Goal-Based Requirements Analysis Tool (GBRAT) was developed to support the GBRAM and maintains document-level traceability between goals and their source documents [8]. To date, the FBRAM presented in Chapter 2 represents the first attempt to rigorously maintain traceability while acquiring legal requirements from regulations at the statement and phrase-levels. Because slight or careless alterations in legal wording can result in severe legal penalties, a method similar to the FBRAM is needed to ensure due diligence and good faith when achieving compliance.

1.3 Related Work

Presently, there is no single method or theory that specifically targets legal requirements acquisition. In addition to describing the complexity of this problem, this dissertation also serves to evaluate the effectiveness of traditional practice in legal requirements acquisition. Therefore, related work

discussed in this section touches upon related challenges, including: a progression of methodology in requirements engineering to extract goals from natural language documents upon which this dissertation is based [4, 28, 36, 31]; two related approaches to acquire formal specifications from regulations [53, 113]; and two tool-supported methods to provide rulemaking and enforcement assistance [158, 159] and compliance assistance [100, 101, 102]. Finally, we conclude with a review of controlled languages, which are used to map between formal models and natural language in requirements engineering [26, 27, 106, 52, 152], databases [41] and artificial intelligence [47, 98], standard lexicons that are used to manage requirements terminology [6, 7, 50, 97, 134, 160, 161] and compliance standards [79, 88, 91, 90].

1.3.1 Goal-oriented Requirements Engineering

Antón introduced the Goal-Based Requirements Analysis Method (GBRAM) to extract goals from traditional requirements sources (e.g., interviews, scenarios) [4, 5] and later privacy policies [7]. GBRAM provides engineers with several heuristics to: identify goals, stakeholders, agents and constraints; classify and refine goals; eliminate redundancies; and reconcile similar goals [4]. While the GBRAM was ground-breaking, it did not require engineers to maintain traceability between goals and the written sources of knowledge at the statement- or phrase-levels. This is usually not problematic if the knowledge sources are relatively small (e.g., a few pages) and the goal statements are largely dissimilar. In addition, because the GBRAM was designed for traditional requirements sources, the GBRAM heuristics do not account for the full range of phenomenology in policies and regulations, including conditions, exceptions and modalities that indicate permissions and obligations. In general, however, these limitations can yield goal models with missing and unintended interpretations that deviate from the originating legal text. Without extensions to the GBRAM, engineers should have multiple, independent experts review the resulting goal statements to identify these potential problems.

The GBRAM heuristics build upon the Inquiry-Cycle Model proposed by Potts et al. in which a requirements engineer asks *what, who, where, how, why* and *when* questions based upon information contained in a statement or phrase to identify new requirements [139]. Breaux et al. identified several natural language patterns [26, 27, 28] to formalize the *what, who, where, how* and *why* questions in Description Logic [33]. These patterns improve refinement in goal models by mapping goal phrases to formal predicates that are used to reason about goal properties. Dardenne et al. formalized several *when* questions through specific action words in Typed, First-Order Temporal Logic [51] and Konrad and Cheung identified patterns to formalize several *when* questions in Real-Time Temporal Logic [106]. GBRAM has been extended with new heuristics to identify rights, obligations and constraints in policies and regulations in a pilot study [28] and three case studies [31, 32, 36]. Rights and obligations are goals that describe which actions a stakeholder is permitted or required to perform, respectively. The FBRAM incorporates new heuristics for encoding legal concepts that are not addressed by these other methods and techniques. In addition, the FBRAM provides a new, tool-supported model for maintaining traceability at the statement- and phrase-

levels to make the construction of requirements models more rigorous, systematic and consistent with the legal text.

1.3.2 Extracting Models from Regulations

Early efforts to model legislation and statutory law identified important challenges for the analysis of U.S. federal regulatory law. Many researchers agree that a systematic methodology is needed to extract such models [2, 15, 18, 147, 149, 154]. For example, Sergot et al. adopted "short term, relatively ad hoc solutions to problems of knowledge representation" [15] and still claimed to "identify the intended interpretation [of indecision and ambiguity in the law] with little difficulty" [147]. However, there is disagreement about whether such models should be constructed using a first-order logic system such as Prolog [18, 147] or by using a domain-specific language with operators tailored for regulatory semantics [15, 114, 154]. Sergot et al. prefer the former approach for the practicality of engineering a system [147]; however, Stamper correctly observes that a domain-specific language provides the best "economy of expression" [154], which is a widely accepted viewpoint among practitioners [117]. Bench-Capon et al. claim domain-specific languages, which he refers to as deep models with normative semantics (e.g., permissons, obligations, etc.), are needed to express regulations that are not yet grounded in organizational practices [14]; a focus of this dissertation. The FBRAM combines a domain-specific language, represented by a context-free grammar, that overlays a legal text to create a first-order logic model of predicates from the text. The model is based on a normative semantics and enables tool-supported analysis of legal requirements to identify several ambiguities.

Finally, early work also identified the following additional challenges that are addressed by the FBRAM in this dissertation: legal models should match the language and structure of legal documents [15, 147, 149] and normalize expressions of the logical structure [2, 18]; models should map between logical propositions and corresponding paragraphs in the law [147, 149]; models should syntactically and semantically explicate ambiguity [18]; and models should encode cross-references and support their use in resolving exceptions, also called *counterfactuals* [15, 147]. These works contribute to a foundation of related work that has appeared over the past several decades in the proceedings of the International Conference on Artificial Intelligence and Law. Among more recent works, for example, Martinek and Cybulka formalize so-called "meta-provisions" for amending and repealing laws and their consequences using the Event Calculus [112]; a broader legal requirements management issue not addressed in this dissertation.

Several recent efforts to model regulations in first-order logic have made little progress over prior work [53, 100, 113]. Efforts that focus on normative statements, however, have introduced new theory for reasoning about delegations [36, 70] and balancing rights with obligations [36]. In addition, recent contributions toward a rigorous and systematic method for acquiring models from law include: ontologies that encode legal language and concepts [3, 36, 30, 107]; document models that represent the structure of the law [30, 159, 102]; and constraint normalization using Boolean logic [36, 159, 102]. For example, Allen and Saxon present the A-HOHFELD language for

expressing Hohfeld concepts and performing legal reasoning [3]. Breaux and Antón [30, 36] and Giorgini et al. [70] classify legal statements using normative concepts, whereas other approaches are limited to identifying terms-of-art and constraints [159, 102, 107]. Breaux et al. also identify several patterns to balance rights and obligations through logical inference [36]. This work extends two related efforts to check consistency between permissions, obligations and refrainments by Cholvy [42] and to show how permissions presuppose, and thus create exceptions to, implied obligations by Boella and van der Torre [21]. The methods by Kerrigan and Lau [101, 102] and van Engers et al. [158, 159] rely heavily on automated tools with known limitations, avoiding complex irregularities that must be addressed by a rigorous method. To varying degrees, the FBRAM presented herein addresses many of the problems that van Engers et al. and Lee et al. identify and that Kerrigan et al., May et al. and Dinesh et al. do not address, including the formalization of definitions and normative statements [159, 101] and techniques to manage verbosity, ambiguity, polysemy, and redundancy [53, 101, 107, 113] and cross-references [53, 113].

Finally, researchers have sought to automate extracting models from natural language text [73] and regulations [17, 158, 102, 103]. Existing automation includes identifying common abstractions in text [73], inferring the document model from the legal text [102, 103], identifying inconsistencies between the document model and cross-references [158], and identifying instances of legal concepts based on pre-defined heuristics, including normative statements (e.g., permissions, obligations, etc.) [17, 104, 103] and terms-of-art [158, 102, 103]. Kiyavitskaya et al. have gone further by combining document model identification with knowledge of normative statements to perform limited case-splitting across continuations or nested sub-paragraphs [103]. As these techniques continue to mature, they stand to reduce the manual effort of formalizing regulations as legal requirements in a comprehensive and rigorous methodology such as the FBRAM. At present, the FBRAM provides a roadmap for important challenges that existing and future tools must address to acquire legal requirements from laws and regulations.

1.3.3 Policy Languages, Models and Methods

Policy languages, models and methods have been proposed to formalize policies that describe and control the behavior of software systems. Policies are similar to requirements because they describe the environment of a machine [167]. Moffett and Sloman proposed the theory of specifying policies as system objects [121] and later founded the IEEE-sponsored Workshop on Policies for Distributed Systems and Networks. Policy languages typically describe an event-action-condition cycle, wherein an event causes an action to be taken, only if a pre-condition is satisfied [111, 110]. Policies exist to achieve organizational goals, thus policy languages and methods have sought to support goal refinement into implementable policy actions [12, 111]. Similar to regulations, policy languages and models represent the complex interactions between multiple agents. These interactions have been expressed using a normative semantics, including permissions, obligations and refrainments [96, 111, 120, 135]. To avoid conflicts, policy languages can employ a policy hierarchy [111], which is similar to the use of exceptions to yield priorities among legal requirements; an approach introduced

in this dissertation. Furthermore, several web-based policy languages include special operators to express: speech acts between agents (e.g., delegate, revoke, request, cancel) [95, 96]; privacy policies, such as data collection and use practices [46]; and access control rules [122]. These techniques are domain-specific and address phenomena that do not appear in all U.S. federal regulations. Unlike policy languages, which are intended to reduce system administrative effort, the FBRAM provides engineers a methodology to navigate and acquire industry-wide legal requirements from laws that potentially govern multiple system specifications and designs.

1.3.4 Controlled Languages, Lexicons and Dictionaries

Controlled languages, which comprise a subset of natural language, have been developed in requirements engineering [26, 27, 52, 73, 106, 152], databases [41] and artificial intelligence [98, 47] to reduce ambiguity and inconsistency in natural language specifications. Goldin and Berry employ the AbstFinder tool to identify common abstractions in natural language corpora using signal processing methods [73]. Although each abstraction, consisting of a set of related words, is purely syntactic, the set coincides with multiple word senses that can be used to standardize requirement statements by ensuring that known word senses are explicated upon. Smith et al. describe the PROPEL tool that uses disciplined natural language (DNL) templates to capture requirements [152]. The templates permit a limited number of concise, highly structured phrases that correspond to formal properties in finite state automata. Konrad and Cheng employ a structured English grammar with special operators tailored to the specification of real-time properties [106]. Denger et al. use natural language patterns to capture conditional, temporal and functional requirements statements [52]. Abstractions, templates, structured grammars and patterns require the engineer to focus the domain description in a manner consistent with pre-defined concepts and operators in a formal method. Similarly, the FBRAM employs natural language patterns to identify key concepts in legal text (see Section 2.3) and to help engineers normalize descriptions of legal requirements (see Section 2.5.1). In addition, the FBRAM markup language and upper ontology (see Section 2) comprise a controlled language model that is used to structure legal requirements in a consistent manner.

In artificial intelligence, four approaches exist to map a subset of the English language to entity-relationship (ER) models [41] and Description Logic, including Semantic Parameterization by Breaux et al. [33], Attempto Controlled English (ACE) by Kaljarund et al. [98] and Computer-Processable ENGlish (PENG) by Cregan et al. [47]. Peter Chen proposed eleven rules to manually extract entities, relations and attributes in ER models from English sentences [41]. Semantic Parameterization is a process for modeling domain descriptions using Description Logic that has been applied to goal and regulatory statements [33]. ACE [98] and PENG [47] include two approaches to align a subset of natural language (English) with a formal semantics in DL. An important issue that arises during this alignment includes words with an anaphoric or cataphoric function, such as English pronouns and definite articles (e.g., this, that, the), that refer the reader to a particular thing or individual described in a prior or subsequent context, respectively. The FBRAM provides

operators for managing traceability as anaphoric and cataphoric ambiguities are resolved.

In requirements engineering, it is common practice to standardize the natural language vocabulary using a lexicon or dictionary. Antón et al. applied the Goal-Based Requirements Analysis Method (GBRAM) [4] to policies to extract goals that begin with a verb followed by a goal phrase [6, 7]. In GBRAM, these verbs are standardized in a shared lexicon to avoid redundant goals. Overmyer et al. describe the Linguistic Assistant for Domain Analysis (LIDA) tool that maintains a list of words acquired from natural language documents; the words are used to identify classes and attributes to be expressed the Unified Modeling Language (UML) [134]. Similarly, Kaindl shows how to identify binary relationships between nouns in natural language definitions and map them to new classes in the UML [97]. Goldin and Berry introduced AbstFinder, a tool for identifying abstractions using signal processing techniques [72]. Wasson et al. employ a domain map to facilitate effective communication between domain experts and engineers [160, 161]. The domain map contains technical words classified into hierarchies of superordinate and sub-ordinate terms and is used to identify ambiguous terms based on their commonality and domain-specific interpretation. Cysneiros and Leite model non-functional, natural language requirements in the UML using class, sequence and collaboration diagrams [50]. Their approach uses a Language Extended Lexicon (LEL) to codify the natural language vocabulary in terms of denotations and connotations. The FBRAM employs phrase heuristics to identify concepts in legal documents that map to case roles using an upper ontology. The FBRAM also maps terms from definitions into a specialization hierarchy or lower ontology. Together, these phrase heuristics and lower ontology comprise a formal lexicon to improve legal requirements acquisition.

1.3.5 Management and Technical Compliance Standards

Compliance management and technical standards provide organizations broad guidance in how to plan, implement, and monitor security controls in information systems. For example, the Control Objectives for Information and related Technology (COBIT) [88] and ISO/IEC 17799:2005, subtitled "Code of practice for information security management" [91], are two popular management standards that focus on personnel, process and resource planning and management. Alternatively, ISO/IEC 15408:2005, subtitled "Evaluation criteria for IT security" and commonly called the Common Criteria, is a technical standard that contains functional security requirements [90]. In addition, the U.S. National Institute of Standards and Technology maintains the Special Publications Series 800, which include numerous technical standards that target specific security controls and computer systems. For example, the Special Publication 800-66 provides a resource guide for implementing the HIPAA Security Rule, which was developed because no single publication addressed compliance with the Security Rule at the time [79]. While companies are implementing ISO/IEC 15408 to support their HIPAA compliance posture [116], a study by the SANS Institute found HIPAA requirements that are not contained in ISO/IEC 17799, including requirements for preventing, monitoring and terminating access to patient health information [22]. Moreover, the "stakeholder focus" of the HIPAA Privacy Rule requires a detailed review of business infor-

mation practices that are not specifically addressed by these technical and management security standards. Individuals and organizations that seek to comply with U.S. federal regulations should first determine their rights and obligations under the law before deciding if compliance with an existing standard covers the full scope and intent of that law. The FBRAM facilitates itemizing and reviewing legal requirements to make this determination systematically.

1.4 Overview of Remaining Chapters

The remainder of this dissertation is organized as follows: Chapter 2 presents the abstract model, a grounded theory, that describes the Frame-Based Requirements Analysis Method (FBRAM); Chapter 3 presents the research methodology used to discover and empirically validate the abstract model, including the exploratory, multi-case study and human subject experimental designs; Chapter 4 presents the results from the multi-case study; Chapter 5 presents the results from the human subject experiment; and Chapter 6 describes the limitations of the FBRAM, including threats to validity, and a summary of the contributions and future work. Important terminology is defined throughout this dissertation with definitions referenced in the Index.

Chapter 2

Abstract Model

The whole of science is nothing more than a refinement of everyday thinking.
Albert Einstein (1879–1955)

The Frame-Based Requirements Analysis Method (FBRAM) provides engineers a means to extract formal requirements from U.S. federal regulatory documents. To demonstrate due diligence and good faith, the FBRAM ensures that engineers trace relevant words and phrases from their exact position in the legal text to an unambiguous role in a formal requirement specification. Furthermore, FBRAM enables engineers to formalize their interpretations by resolving ambiguities that result from certain English conjunctions (and, or), natural language context sensitivity and cross-references. The FBRAM abstract model is comprised of the following four method artifacts:

- A *document model*, discussed in Section 2.1, which represents the regulatory document structure (e.g., parts, sections, paragraphs, etc.) and coordinates this structure with cross-references and frames identified by engineers;
- A reusable, domain-independent *upper ontology* that includes statement- and phrase-level concepts, discussed in Section 2.2;
- Natural language *phrase heuristics* that map concepts in the upper ontology to informal definitions and natural language patterns, discussed in Section 2.3; and
- A frame-based *markup language*, formalized by a context-free grammar and discussed in Section 2.4, that engineers apply to maintain traceability from the legal text to concepts in a lower ontology and formal assertions.

These artifacts are employed in the FBRAM process model that appears in Figure 2.1, in which method artifacts, process inputs and process outputs appear in square-cornered boxes and procedures appear in round-cornered boxes. Numbers in black circles represent the linear-temporal progression of the following composite procedures: (1) apply the document model; (2) apply the markup language; and (3) parse the annotated regulation. The process is further divided into

three columns: the *inputs/ outputs*, consisting of the inputs to, and outputs from, each numbered procedure; the *method procedures* that transform the inputs into the outputs; and the *method artifacts* that are reused to perform those procedures. The final legal requirements produced by the FBRAM can be analyzed to reason about compliance decisions in a relevant organizational context. Several techniques for performing this analysis are discussed in Section 2.5.

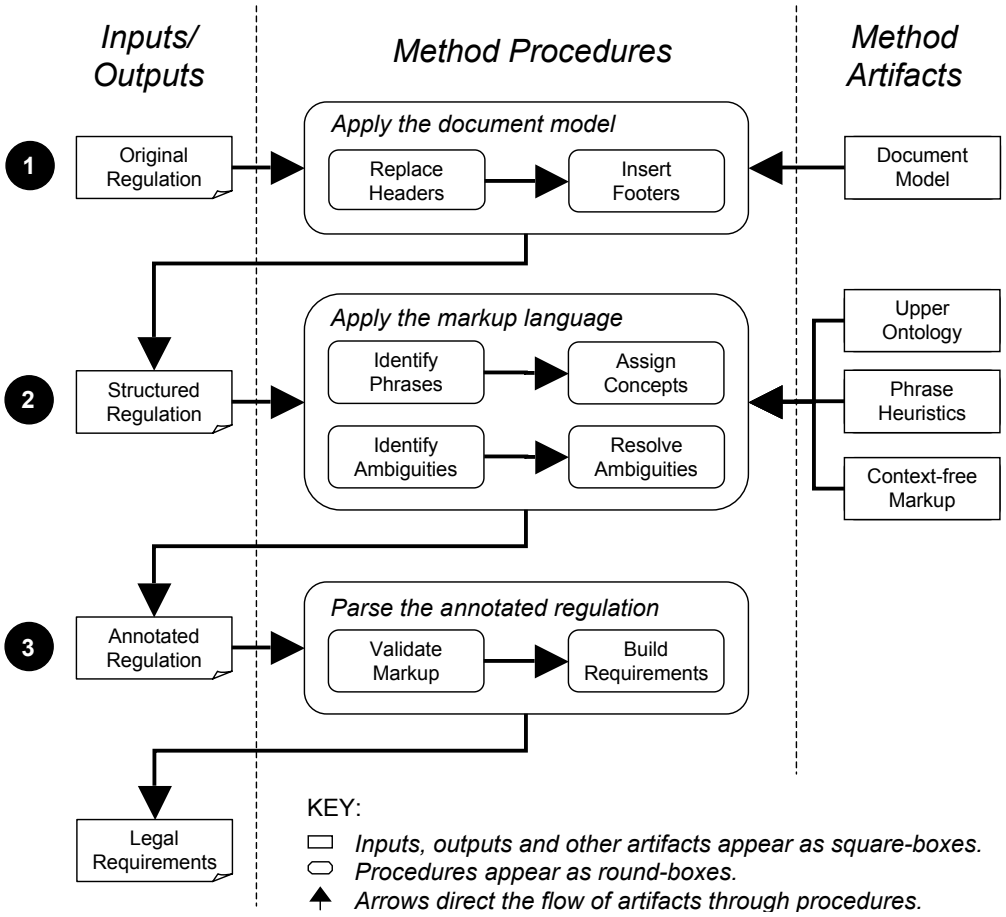

Figure 2.1: Frame-Based Requirements Analysis Method process model

The entire process is discussed in the following subsections by introducing the method artifacts and describing their respective role in facilitating the method procedures. The following discussion uses a running example from HIPAA Privacy Rule §164.520(a)(2)(ii) that describes a group health plan's obligation to maintain and provide a privacy notice to patients.

2.1 Regulatory Document Model

The document model enables forward and reverse-traceability between requirements and the indices of divisions in the legal text, which contain the originating phrases and statements for these requirements. Divisions have optional headers consisting of numbered or lettered indices and titles

and each division may contain nested sub-divisions. Divisions are used to formally represent parts, sections, paragraphs and sub-paragraphs in U.S. federal regulations as specified by the drafting guidelines of the U.S. National Archives [131]. The document model is used to resolve cross-references that appear in exclusion statements or exception phrases, which can be formalized as priorities between requirements [31]. In addition, statements may begin in one paragraph and end in a sub-paragraph, called a *continuation*. Continuations often coordinate a set of shared constraints (e.g., subjects, actions, conditions, etc.) in the leading paragraph with alternative permissions, obligations and refrainments in sub-paragraphs. Consider the division syntax in HIPAA Privacy Rule excerpt §164.520(a)(2)(ii), below; it describes two obligations to notify patients of their privacy practices and shares the same subject constraint (a group health plan) for these two obligations. The sub-paragraph (a)(2)(ii) is a sub-division with the index "(ii)" and contains two sub-divisions, indexed "(A)" and "(B)"; these three divisions to do not have titles. However, the encapsulating paragraph (a)(2) is titled "Exception for group health plans.", which is a brief phrase describing the content of the encapsulating paragraph.

> (2) Exception for group health plans.
>
> ...
>
> (ii) A group health plan must:
>
> (A) Maintain a notice under this section; and
>
> (B) Provide such notice to any person

To support traceability, the document model formalizes divisions in the legal text. The document model semantics are expressed in the W3C eXtensible Schema Language (XSL) [19, 155] and instances are encoded in the W3C eXtensible Markup Language (XML) [23]. The engineer applies the model to the legal text by replacing division headers with an opening XML `<div>` tag that represents the start of a division and replaces the header index and title, if any, with corresponding attributes `index` and `title` in the opening tag; the engineer adds the closing XML `</div>` tag at end of the division. The XML Schema that formalizes the document model appears in Appendix C. The above excerpt appears in Figure 2.2, with the document modal annotation appearing in a `colored highlight`.

Cross-references that are strings in a regular language [150], such as certain citations in the Code of Federal Regulations, can be parsed to automatically index divisions in the document model. For example, the citation §164.520(a)(2)(ii)(B) can be parsed to obtain a comma-separated path "164,520,(a),(2),(ii),(B)" that can be used to iteratively traverse sub-divisions in the document model, starting with Part 164 and finishing at sub-paragraph (B) where the phrase "Provide such notice to any person" is obtained. The frame-based markup described in Section 2.4 further narrows this division-level traceability to individual words and phrases that are acquired to formulate requirements specifications.

```
1    <?xml version="1.0"?>
2    <document>
3    ...
4    <div index="(2)" title="Exception for group health plans.">
5        ...
6        <div index="(ii)">
7            A group health plan..., must:
8            <div index="(A)">
9                Maintain a notice under this section; and
10           </div>
11           <div index="(B)">
12              Provide such notice to any person
13           </div>
14           ...
15           </div><!-- end of (2)(ii) -->
16       </div><!-- end of (2) -->
17   </document>
```

Figure 2.2: Example instance of the document model

Because the legal text's indentation and font styles may be lost or corrupted when the text is transferred to a plain text format, the engineer must manually apply the document model to the plain text. After this application, however, existing tool support restores the indentation and some font styles automatically by parsing and presenting the document model instance in a graphical user interface.

2.2 Reusable, Domain-independent Upper Ontology

Whereas the document model from the previous section is used to formalize the legal syntax, the upper ontology described in this section is used to formalize legal semantics. The reusable, domain-independent upper ontology is a semantic model that assigns meaning to frames, which consist of phrases and sentences that are identified by an engineer in a legal text. This meaning is used to consistently map frames to functions for performing different types of requirements analysis, discussed further in Section 2.5. The upper ontology that appears in Figure 2.3, which is represented using the Unified Modeling Language (UML) [76], has been developed across a succession of prior pilot studies [26, 27, 28] and case studies [31, 36] and describes sentence and phrase-level concepts in legal text. The sentence-level concepts are defined as follows:

Def.S.1: *Permission* – an act that an actor is permitted to perform.

Def.S.2: *Obligation* – an act that an actor is required to perform.

Def.S.3: *Refrainment* – an act that an actor is prohibited from performing.

Def.S.4: *Exclusion* – an act that an actor has no express permission to perform or that an actor is not expressly required or prohibited from performing.

Def.S.5: *Fact* – an act or state of being that is conditionally true.

Def.S.6: *Definition* – a statement of the meaning of a word or phrase.

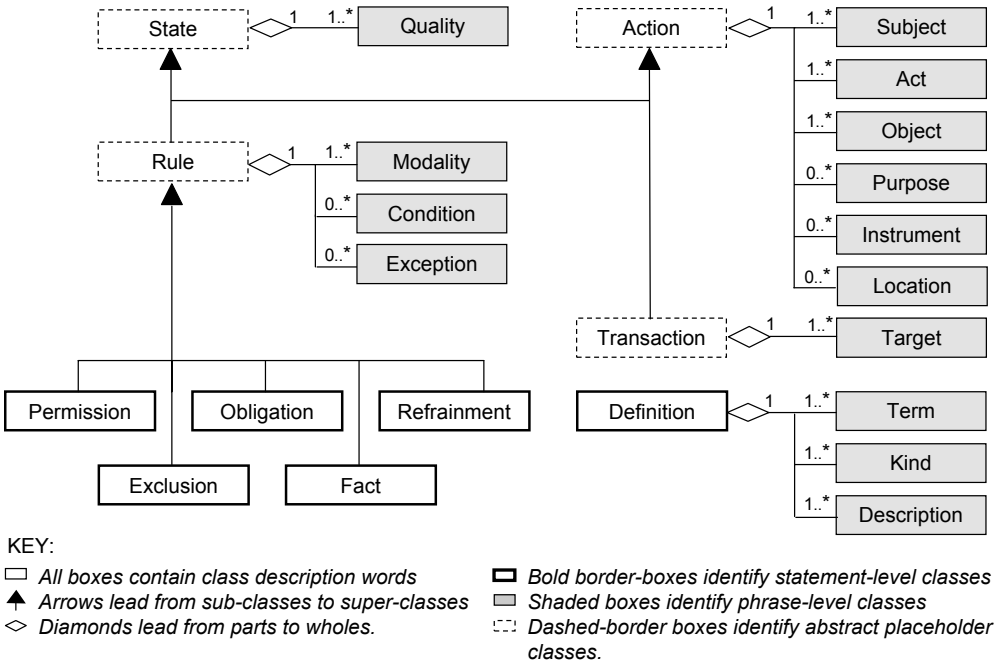

Figure 2.3: Reusable, domain-independent upper ontology

The statement-level concepts permission, obligation, refrainment and exclusion correspond to the Hohfeld concepts right, duty, no-right and privilege, respectively [80]. Sentences that describe acts will have properties assigned to one or more of the following phrase-level concepts:

Def.P.1: *Subject* – the actor who performs an action.

Def.P.2: *Act* – the act performed by the subject.

Def.P.3: *Object* – the object on which the action is performed.

Def.P.4: *Target* – to where/ with whom an action is performed by the subject.

Def.P.5: *Source* – from where an action is performed by the subject.

Def.P.6: *Purpose* – an act describing why an action is performed.

Def.P.7: *Instrument* – an act or thing describing how an action is performed.

Def.P.8: *Location* – the place where an action is performed.

Def.P.9: *Quality* – a property of a state of being that exists at some interval.

Def.P.10: *Modality* – the modality of the action (e.g., may, must, etc.)

Def.P.11: *Condition* – an event that occurs before, during or after executing a rule.

Def.P.12: *Exception* – an event that does not occur before, during or after executing a rule.

Several phrase-level concepts correspond to questions that requirements engineers frequently ask in the Inquiry-Cycle Model introduced by Potts et al [139]. Table 2.1 presents a mapping from the phrase-level concepts to questions in the inquiry cycle model [33].

Table 2.1: Correspondence between phrase-level concepts and Inquiry-Cycle Model

Concept	**Inquiry-Cycle Model Question**
subject	*Who* performs the action?
object	Upon *what* is the action performed?
target	With *whom* is the transaction performed?
source	From *where* is the action performed?
purpose	*Why* is the action performed?
instrument	*How* is the action performed?
condition	*When* is the action performed?
exception	*When* is the action not performed?

Sentences that describe definitions will have properties assigned to one or more of the following phrase-level concepts:

Def.D.1: *Term* – the word which the definition defines.

Def.D.2: *Description* – an equivalent description of the defined term.

Def.D.3: *Kind* – a word that describes one kind, sub-class or sub-type of the defined term.

Each of the above concepts in the upper ontology is assigned a *concept code* that is used by the engineer in the frame-based markup to link phrases and statements from the legal text with these concepts, a topic we discuss in Section 2.3.

2.3 Natural Language Phrase Heuristics

To assist the engineer with assigning concepts from the upper ontology to statements and phrases in the legal text, the engineer is provided with several phrase heuristics. A phrase heuristic for identifying instances of concepts consists of a single concept in the upper ontology and a natural language phrase that can be automatically matched with phrases in the legal text by using a simple keyword search. Table 2.2 presents several such phrase heuristics that were identified in HIPAA §164.520-164.524. The patterns that appear with an asterisk (*) match statements that also describe *delegations*, a kind of transaction in which an actor delegates a permission, obligation or refrainment to another actor. Engineers must compare instances of these phrases in the legal text to the intended meaning that is described by the corresponding concepts in the upper ontology, because some phrases are ambiguous. For example, the phrase “may” can mean that an act is

possible but not necessarily permissible in the future, which conflicts with the intended meaning expressed in the definition of the permission concept in Section 2.2.

Table 2.2: Commonly used phrase heuristics for identifying slot concepts

Phrase	**Type**	**Phrase**	**Type**
if	Condition	may not require*	Obligation
when	Condition	must	Obligation
whenever	Condition	must deny*	Obligation
except as provided by	Exception	must permit*	Obligation
except when	Exception	must request*	Obligation
except that	Exception	has a right to	Permission
is not effective under	Exception	may	Permission
except as otherwise provided	Exception	may deny*	Permission
is excepted from	Exception	may require*	Permission
is not required to	Exclusion	retains the right to	Permission
may not	Obligation	does not have a right to	Refrainment

International standards [87] and best practices [78, 85, 153] recommend that requirements engineers use the modal verbs "shall" and "should" for specifying mandatory and desirable requirements, respectively. These modal verbs describe the "compliance level" [78, 85] or "degree of necessity" [87] for a requirement and have a similar meaning in U.S. federal regulations [131]. Regulations introduce a third type of requirement that is stated using the modal verb "may" and aligns with *discretionary requirements*, which organizations may choose to ignore or implement at their discretion [131].

2.4 Frame-based Markup Language

The frame-based markup language has a context-free grammar that can characterize different interpretations of a single legal text. By applying the markup language to the legal text, an engineer removes ambiguity that coincides with interpretations that are undesirable or potentially misleading. To apply the markup, engineers must align concepts from the upper ontology with legal statements and phrases and remove logical, attributive and referential ambiguities described in Section 1.1.3 using special operators in the markup language. The complete, unambiguous context-free grammar for the markup language appears in Appendix B. Table 2.3 presents the concept codes that are used in the markup below to align sentences and phrases with concepts in the upper ontology.

The running example from HIPAA Privacy Rule §164.520(a)(2)(ii) appears in Figure 2.4 with the markup in `colored highlight`; the document model tags are excluded from this example for easier reading.

While the example in Figure 2.4 may be difficult to read, existing tool-support includes an editor with syntax highlighting that eases reading by distinguishing the markup language's operators from the legal text. The markup is used to structure legal text into two types of nested blocks denoted

Table 2.3: Slot concept codes used in subsequent examples

Code	Concept	Code	Concept
a	Act	o	Object
c	Condition	O	Obligation
F	Fact	s	Subject
i	Instrument	t	Target

```
1   (ii) {#O [#s/1 A group health plan [that provides
2        health benefits solely through an insurance
3        contract with a health insurance issuer or HMO,
4        & and that creates or receives [protected
5        health information in addition to  [∼summary
6        health information as defined in §164.504(a)]
7        | or information on whether the individual
8        [is participating in the group health plan,
9        | or [is enrolled in | or has disenrolled from]
10       a health insurance issuer or HMO offered by
11       the plan]]]], [#m must]: {
12       (A) [#a Maintain] [#o/2 a notice] {#i under
13           [∼this section]}; | and
14       (B) [Provide] [#o*2 such notice] {#c upon
15           [request]} {#t to [any person]}}}. {#F [#s∼
16           The provisions of paragraph (c)(1) of this
17           section] [#m do not] [#a apply]} {#o to
18           [*1 such group health plan]}}
```

Figure 2.4: Example application of the markup language to legal text

by opening and closing brackets. *Pattern blocks*, denoted by curly "{ }" brackets, indicate the start and end of a requirements natural language pattern; these patterns are mapped to phrase heuristics to check consistency throughout the markup application. *Value blocks*, denoted by square "[ ]" brackets, indicate spans of text that are mapped to slot values by the parser. In addition, a block is typed if the opening bracket is followed by a number sign "#" and the concept code consisting of a sequence of letters (see lines 1, 11, 12, etc.) Within a pattern or value block, the English conjunctions "and" and "or" are mapped to logical connectives using the ampersand "&" and vertical-bar "|" operators for logical-and and logical-or, respectively (see lines 4, 7, 9, 13). Mapping English conjunctions to logical connectives will resolve logical ambiguities present in the legal text.

Recall from Section 1.1.3 that legal text contains attributive and referential ambiguities, which coincide with context-sensitive phrases that can be attributed to multiple nouns or pronouns that serve a anaphoric (backward-referencing) or cataphoric (forward-referencing) function. To resolve attributive and referential context-sensitive ambiguities, the engineer uses a virtual clipboard that

servers to temporarily store and retrieve phrases, which appear in the legal text, to and from numbered clipboard locations. The clipboard operations consist of the copy "/" operator, cut "\" operator or "∗" paste operator followed by the numbered clipboard location where the legal phrase is to be stored or retrieved by the parser. Recall that referential ambiguity includes words that have an anaphoric or cataphoric function. The phrases "such notice" (line 14) and "such group health plan" (line 18) constitute referential ambiguities that serve a cataphoric function, because they refer to the group health plan described previously on line 1. If the paste operator is applied to a block that contains text, as is the case in these two phrases on lines 14 and 18, the text in the block will be replaced by the pasted text that is stored at the corresponding clipboard location.

Cross-references are another source of referential ambiguity. To resolve cross-references, the engineer encapsulates the complete cross-reference phrase in a value block and inserts the cross-reference "∼" operator after the opening bracket of this block. For example, the phrase "summary health information as defined in §164.504(a)" (line 6) indicates that this rule is refined by the definition of summary health information in §164.504(a). The parser matches the phrase against a regular expression, which identifies the division cited by "§164.504(a)", and creates the relation between this rule and candidate statements in the corresponding division identified by Part 164, Section 504, paragraph (a). The engineer must then decide which statements from among the list of candidates are valid interpretations for this cross-reference.

The parser detects syntax and semantic errors, such as missing brackets, cycles that occur in the copy, cut and paste operations, unknown concept codes, unidentified cross-references, etc., and alerts the engineer who must then resolve these errors. Existing tool support includes an editor that parses the document model and provides syntax highlighting, which allows engineers to identify many of these errors during the markup application procedure.

2.5 Frame-based Requirements Analysis

Frame-based requirements are used to perform tool-supported analysis that can help engineers focus the requirements engineering effort on the needs of their organization's business practices with respect to relevant laws. In this section, existing tool-supported techniques that were developed to support this dissertation are discussed that: (1) generate formal requirements specifications from frames using a standard template; (2) construct a lower ontology from definitions; (3) balance stakeholder rights with obligations; and (4) and construct a priority hierarchy from exceptions between rules.

2.5.1 Presenting Requirements Using Standard Templates

The frame-based markup has a denotational semantics that is mapped to a formal requirements specification using a standard template (see Figure 2.5). These templates represent "shallow" frames that are comprised of a statement-level concept (e.g., permission, obligation, etc.) and a set of phrase-level properties that exhaustively describe the original regulatory statements, which

were framed by the markup. Unlike the legal text, these templates employ several requirements engineering "best practices" that support easier reading and analysis, including the abilities: to uniquely identify individual requirement statements [85, 153]; to separate and list attributes of a requirement [85, 153]; and to decompose requirement statements into subclauses [85].

After the markup has been applied and parsed, existing tool support serializes the frame-based requirements using XML and transforms the serialized frames into the eXtensible HyperText Markup Language (XHTML) [136] using an eXtensible Schema Language Transformation (XSLT) [109]. In XHTML, the requirements are presented in a table format. Parsing the example markup from Figure 2.4 yields two requirements (to maintain notice and to provide notice); the second requirement is presented in Figure 2.5 using the same table format that is used in practice throughout the case studies described in Section 3.1.3.

Frame:	Obligation
Pattern:	[*subject*] [modality] [*act*] [*object*] {upon [*condition*]} {to [*target*]}
Trace:	ID 5, Line 1:0, Source: 164.520(a)(2)(i)(B)(ii)
Slots	**Values**
condition	*upon…* request
subject	A group health plan that provides health benefits solely through an insurance contract with a health insurance issuer or HMO *A group health plan that creates or receives* protected health information in addition to summary health information as defined in §164.504(a) *A group health plan that creates or receives* information on whether the individual is participating in the group health plan, or is enrolled in or has dis-enrolled from a health insurance issuer or HMO offered by the plan
modality	must
act	Provide
object	a notice
target	*to…* any person

Figure 2.5: Example instance of the frame-based requirement template

The example in Figure 2.5 begins with a header that consists of the statement frame type (Frame), the requirement natural language pattern (Pattern) and the traceability information (Trace) with the requirement ID, the line number, the line index and the corresponding paragraph number in the legal text. After the header, each slot is listed with the slot concept (a phrase-level concept from the upper ontology) and the slot value. Because the slot values may be expressed using logical connectives (e.g. see the subject slot value in Figure 2.5), the values are presented in a tree-format comprised of logical-and branches (solid lines) and logical-or branches (dotted lines).

Legal requirements often share similar requirements natural language patterns. Table 2.4 shows

the most frequently occuring patterns that were derived from HIPAA §164.520-164.526 using the frame-based markup language. Hull et al. recommend using similar patterns, called boilerplates, to specify requirements [85]. The patterns in Table 2.4 illustrate two important observations. First, certain roles (e.g., instrument, purpose, condition, duration) serve to elaborate some requirements, but are not required to state all requirements, as observed in patterns 1, 2 and 3. Second, the order of certain roles is arbitrary when requirements are extracted from legal documents. For example, patterns 3, 4 and 5 illustrate how the `condition` role can appear prefixed, postfixed or inserted into the requirement statement.

Table 2.4: Commonly used requirements natural language patterns

Rank	Count	Pattern
1	35	[subject] [modality] [act] [object]
2	6	[subject] [modality] [act] [object] [instrument]
3	5	{if [condition]} [subject] [modality] [act] [object]
4	5	[subject] [modality] [act] [object] {if [condition]}
5	5	[subject] [modality] [act] [object] [condition] {to [target]}

2.5.2 Sorting Requirements in a Lower Ontology

In legal text, a *term-of-art* is defined as a word or phrase having a specific, precise meaning in a given specialty, apart from its general meaning in ordinary contexts [68]. In many cases, terms-of-art are presented in a separate Definitions section, as specified by the drafting guidelines of the U.S. National Archives [131]. In addition to definitions, it is common for regulations to provide other words that serve to illustrate different *kinds* of such terms. These kinds can be viewed as specializations in meaning and organized in a lower ontology. The frame-based markup provides engineers a means to identify terms-of-art and link them to kinds and definitions that appear within a legal text. Figure 2.6 shows an excerpt from Privacy Rule §164.103 that describes one definition and the stakeholder class hierarchy, a lower ontology, that was derived from this definition.

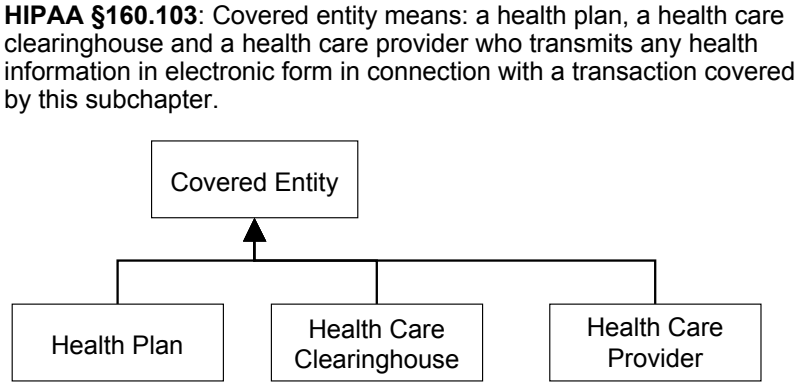

Figure 2.6: Example lower ontology derived from one definition

Using these lower ontologies, engineers can select which terms are relevant to their organization and thus restrict their focus to legal requirements that contain only those terms. However, because the class relation is transitive, engineers should also consider all the "higher" classes in the transitive closure of their selected terms. For example, an engineer who selects the term "health care clearinghouse" to describe their organization should also consider legal requirements that affect a "covered entity." This decision is logically based on evidence supported by the lower ontology in Figure 2.6. Therefore, these lower ontologies will help engineers systematically demonstrate their rationale for broadening or restricting requirements coverage based on their interpretation of these terms and their respective kinds.

2.5.3 Balancing Rights and Obligations

Regulations contain statements that describe transactions, in which two actors engage in a shared action such as a delegation, provision or notification [36]. For example, the obligation presented in Figure 2.5 that states "a group health plan... must provide [notice] upon request to any person," is a transaction between "a group health plan" and "any person." To ensure broader requirements coverage, engineers must infer legal requirements that cover these other parties in transactions; these parties are assigned to the `target` role in a frame-based requirement. For example, from the obligation in Figure 2.5 an engineer can infer a permission, depicted in Figure 2.7, in which "any person may receive [notice] upon request from a group health plan..."

Frame:	Permission (Inferred)
Pattern:	[*subject*] [modality] [*act*] [*object*] {upon [*condition*]} {from [*target*]}
Trace:	ID 5i, Line 1:0, Source: 164.520(a)(2)(i)(B)(ii)
Slots	**Values**
condition	*upon…* request
subject	any person
modality	may
act	Receive
object	a notice
target	*from…* A group health plan that provides health benefits solely through an insurance contract with a health insurance issuer or HMO *A group health plan that creates or receives* protected health information in addition to summary health information as defined in §164.504(a) *A group health plan that creates or receives* information on whether the individual is participating in the group health plan, or is enrolled in or has dis-enrolled from a health insurance issuer or HMO offered by the plan

Figure 2.7: Example implied permission inferred from a stated obligation

The implied permission in Figure 2.7 is different from the stated obligation in several ways. The frame header information (e.g., Frame, Pattern and ID in the Trace) have changed to reflect that this permission was inferred. The traceability information (e.g., Line and Source) to the legal text remain the same to show from which legal statement (an obligation) this permission was inferred. Among the slots, the property values for the `subject` and `target` have been swapped and the original `target` pattern text "*to...*" has been replaced with the new text"*from...*" to reflect the new `act` "Receive" that was implied by the obligation. By maintaining a reusable list of transaction verbs (e.g., provide, receive, notify, etc.) and delegation verbs (e.g., permit, require, request, etc.) (see Appendix D), several of these requirements can be automatically identified from stated transactions and delegations by existing tool support [36]. In addition, existing tool support can use these verbs to identify under-specifications and omissions resulting from missing `target` roles.

2.5.4 Prioritizing Requirements through Exceptions

Regulations include exceptions between permissions, obligations and refrainments that create priorities between legal requirements. For example, Figure 2.8 shows an excerpt from Privacy Rule §164.512(f)(2) that grants a permission (rule 182) to disclose protected health information (PHI) to law enforcement with one exception. The exception phrase"*Except for disclosures required by law as permitted by §164.512(f)(1)*" contains a cross-reference to the previous paragraph §164.512(f)(1). The previous paragraph describes four other permissions to disclose PHI to law enforcement for the purpose of reporting gunshot wounds (rule 178) and when requested by subpoena (rules 179-181). The italicized exception phrase creates a priority between the first permission (rule 182) and these four other permissions (178-181) in which the first permission has a lower priority that the other four permissions.

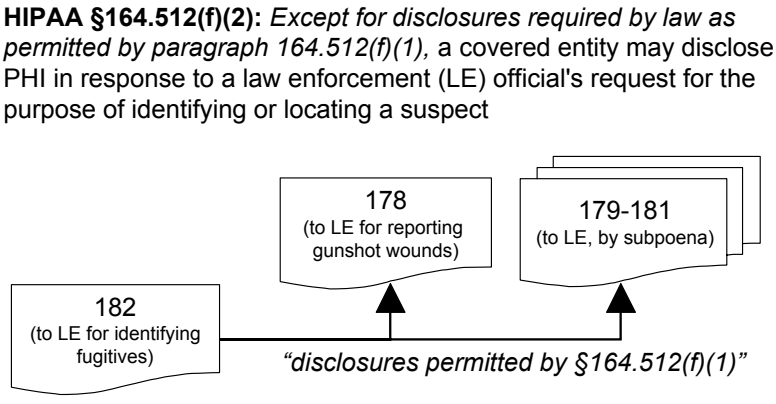

Figure 2.8: Example of four priorities derived from one exception

The FBRAM assists with the acquisition, maintenance and presentation of these priorities between legal requirements. Recall that the document model is used to map paragraph indices to

individual legal requirements, which are identified using the frame-based markup. Because the upper ontology distinguishes phrase-level exceptions that contain cross-references, engineers can quickly identify which requirements are affected by these exceptions. Using the document model, the engineer can formalize the cross-reference citation and automatically define the domain and range of a formal exception. The combination of these two models reduces the time required by engineers to identify these priorities and eases the procedure for formalizing these priorities in a priority hierarchy.

2.6 Chapter Summary

In this chapter, the Frame-Based Requirements Analysis Method (FBRAM) abstract model is presented, which consists of the four method artifacts, including the document model, a reusable, domain-independent upper ontology, the phrase heuristics and the frame-based markup language. The process model shown in Figure 2.1 coordinates these artifacts in the FBRAM process that consists of three method procedures: apply the document model, apply the markup language and parse the annotated regulation. The document model is necessary to preserve forward- and reverse-traceability between legal requirements and indexed paragraphs in the legal text, which in turn is required to determine which legal requirements are indexed by cross-references. The phrase heuristics are used in coordination with the markup language to map concepts from the upper ontology onto the legal text. In addition, the markup provides special operators for identifying natural language requirements patterns and removing attributive, logical and referential ambiguity. The result of the FBRAM process is a set of legal requirements that can be used to perform various types of analysis, including: presenting requirements using standard templates, sorting requirements in a lower ontology, balancing rights and obligations and prioritizing requirements through exceptions.

Chapter 3

Validity and Empirical Design

The scientist is not a person who gives the right answers, he's one who asks the right questions.
Claude Lévi-Strauss (1908–)

Science and empiricism seek to identify repeatable observations of the world through case studies and experimentation. Several factors introduced by the investigator or imposed by the environment affect the kinds and scope of claims that an investigator can make regarding his or her scientific contributions. The Frame-Based Requirements Analysis Method (FBRAM) represents a contribution to knowledge that is formalized by the abstract model and theory presented in Chapter 2. In Section 3.1, we explore the case study research methodology used to acquire this theory. The experimental research methodology used to validate the theory is described in then Section 3.2.

3.1 Grounded Theory and Case Study Design

The abstract model presented in Chapter 2 formalizes a *grounded theory*, which is systematically obtained from a dataset and is valid for that dataset [71]. Because an individual's knowledge is limited by personal experience and subject to personal opinion and bias, researchers must employ a rigorous research methodology to systematically derive theory from repeatable observations of the world. The research methodology used to discover the abstract model employs a descriptive or exploratory, multi-case study design [48] directed at discovering constructivist knowledge claims and outcomes, as opposed to explanatory case studies that examine underlying causes [165]. This design is necessary to first understand *what* we are observing before we can understand *how* or *why* it comes about. Unlike controlled experiments designed to reject a null hypothesis that has a narrow focus, a case study seeks to answer research questions that have a broad focus. Case studies trade the powerful external validity or generalizability of experiments for the ability to analyze observable phenomena when measurements in a controlled laboratory setting are cost-prohibitive or impractical.

This section presents the research questions, units of analysis, case selection, materials and strategy for mitigating threats to validity. The empirical results that were acquired by applying

this case study design and that serve to validate the abstract model are presented in Chapter 4.

3.1.1 Research Questions

Research questions use a broad focus to describe or explain environmental phenomena that cannot be controlled in a laboratory setting. The multi-case study used to acquire the abstract model was designed to answer two fundamental research questions:

RQ_1: What kinds of legal requirements and ambiguity exist in legal texts?

RQ_2: What can software engineers do to systematically acquire legal requirements from legal texts?

Research question RQ_1 investigates the necessary formal semantics to represent, codify and formalize legal requirements for the purpose of analysis, such as identifying and classifying ambiguity, sorting or prioritizing requirements, etc. Research question RQ_2, however, investigates the human inferences that are required to systematically transform statements of law into legal requirements governing information systems. These inferences include deconstructing the natural language context into formal assertions by identifying syntactic and semantic queues, such as condition and exception keywords and cross-references, that indicate how to interpret laws and regulations during legal requirements acquisition.

3.1.2 Units of Analysis

The units of analysis in the multi-case study include natural language features, such as sentences, clauses and phrases in legal documents. The analysis procedure consists of systematically identifying, enumerating, decomposing, codifying and classifying language features using concepts and properties in a reusable, upper ontology. As similar features are aligned with a shared concept or property, reusable heuristics are developed to consistently identify these natural language features. The upper ontology is iteratively constructed from this analysis. The heuristics that coincide with the introduction of new concepts and properties must be validated by applying them to previously analyzed and forthcoming documents to ensure consistency across the entire dataset. Limitations, which bound external validity, and contradictions, which bound reliability, are sought after, documented and addressed to the maximum extent possible.

3.1.3 Case Selection and Materials

The multi-case study examines six cases, which appear chronologically in Table 3.1, to investigate research questions RQ_1 and RQ_2, above. The cases were purposefully selected to reduce confounding factors until the theory was reasonably settled. The theory is "settled" when no contradictions and ambiguities are observed within the theory, which is based upon units that are enumerable and discrete. For example, if a theory is constructed by enumerating sentences in a legal document, then an ontology is settled for a specific document (the dataset) when that ontology can be used

to classify every sentence in the document without contradiction or ambiguity. This approach enables the investigator to reliably and incrementally scale the theory to new cases without becoming overwhelmed by multiple, confounding factors.

Table 3.1: Cases and materials studied to discover the abstract model

Case Name	**Case Materials**
Goals	The most frequent 100 goals from over 1200 goals that were acquired from over 100 Internet privacy policies in the finance and healthcare domains [27, 26].
Facts	HIPAA fact sheet titled "Protecting the Privacy of Patients' Health Information" that summarizes the HIPAA Privacy Rule for patients and consumers [129].
Practices	HIPAA Privacy Rule, 45 CFR §164.520-164.526, governing patient information access, consent, notification and review of privacy practices [36].
Privacy	HIPAA Privacy Rule, 45 CFR §160.310, §164.501-164.532, governing access to protected health information [31].
Safety	ETOPS, 14 CFR §121.374, governing aircraft maintenance and airworthiness certification [30].
Accessibility	Accessibility Standards, 36 CFR §1194, governing access to information by individuals with disabilities [32].

The six cases in Table 3.1 are studied in an order of increasing domain variability to control complexity and potential confounds. The first four case studies (Goals, Facts, Practices and Privacy) are limited to phenomena in a single domain, information privacy. Among these four cases, the Goals and Facts case studies are formative and conducted early, leading to substantial insights and gains in new theory. For example, the Goals case study examines semi-structured goal statements that were previously distilled by Antón et al. [7], thus greatly reducing the complexity of the legal domain. Several privacy policies in the Goals case are legally required by the HIPAA Privacy Rule, which makes the Facts and Practices cases ideal for follow-on studies to transition from modeling semi-structured goal statements to modeling unstructured regulatory statements about similar domain phenomena. In addition, the fact sheet in the Facts case summarizes the HIPAA Privacy Rule for consumers and thus excludes complex issues of cross-referencing and document structure that appear in the latter cases. The Practices case study was the first to study regulations directly by examining four sections in the Privacy Rule that were also summarized in the Facts case.

The Practices, Privacy and Accessibility case studies analyze U.S. federal regulations that are "economically significant" or "major" rules and have a net present or annual cost to society in excess of $100 million, as estimated by the Office of Management and Budget [130]. Together, the first five cases all examine information privacy and information accessibility, which concerns product and service requirements. The Safety case was included in the multi-case study to evaluate whether the theory supports models of process requirements in a domain that is not traditionally information-intensive.

3.1.4 Mitigating Threats to Validity

The quality of a case study research design is evaluated by identifying and addressing *threats to validity* to the greatest extent possible. A well-designed case study will seek to mitigate these threats early and describe those threats that emerge during the conduct of the study with the reported findings. In this section, we examine the threats that this study design sought to mitigate early. In Section 6.1.1, we examine threats to validity that emerged during the conduct of the multi-case study. The following types of validity are discussed herein:

1. *Construct validity* is the correctness of operational measures used to collect data, build theory and report findings from the data [165], and the extent to which an observed measurement fits a theoretical construct [148].

2. *Internal validity* is the extent to which measured variables cause observable effects in the data [165].

3. *External validity* determines the scope of environmental phenomena or domain boundaries to which the theory and findings generalize [165].

4. *Reliability* describes the consistency of the theory to explain environmental phenomena over repeated observations and the repeatability of the operational procedures for collecting data [165].

The research methodology for descriptive case studies that seek to obtain grounded theory must identify and thwart threats to external validity, construct validity and reliability. Internal validity is relevant only when building explanatory theory that explains underlying causes and not descriptive theories evidenced by the abstract model presented in Chapter 2 [148, 165].

The research methodology employs the following case study protocol: for each subsequent case study, the investigator will: (1) validate existing theory, by (a) identifying limitations in existing theory; and (b) reconciling contradictions between existing theory and each new dataset; and (2) expand existing theory, by observing and correctly describing all new phenomena in each case. Limitations occur when the theory does not describe some new phenomena. For example, an observable limitation occurs if the upper ontology in Section 2.2 does not contain a concept to describe some new phenomena. Contradictions occur when a new phenomena is contrary to an existing theoretical construct. For example, if the concept "condition" in the upper ontology were assumed to mean "pre-condition," a state or event that occurs prior to some other state or event, but was used to describe durations and post-conditions, which describe states or events that occur during or after some other state or event, respectively. This contradiction is resolvable by broadening the definition to cover multiple phenomena or restricting the definition to one phenomena and introducing new concepts to cover the remaining, previously unobserved phenomena. Table 4.1 in Chapter 4 describes the results of applying this case study protocol to yield the evolution of the Frame-Based Requirements Analysis Method (FBRAM).

In the multi-case study design, construct validity is addressed by maintaining a chain of evidence that logically ensures the consistency and completeness of the theoretical constructs across multiple sources of data. The theoretical constructs consist of the abstract model elements described in the FBRAM process model, including the document model, the upper ontology, the phrase heuristics and the context-free markup language. The sources of data consist of the case study materials, including the policy goals, fact sheet and regulations, which were selected to incrementally introduce variability and diversity in domain phenomena. By applying the case study protocol above, the investigator identifies at which point a theoretical construct becomes invalid based upon some observable measurement and what steps are taken to reconcile this inconsistency by updating the theory. The protocol also ensures that, if a construct changes during the analysis, affected measurements are resampled using the new construct to ensure validity of the findings.

External validity in grounded theory is limited to the scope of the dataset by its epistemology [71]. Consequently, grounded theory is inherently conservative about claims of scalability or generalizability to other domains. To improve external validity, this case study design employs purposefully selected, individual cases described in Table 3.1. These cases are selected to incrementally expand the scope (i.e., variety and frequency of change) of domain phenomena that the theory claims to describe across three domains: privacy, accessibility and safety. In addition, this design employs replication logic described in the case study protocol. At each stage of the replication and based upon the current state of the grounded theory, this logic is used to perform *literal replication* to predict equivalent findings and *theoretical replication* to predict contradictory findings for predictable reasons. Theoretical replication was conducted to incrementally observe *intra-domain variation* in the first four cases (Goals, Facts, Practices, Privacy) and *inter-domain* variation in the last three cases (Privacy, Safety and Accessibility), as discussed in Section 3.1.3.

Reliability concerns the ability to repeat the findings within a single case, as opposed to repeating findings across multiple cases, which concerns external validity. To improve reliability, the case study design employs the above case study protocol, software-based tools and a case study database. The software-based tools implement the theory described in Chapter 2, including a parser based on the context-free markup language and a generator for presenting the legal requirements using the frame-based requirements template. The case study database is the collection of all artifacts acquired by applying the theory to the domain on a per-case basis, including: numbered legal requirements codified using the upper ontology; all case materials, including the policy goals, fact sheet and regulatory documents, structured using the document model; and the instances of all phrase heuristics that appear in each case.

3.2 Experimental Design

Whereas case studies are used to build theory through induction from purposefully selected cases, experiments are used to test theory by attempting to disprove a fundamental proposition that is deducible from that theory [138]. This proposition, called a *hypothesis*, is tested by measuring

the effect of a treatment, called an *independent variable*, on an observable phenomena, called the *dependent variable*. The design of experiment serves to attribute and control error or confounds that can arise while manipulating, observing and measuring these variables [64]. Without proper experimental controls, it is possible to attribute the observed effects to the wrong causes and thus draw an inaccurate conclusion.

The abstract model presented in Chapter 2 formalizes a theory of legal requirements acquisition. The experimental design described in this section measures the effect of the second procedure in the FBRAM process model (see Figure 2.1), called *apply the markup language*, on legal requirements acquisition. The experiment also serves to compare and contrast the effects of applying this second procedure with the effects of traditional practice, which consists of applying natural language templates to acquire requirements.

The remainder of this section describes the dependent and independent variables, the hypotheses, the participant population and recruitment method, and the experimental environment, procedure and stimulus used to collect observations from the population in order to test the hypotheses.

3.2.1 Dependent and Independent Variables

In an experimental design, we are interested in measuring the effect, if any, of the independent variables on the dependent variables. There are two independent variables observed in this experiment during legal requirements acquisition: (1) the effect of the upper ontology and (2) the effect of the upper ontology with the phrase heuristics. The two dependent variables, completeness and consistency, are defined and measured as follows:

1. *Completeness* means the extent to which a set of requirements describe all desirable and undesirable behaviors in a given environment for a specific purpose. *Coverage* is a synonym for completeness in this dissertation. Achieving a *complete* requirements set is desirable but unrealistic because of the large number of environmental states of most present-day software systems of reasonable size and effectiveness. Therefore, completeness is measured using the set R_e of *expected requirements* for a specific purpose, which in this context is the alignment of generic software systems with specific laws, and the set R_a of *acquired requirements* by a participant in the experiment. *Recall* is the measure for completeness, defined as the proportion of expected requirements acquired to the total number of expected requirements, illustrated by the following formula: $|R_e \cap R_a|/|R_e|$

2. *Consistency* means the extent to which each engineer acquires the same requirements from the same source, such as a given legal text, for the same purpose. In requirements engineering, consistency refers to the absence of conflicts between multiple requirements [85], whereas the former definition used in this dissertation refers generally to the absence of differences. The former definition is narrowly applied by measuring consistency in terms of the set R_e of *expected requirements*, treating all acquired but unexpected requirements as inconsistent with the purpose of this experiment. *Precision* is the measure for consistency, defined as the

proportion of expected requirements acquired to the total number of acquired requirements by a participant in this experiment, illustrated by the following formula: $|R_e \cap R_a|/|R_a|$.

Expected Legal Requirements

The set R_e of expected legal requirements was acquired from the sample legal text in Figure 3.2 that is used in the experimental procedure described in Section 3.2.4. These requirements are predicted by the phrase heuristics (see Table 2.2) and acquired using the markup application procedure. As discussed in Section 4, the heuristics have been validated in a multi-case study and shown to be unambiguous across three separate U.S. Federal regulations, including the HIPAA Privacy Rule from which the sample legal text was obtained. The seven expected legal requirements appear below: the modal verb phrases that match the phrase heuristics appear in *italics*; the references to the legal text from which each requirement was obtained appear in **bold**.

P_1 (Permission) Except as provided by paragraph (a)(2) or (3) of this section, an individual *has a right to* adequate notice of the uses and disclosures of protected health information that may be made by the covered entity, and of the individual's rights and the covered entity's legal duties with respect to protected health information. **§164.520(a)(1)**

P_2 (Permission) An individual enrolled in a group health plan *has a right to* notice: from the group health plan, if, and to the extent that, such an individual does not receive health benefits under the group health plan through an insurance contract with a health insurance issuer or HMO. **§164.520(a)(2)(i)(A)**

P_3 (Permission) An individual enrolled in a group health plan *has a right to* notice: from the health insurance issuer or HMO with respect to the group health plan through which such individuals receive their health benefits under the group health plan. **§164.520(a)(2)(i)(B)**

O_4 (Obligation) A group health plan that provides health benefits solely through an insurance contract with a health insurance issuer or HMO, and that creates or receives protected health information in addition to summary health information as defined in 164.504(a) or information on whether the individual is participating in the group health plan, or is enrolled in or has disenrolled from a health insurance issuer or HMO offered by the plan, *must*: maintain a notice under this section. **§164.520(a)(2)(ii)(A)**

O_5 (Obligation) A group health plan that provides health benefits solely through an insurance contract with a health insurance issuer or HMO, and that creates or receives protected health information in addition to summary health information as defined in 164.504(a) or information on whether the individual is participating in the group health plan, or is enrolled in or has disenrolled from a health insurance issuer or HMO offered by the plan, *must*: provide such notice upon request to any person. **§164.520(a)(2)(ii)(B)**

E_6 (Exclusion) A group health plan that provides health benefits solely through an insurance contract with a health insurance issuer or HMO, and does not create or receive protected health information other than summary health information as defined in 164.504(a) or information on whether an individual is participating in the group health plan, or is enrolled in or has disenrolled from a health insurance issuer or HMO offered by the plan, *is not required to* maintain or provide a notice under this section. §**164.520(a)(2)(iii)**

E_7 (Exclusion) An inmate *does not have a right to* notice under this section. §**164.520(a)(3)**

During the markup application procedure to acquire the expected legal requirements, case-splitting was performed on the the subject, action and object slots in the frame-based markup. For example, the phrase "must: maintain and provide a notice under this section" was case-split to yield the separate legal requirements O_4 "must: maintain..." and O_5 "must: provide..." Complex conditions, exhibited in the legal requirements O_4, O_5 and E_6, were not case-split, however. Furthermore, these requirements are unaltered by the more advanced analysis techniques described in Section 2.5, including resolving cross-references and balancing rights with obligations.

Methods to Calculate Expected Legal Requirements Acquired

There are two methods to calculate the number of expected legal requirements acquired by each participant, described mathematically by the formula $|R_e \cap R_a|$; one method for traditional practice and the other method for the markup application procedure. Because traditional practice yields open-ended, minimally-structured natural language requirements, the calculation for traditional practice employs qualitative metrics for comparing natural language requirements. These metrics appear in Appendix E, were developed by Breaux et al. and validated in a prior case study[32]. The metrics are used to nominally measure the semantic variance between two requirements, based on the theory of goal refinement [32]. Using these metrics, the investigator first broadly determines if one requirement is equivalent (metric S-E) to another, or whether it refines (metric S-R) or generalizes (metric S-G) the other requirement. Next, the investigator measures specific phrases between the two requirements to identify refined concepts (metric P-R1) or new concepts (metric P-R2) and to identify generalized concepts (metric P-G1) or missing concepts (metric P-G2).

The number of expected legal requirements acquired by each participant corresponds to the number of measurements observed using the equivalence metric (S-E), excluding all requirements that are measured with variations in generality (metrics S-G, P-G) and refinement (metrics S-R, P-R). These variations are recorded and discussed as explanations of nominal variance but they are excluded from the number of expected legal requirements acquired. Variations in modality (metric P-M) that correspond to "balancing rights and obligations" (see Section 2.5.3) are deemed equivalent, because the implied rights and obligations are restatements of the legal requirement using different stakeholder viewpoints. All other variations in modality, including conflicts between permissions and refrainments, are deemed non-equivalent.

The second method for calculating the number of expected legal requirements acquired using the markup application procedure is automated with tool support. Because the markup application procedure employs a machine-parseable markup language, each character in each frame-based requirement is mapped to a unique index in the legal text from which the requirement was acquired. Thus, comparing the set R_a of acquired requirements with the set R_e of expected legal requirements corresponds to a pairwise comparison of the character positions in the legal text of two frame-based requirements. In addition to checking that the text in each requirement originated from the same location in the legal text, the upper ontology concepts assigned to each requirement must match for the two requirements to be deemed equivalent; otherwise, the two requirements are non-equivalent.

The hypotheses that will test the effect of the independent variables on the dependent variables is now discussed.

3.2.2 Falsifiable Hypotheses

A hypothesis is falsifiable, if it is capable of being tested by experience and shown to be false [138]. In experimental designs that employ statistical significance tests, the investigator attempts to disprove a *null hypothesis*, which is a statement that an independent variable has "no effect" during a systematic observation of a dependent variable [64]. Using an appropriate statistic, the investigator will reject the null hypothesis and accept an alternate hypothesis, also called the *research hypothesis*, if the statistic shows the null hypothesis is false within a pre-determined, statistical probability. Investigators can only disprove hypotheses; accepting the alternate hypothesis is not equivalent to proving the alternate hypothesis is true. Instead, the alternate hypothesis is accepted as a reasonable explanation until contradictory evidence that disproves the alternate is obtained.

The following hypotheses are tested in this experiment to measure the effect of the independent variables (the applied upper ontology and phrase heuristics) on the dependent variables (consistency and completeness of acquired legal requirements):

$H_{1.0}$ Traditional practice yields *equivalent or greater coverage* than using the upper ontology during legal requirements acquisition.

$H_{1.1}$ Using the upper ontology yields *greater coverage* than traditional practice during legal requirements acquisition.

$H_{1.2}$ Using the phrase heuristics and upper ontology yields *greater coverage* than using only the upper ontology during legal requirements acquisition.

$H_{2.0}$ Traditional practice yields *equivalent or greater consistency* than using the upper ontology during legal requirements acquisition.

$H_{2.1}$ Using the upper ontology yields *greater consistency* than traditional practice during legal requirements acquisition.

$H_{2.2}$ Using the phrase heuristics and the upper ontology yields *greater consistency* than using only the upper ontology during legal requirements acquisition.

The two null hypotheses, $H_{1.0}$ and $H_{2.0}$, assume the markup application procedure has no effect on the completeness and consistency of expected legal requirements acquired and that traditional practice is sufficient for this domain. The alternate hypotheses $H_{1.1}$ and $H_{2.1}$ assume that the upper ontology significantly increases legal requirements' completeness and consistency, respectively, over traditional practice. The third alternate hypotheses $H_{1.2}$ and $H_{2.2}$ further assume that the phrase heuristics and upper ontology together facilitate greater completeness and consistency than using only the upper ontology to acquire legal requirements. The following sub-sections describe the hypothesis tests for completeness and consistency.

Test for Completeness

The test for completeness determines if the markup application procedure has any observable effect on the proportion of expected legal requirements acquired to the total number of expected requirements, called the *recall*. The null hypothesis $H_{1.0}$ is true if the markup application procedure has "no effect" on completeness as compared to traditional practice. $H_{1.0}$ is rejected if the mean recall of participants using only the upper ontology is *significantly* greater than the mean recall of participants using traditional practice. Similarly, the alternate hypothesis $H_{1.2}$ is rejected if the mean recall of participants using the upper ontology with the phrase heuristics is *not significantly* greater than the mean recall of participants using only the upper ontology. Significance is measured using the Student's T-test statistic for comparing two means and a *p-value* $= 0.05$ [1]. The Student's T-test statistic assumes the sample population is normally distributed, the two samples are independent and the sample variances are equivalent.

For n participants, the mean recall of expected legal requirements acquired is calculated as follows: $Recall = (\sum_{i=0}^{n} |R_e \cap R_a|/|R_e|)/n$

Test for Consistency

The test for consistency determines whether the markup application procedure has any observable effect on the proportion of the expected legal requirements acquired to the total number of acquired requirements, called *precision*. The null hypothesis $H_{2.0}$ is true if the markup application procedure has "no effect" on the consistency of expected legal requirements acquired as compared to traditional practice. $H_{2.0}$ is rejected if the mean precision of participants using only the upper ontology is *significantly* greater than the mean precision of participants using traditional practice. Similarly, the alternate hypothesis $H_{2.2}$ is rejected if the mean precision of participants using the upper ontology with the phrase heuristics is *not significantly* greater than the mean precision of participants using only the upper ontology. Significance is measured using the Student's T-test statistic for comparing two means and a *p-value* $= 0.05$ [1].

For n participants, the mean precision of expected legal requirements acquired is calculated as follows: $Precision = (\sum_{i=0}^{n} |R_e \cap R_a|/|R_a|)/n$

3.2.3 Participant Population and Recruitment

The participant population from which the sample is drawn consists of undergraduate and graduate students enrolled in software engineering courses at North Carolina State University (NCSU). This population includes individuals interested in all aspects of software engineering, from requirements and design to implementation and testing. The experiment measures several demographic factors of the sample population that may influence participant performance, including: the total number of years of software industry experience; the presence of experience reading and interpreting laws or regulations; and the participants' first (L1) and second (L2) language fluency to identify non-native English speakers.

The participants are recruited using a coursework incentive, in which the experiment is delivered through a graded assignment following a relevant lecture on requirements engineering. Participation in the experiment is anonymous and voluntary; the course instructor will not know which students choose to participate in the experiment, although, the instructor will know which students completed the assignment. Participants will be randomly divided into three conditions distinguished by the requirements acquisition task that they will perform: (a) the *traditional practice group* uses templates to specify requirements [69, 85, 153]; (b) the *ontology group* uses only the upper ontology and the markup application procedure to specify legal requirements; and (c) the *ontology with heuristics group* uses the upper ontology, phrase heuristics and the markup application procedure to specify legal requirements.

Random assignment of participants to conditions ensures that the influence of confounding variables is reduced because these variables have an equally likely chance of affecting any one condition. If the sample size is small, counter-balancing is used to ensure a potential confound, such as the number of years of software industry experience, is equally distributed across conditions.

3.2.4 Environment, Procedure and Materials

The experiment is conducted in a monitored laboratory setting to reduce unwanted interference with the experimental procedure. In this setting, participants are individually assigned to a computer workstation where the experiment is conducted electronically and participants are required to complete the experiment within one hour. Table 3.2 describes the experimental procedure, which is decomposed into six chronologically ordered steps. Each step coincides with the estimated time required to complete the step, the description of the step and the materials used to complete the step, which appear in *italics*.

The estimated time for participants to complete each step of the procedure is based on the maximum time required by 12 participants in a pilot study conducted in a graduate requirements engineering seminar at NCSU during the 2008 Spring semester [24]. In the procedure, each condition

Table 3.2: Experimental procedure to evaluate legal requirements acquisition

Step	Time	Description
1	1 min.	The participant reads, electronically signs and receives a copy of the *informed consent form*, which introduces the purpose of the experiment and his or her rights under the Institutional Review Board (IRB), as required by the National Science Foundation under 45 CFR §690.101-124.
2	5 min.	The participant studies a *tutorial*, which introduces the participant to relevant terminology needed to complete the task in step 5.
3	5 min.	The participant completes a *tutorial competency test*, which measures the participant's recognition of the terminology using closed-form questions.
4	7 min.	The participant studies a *procedure* with examples that describe the task they will perform in step 5.
5	30 min.	The participant performs the task to acquire legal requirements from a *sample legal text*.
6	2 min.	The participant completes a *demographic survey*, which measures variables, such as participant gender, years of industry experience, language fluency, etc.

receives the same informed consent form and demographic survey during steps 1 and 6, respectively. The tutorial and competency tests during steps 2 and 3 are tailored to the three conditions. In step 2, the traditional practice group studies requirements in the solution and problem domains [85] using standard templates [85, 153]. Table 3.3 illustrates the example template provided to participants for specifying requirements in the problem domain; Table 3.4 illustrates the example template provided for the solution domain. In step 2, the ontology group and the ontology with heuristics group both study the upper ontology concepts from Section 2.2 with one example per concept (see Table 3.5).

Table 3.3: Template for specifying requirements in the problem domain

Template	**The** <stakeholder type> **shall be able to** <capability> **during** <performance> **of** <event> **if** <operational condition>.
Example	**The** covered entity **shall be able to** provide a privacy policy to the individual **during** the first service delivery **if** the covered entity has a direct treatment relationship with the individual.

In step 3, each of the three groups classifies the 12 legal requirements in Figure 3.1 based upon the terminology in their respective tutorials as follows: the traditional practice group classifies the requirements as either (1) in the solution domain, (2) in the problem domain, or (3) none; the

Table 3.4: Template for specifying requirements in the solution domain

Template	**The** <system> **shall** <function> <object> **every** <performance> <units>.
Example	**The** visitor directory **shall** record the individual's patient identifier **every** service delivery.

Table 3.5: Concepts and definitions in the upper ontology

Concept	**Definition**
Permission	Any action that an actor is permitted to perform; permissions include the rights of actors. Example: the covered entity may use or disclose protected health information.
Obligation	Any action that an actor is required to to perform. Example: The covered entity must permit the individual to request restrictions on disclosures.
Refrainment	Any action that an actor is required to not perform. Example: The covered entity may not use protected health information in violation of such a restriction.
Exclusion	Any action that an actor is not permitted, required, or prohibited from performing. Example: The covered entity is not required to agree to such a restriction.

ontology group and ontology with heuristics group both classify the requirements as either (1) a permission, (2) an obligation, (3) a refrainment, (4) an exclusion or (5) none. Each participant receives the 12 requirements in a randomized order to reduce contrast effects, such as primacy and recency effects, that the act of classifying one requirement may impose on the classification of another [146]. These requirements were selected as a stratified sample from the HIPAA Privacy Rule and Accessibility Standards to both emphasize phenomena in the problem and solution domains, separately and in combination, and to cover each of the four modalities that appear in the upper ontology, including permissions, obligations, refrainments and exclusions.

During step 4, the three groups receive individualized instructions that describe the exercise for their condition to be performed in step 5. The traditional practice group is instructed to read a sample legal text and type requirements into an empty table using the templates provided in step 2. The table allows an unlimited number of entries where each entry corresponds to a separate requirement. The ontology and ontology with heuristics groups are instructed on how to use a special text editor developed for applying the markup language to the sample legal text and a compiler to test the correctness of the applied markup against the context-free grammar in Appendix B. The instructions provided to the ontology with heuristics group also introduce the phrase heuristics that participants should use to identify the correct modality for a requirement. These phrase heuristics include the modal verb phrases that appear in the sample legal text. In step 5, all three groups receive the same sample legal text that appears in Figure 3.2.

Q_1: Controls and keys shall not require tight grasping, pinching or twisting of the wrist.

Q_2: A material change to any term of the privacy notice may not be implemented prior to the effective date of the notice.

Q_3: Federal agencies cannot claim a product is not commercially available because no product meets all the standards.

Q_4: A covered entity is not required to agree to a restriction on disclosures.

Q_5: An inmate does not have a right to a privacy notice.

Q_6: The individual has a right of access to their protected health information.

Q_7: The product must allow people to interrupt, pause and restart the audio.

Q_8: The individual retains the right to obtain a copy of the privacy notice.

Q_9: The covered entity may use protected health information.

Q_{10}: The required statements may be altered to reflect the fact that the notice covers more than one covered entity.

Q_{11}: The covered entity need only produce the protected health information once in response to a request for access.

Q_{12}: The web form shall allow people using assistive technology to access the information.

Figure 3.1: Twelve legal requirements appearing in the tutorial competency test

3.3 Chapter Summary

In this chapter, the research methodologies used to acquire and validate the Frame-Based Requirements Analysis Method (FBRAM) are presented. The case study research methodology to acquired the abstract model and grounded theory employs a multi-case study design that consists of a set of research questions, units of analysis and purposefully selected cases, which include several prominent U.S. federal regulations. The threats to validity are discussed and procedures for addressing these threats are detailed in the context of grounded analysis. A human subject, experimental design is presented to validate the FBRAM for multiple users using the same sample legal text. The experimental design evaluates three conditions on legal requirements acquisition and specification: traditional requirements engineering practice, and the effects of using only the upper ontology versus using the upper ontology with the phrase heuristics in the FBRAM markup application procedure. The experimental design consists of a set of falsifiable hypotheses, participant recruitment and selection criteria, and the environment, procedure and materials that are used to conduct the experiment. Together, the multi-case study design and human subject, experimental design constitute the complete validation plan for the abstract model.

164.520 Notice of privacy practices for protected health information.

(a) Standard: notice of privacy practices.

(1) Right to notice. Except as provided by paragraph (a)(2) or (3) of this section, an individual has a right to adequate notice of the uses and disclosures of protected health information that may be made by the covered entity, and of the individuals rights and the covered entity's legal duties with respect to protected health information.

(2) Exception for group health plans.

(i) An individual enrolled in a group health plan has a right to notice:

(A) From the group health plan, if, and to the extent that, such an individual does not receive health benefits under the group health plan through an insurance contract with a health insurance issuer or HMO; or

(B) From the health insurance issuer or HMO with respect to the group health plan though which such individuals receive their health benefits under the group health plan.

(ii) A group health plan that provides health benefits solely through an insurance contract with a health insurance issuer or HMO, and that creates or receives protected health information in addition to summary health information as defined in 164.504(a) or information on whether the individual is participating in the group health plan, or is enrolled in or has disenrolled from a health insurance issuer or HMO offered by the plan, must:

(A) Maintain a notice under this section; and

(B) Provide such notice upon request to any person. The provisions of paragraph (c)(1) of this section do not apply to such group health plan.

(iii) A group health plan that provides health benefits solely through an insurance contract with a health insurance issuer or HMO, and does not create or receive protected health information other than summary health information as defined in 164.504(a) or information on whether an individual is participating in the group health plan, or is enrolled in or has disenrolled from a health insurance issuer or HMO offered by the plan, is not required to maintain or provide a notice under this section.

(3) Exception for inmates. An inmate does not have a right to notice under this section, and the requirements of this section do not apply to a correctional institution that is a covered entity.

Figure 3.2: Sample legal text appearing in the requirements acquisition exercise

Chapter 4

Findings of the Multi-case Study

If you wish to measure according to its rules something which does not accord with your rules, forget your ways, you must first seek its rules!

Richard Wagner (1813–1883)

The descriptive, multi-case study described in Section 3.1 serves to develop and validate the method procedures and artifacts from the FBRAM that are presented in Figure 2.1 in Chapter 2. This section presents the evolution of the FBRAM, answers the research questions from Section 3.1.1 and demonstrates the extent to which the FBRAM is complete across three domains, including privacy, accessibility and safety. The study findings presented herein consist of a comparative analysis of the different case study findings, which is motivated by the results of applying the upper ontology to each case document described in Table 3.1. Following the findings of the comparative analysis, the unique effects of several of the upper ontology statement and phrase-level concepts are discussed with comprehensive examples from relevant cases.

Table 4.1 presents this evolution including the new concepts, method artifacts and method procedures that were discovered during each successive case study. Several of the phrase heuristics and concepts in the upper ontology were first discovered in the Goals study [27, 26]. Notably, facts were not discovered until the Facts study [28], likely because goal-oriented methods focus on goals and traditionally exclude this legal concept from goal-oriented analyses. As the theoretical foundations continued to settle, new artifacts and automated analysis procedures emerged. For example, the Goals, Facts and Practices studies originally encoded rules in a simple frame logic, called the Knowledge Transformation Language (KTL), which supported open-ended and Boolean queries [26, 36]. This logic was eventually mapped to Description Logic [33] with translation to the Web Ontology Language (OWL) [115] and reasoning support provided by the Pellet OWL-DL reasoner [151]. Upon discovery of the Definition concept, the lower ontology was realized and evidenced by the stakeholder hierarchy and product hierarchy derived from the Privacy and Accessibility studies, respectively. To reduce the effort in encoding rules, the Privacy study replaced the KTL with a tabular frame format [31], which was eventually supported by a machine-readable, context-free markup language during the Safety study [29]. Finally, the document model was developed late in

Table 4.1: Evolution of the Frame-Based Requirements Analysis Method

Case Name	New Concepts	New Artifacts	New Procedures
Goals	• Permission • Obligation • Refrainment • Action • Transaction	• Phrase Heuristics • Upper Ontology • Frame Logic	• Queries
Facts	• Fact • State		• Resolving Ambiguity
Practices	• Definition • Exclusion	• Stakeholder Hierarchy (Lower Ontology)	• Resolving Cross-References and Continuations • Balancing Rights and Obligations
Privacy		• Tabular Frames • Constraint Catalogue	• Priority Hierarchy
Safety		• Context-free Markup	• Applying Markup
Accessibility		• Document Model • Product Hierarchy (Lower Ontology)	

the evolution of the method to support resolving cross-references and continuations.

Research question RQ_1 presented in Section 3.1.1 asks "What kinds of legal requirements and ambiguities exist in legal texts?" Table 4.2 summarizes the types and frequency of legal requirement statements and phrases in the empirical results from the multi-case study. These observations are formalized in the reusable, upper ontology presented in Section 2.2. In each study, a sentence and phrase in the case document was mapped to exactly one statement and phrase-level concept in the upper ontology. In addition, the upper ontology was applied to each sentence in each paragraph of the case document for complete statement-level and phrase-level coverage. The upper ontology was not applied to preliminaries (e.g., title page, table of contents), appendices, division titles (e.g., part, section, and paragraph headings) or footnotes. The natural language patterns acquired by the FBRAM tool-supported process were compared to the upper ontology to identify inconsistencies (e.g., transaction verbs imply the statement must contain a phrase classified by the "target" concept). This rigorous application demonstrates that the FBRAM is complete, because each statement and phrase is classified by exactly one concept in the upper ontology. These results are limited to two participants working in tandem in multiple domains, including information privacy, accessibility and aviation safety; see the experimental results in Section 5 for an evaluation of the FBRAM using multiple participants.

Several important insights were observed across these successive cases. The Goals study demonstrates how modalities for permissions and obligations are removed from policy statements during the goal-mining process [27, 26]. Goals are primitive expressions that omit modality, conditions

Table 4.2: Empirical results: frequency of upper ontology concepts

Case Name	Permission	Obligation	Refrainment	Exclusion	Fact	Definition	Action	Transaction	State	Condition	Exception	Instrument	Purpose
Goals	47	44	10	0	0	0	85	16	0	4	2	11	18
Facts	11	13	1	0	41	0	60	5	3	21	2	1	18
Practices	32	74	3	3	8	0	102	32	5	80	7	43	9
Privacy	251	0	49	0	0	15	300	195	0	554	58	0	307
Safety	2	42	3	0	2	3	60	7	1	19	2	6	13
Accessibility	1	89	10	2	14	22	96	2	22	51	11	13	19

and exceptions, which are critically important to representing policies and regulations. To restore this critical information, several assumptions were made, including that consumer protections are institutional obligations and consumer vulnerabilities are institutional permissions, which may generally be pessimistic and inconsistent across privacy policies. In addition, goals exclude expressions of definitions, facts and exceptions between rules; the requirements-related effects of which are discussed later in this section.

The Facts study provides insight into how one U.S. federal agency summarized a U.S. federal regulation in a consumer fact sheet and what effect this summarization has on how regulatory statements are presented to specific audiences [28]. While these summaries are *informative*, software engineers must not assume such fact sheets are complete representations of the law because they omit important information. For example, modal phrases that indicate whether a statement is a permission, obligation, etc. are frequently prefixed by the adverb "generally" in conjunction with omissions of exclusions, conditions and exceptions, which serve to clarify under what specific situations these legal requirements apply. Below, the obligation $O_{FS.19}$ from the consumer fact sheet summarizes several permissions, obligations and exclusions. For example, the technical stakeholder category "covered entity," which is typically used throughout the regulation, is limited to the subset "doctors," an entity with whom consumers are more familiar. The adverb "generally" in $O_{FS.19}$ conceals several legal requirements that appear in the HIPAA Privacy Rule but that are omitted from the fact sheet, including: exceptions made for group health plans (a type of covered entity) described by permission $P_{520.1}$, exceptions for inmates described by exclusion $E_{520.8}$ and conditions concerning the content of notices described by obligation $O_{520.10}$.

$O_{FS.19}$: Doctors generally will provide the notice.

$P_{520.1}$: Except as provided by... an individual has a right to adequate notice.

$E_{520.8}$: An inmate does not have a right to notice.

$O_{520.10}$: The covered entity must provide a notice that contains...

Another insight appears in the Practices and Privacy studies of the HIPAA Privacy Rule, where a distinct "stakeholder focus" is observable in which a majority, 74% and 100%, respectively, of the subjects of rules are stakeholders. For example, each of the rules above, $P_{520.1}$, $E_{520.8}$ and $O_{520.10}$ from HIPAA §164.520(a) and (b) state the subject as "an individual," "an inmate" and "the covered entity," respectively. In the Practices study, 72% of the rules with non-stakeholder subjects describe the content of privacy notices and written denials of access. Contrast the stakeholder focus with following obligation $O_{1194.76}$ from 1194.23(a) in the Accessibility Standards that illustrates a legal requirement with a product focus. The "product focus" is observable in the Accessibility study in which 92% of the subjects are products, product features or materials and only 3% of the subjects are users of such products. The remaining 5% of the subjects in this study are other stakeholders, such as federal agencies.

$O_{1194.76}$: Telecommunications products... shall provide a standard non-acoustic connection point for teletypewriters.

The stakeholder vs. product focus affects whether legal requirements are written to describe the environment or the machine [167], respectively, as discussed in Section 1.2.1. The stakeholder focus incorporates a broad view of what constitutes requirements, which describe "stakeholder needs" [85] and other stakeholder-related phenomena in the environment [167]. However, the product focus conforms to traditional requirements engineering practice in which requirements consist of "system objectives" [162] and "descriptions of how the system should behave, or of a system property or attribute" [153]. If a software engineer is trained in, or preferential to, one approach over the other, then he or she may overlook critical legal requirements in his or her analysis of legal texts. This shift in focus and the consequences of overlooking one focus or the other is visibly pronounced in the product and stakeholder hierarchies produced from definitions, discussed further in Section 4.2.

The observed phenomena codified in the FBRAM upper ontology, summarized in Table 4.2, yield several important effects, which are the consequences of making systematic inferences to acquire legal requirements from legal texts. These inferences provide answers to research question RQ_2, "What can practitioners do to acquire legal requirements from legal texts?" that is presented in Section 3.1.1. The following sections present several of these inferences with comprehensive examples from the multi-case study.

4.1 The Effect of Facts

Facts describe an act or state of being that is conditionally true. Factual statements do not contain the modal verb phrases in Table 2.2, which indicate permissions, obligations, etc. Facts serve a variety of important functions in the legal documents studied, such as recounting historical events and legislative goals and summarizing legal requirements. The Facts study examined a HIPAA

consumer fact sheet that revealed a rich source of facts, which constitute 63% of that document as compared to between 3%-9% in the three U.S. federal regulatory documents studied. The following four types of facts have been identified in the multi-case study: descriptive, deontic, historical and intentional facts. The number of facts identified in each case study appear in Table 4.3.

Descriptive facts clarify, elaborate or refine an existing legal statement.

Deontic facts describe implied permissions, obligations, refrainments or exclusions.

Historical facts describe past events.

Intentional facts describe future actions and the purpose of entities or actions.

The following list provides examples of each of the four types of facts. The most frequently occurring type is the Deontic fact, specifically those facts that imply exclusions. These facts coordinate broad exceptions to legal requirements, as evidenced by fact $F_{PP.9}$ from the Practices study. These exceptions are further discussed in Section 4.4 in regard to the effect of cross-references. Historical facts, such as fact $F_{CF.1}$ from the Facts study, describe historical context through past events, in contrast to intentional facts, such as fact $F_{CF.3}$, which describe the goal for which actions will be taken to respond to particular situations.

(Descriptive) $\mathbf{F}_{CF.70}$**:** Criminal penalties apply for certain actions such as knowingly obtaining protected health information in violation of the law.

(Deontic) $\mathbf{F}_{PP.9}$**:** The requirements of this section do not apply to a correctional institution that is a covered entity.

(Historical) $\mathbf{F}_{CF.1}$**:** The first-ever federal privacy standards to protect patients' medical records and other health information provided to health plans, doctors, hospitals and other health care providers took effect on April 14, 2003.

(Intentional) $\mathbf{F}_{CF.3}$**:** The first-ever federal privacy standards represent a uniform, federal floor of privacy protections for consumers across the country.

In the Facts study, facts serve to summarize the HIPAA Privacy Rule. Among the 47 facts, 23 state the subject of the fact as the regulation itself, such as "the regulation" and "the final rule," and confer permissions, obligations and other legal requirements to broad categories of stakeholders. For example, the fact $F_{CF.50}$ presented below and acquired from the consumer fact sheet uses the action verb "permit" to indicate a permission is conferred to covered entities to disclose protected health information. Contrast this summary fact with one of the corresponding permissions it represents, $P_{IP.174}$ from HIPAA §164.512(e)(1) that is re-topicalized for the covered entity. The summarizing facts generalize important information, such as the phrase "specific public responsibilities" in fact $F_{CF.50}$ that generalizes "judicial or administrative proceedings" contained in the permission $P_{IP.174}$.

Table 4.3: Empirical results: frequency of factual implications

	Descriptive	**Historical**	**Intentional**	**Permission**	**Obligation**	**Refrainment**	**Exclusion**
Goals	0	0	0	0	0	0	0
Facts	4	11	8	9	6	3	5
Practices	0	0	0	0	0	0	8
Safety	2	0	0	0	0	0	0
Accessibility	3	0	0	0	3	0	6

$\mathbf{F}_{\mathsf{CF.50}}$: The final rule permits covered entities to continue certain existing disclosures of health information for specific public responsibilities.

$\mathbf{P}_{\mathsf{IP.174}}$: The covered entity may disclose protected health information for judicial or administrative proceedings.

In summary, facts provide important mechanisms for recounting historical events, coordinating or summarizing requirements, and describing intentions and outcomes.

4.2 The Effect of Definitions

Definitions describe the meaning of a *term-of-art*, which is "a word or phrase having a specific, precise meaning in a given specialty, apart from its general meaning in ordinary contexts" [68]. These statements are formalized using binary class relations, informally called "is-a" relations, that map a general concept, called a *hypernym* or super-class, to a specialization of that concept, called a *hyponym* or sub-class. In addition, property relations, informally called "has-a" relations, can be expressed to map whole concepts, called *holonyms*, to their constituent parts or properties, called *meronyms*. These relations are expressed using a formal language, such as the Unified Modeling Language [76] or Description Logic [11] and constitute a lower ontology in the FBRAM.

Section 2.5.2 illustrates how a definition is formalized into *terms* and their respective hypernyms, called *kinds*. Because "kinds" described in one definition can appear as "terms" in another definition, these formalizations must be integrated to create a comprehensive, terminological hierarchy. Figure 4.1 represents a lower ontology expressed in the UML that resulted from integrating terms, kinds and other relevant roles. These stakeholder concepts were observed in definitions from HIPAA §160.103, §164.500 and §164.501 and in the subject and target role of legal requirements that were acquired from HIPAA §164.502-64.532. Each box corresponds to a stakeholder concept and arrows point from hyponyms to hypernyms (from sub-classes to super-classes). Gray boxes correspond to concepts that appeared in the subject and target properties of rules but did not appear in the

definitions. The gray boxes illustrate that definitions only describe 45% of stakeholders identified in the Privacy study, demonstrating that engineers cannot rely on the definitions alone to cover all the governed actors. In each box, the concept name appears above the paragraph references to the HIPAA Privacy Rule that contain requirements governing these stakeholders.

For a particular stakeholder, identifying which legal requirements apply in a given situation includes evaluating rules that apply to general classifications of that stakeholder (e.g., via the transitive closure). For example, a "Group Health Plan" must consider rules that directly apply to their stakeholder class as well as rules that apply to their more general classifications, including the "Health Plan" and "Covered Entity" class. 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 legal requirements that govern this class as well as to requirements that govern "Law Enforcement" (the agency or profession), in general. Not all memberships are transitive, however; rules that apply to a "Correctional Institution" do not apply to an "Inmate," or vice versa, despite the fact that inmates have membership in a correctional institution. In summary, the stakeholder hierarchy defines a legal requirement's scope of impact, thereby helping engineers to visually understand which classes of stakeholders are affected by a requirement and by supporting formal reasoning about similarities and conflicts between rules.

Figure 4.2 illustrates how the stakeholder hierarchy can be used to compare legal requirements using formal reasoning. These legal requirements all pertain to the provision of notice and were formalized in the Practices study [36]. The domain description defines a stakeholder specialization hierarchy that includes the covered entity (CE), the health plan (HP), the group health plan (GHP) and the healthcare provider (HCP). The stakeholder hierarchy is a subset of the comprehensive stakeholder hierarchy that appears in Figure 4.1. Using the stakeholder hierarchy, the legal requirements are organized into a requirements specialization hierarchy (displayed horizontally) that compares the requirements by mapping the nouns, verbs, adjectives, etc. to formal predicates in Description Logic formula using the Semantic Parameterization process [33]. In this hierarchy, arrows point from specialized requirements to more abstract requirements. For example, the goal $O_{520.15}$ specializes goal $O_{520.13}$ by including the additional adverb "automatically." The dotted arrow between obligations $O_{520.4}$ and $O_{520.2}$ indicates a conflict inferred from the conflicting deontic modalities "must," which indicates an obligation, and "is not required to," which indicates an exclusion. While this type of formal modeling is intensive for large U.S. federal regulations, the cost of non-compliance may warrant identifying high-risk areas of the regulation for modeling to ensure subtle distinctions between legal requirements are identified and implemented appropriately.

As previously mentioned, the HIPAA Privacy Rule uses a "stakeholder focus" in the presentation of legal requirements. However, the Accessibility Standards use a "product focus", wherein the topic of legal requirements consist of products, product features or components. While many stakeholder and product concepts are described in definitions, product features and components generally appear in the purposes and conditions of legal requirement statements. For example, in Accessibility Standards §1194.23(h), the condition "a telecommunications product delivers output

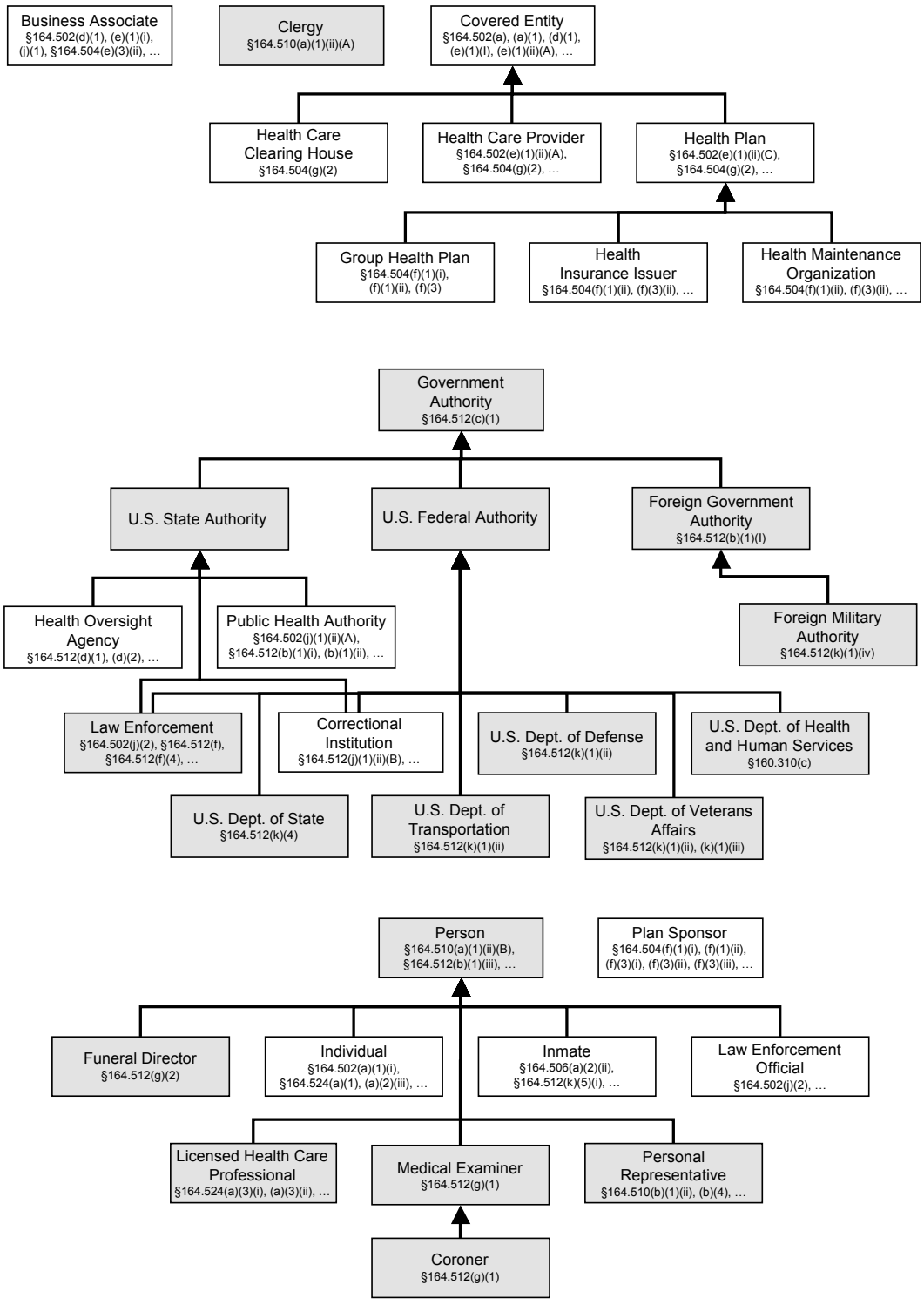

Figure 4.1: Stakeholder hierarchy acquired from the Privacy study

Eight Legal Requirements Pertaining to Provision of Notice

$O_{520.2}$: The GHP must provide notice to any person.
$O_{520.4}$: The GHP is not required to provide notice to any person.
$O_{520.7}$: The CE must provide notice to any person or individual.
$O_{520.8}$: The HP must provide notice to any person or individual.
$O_{520.10}$: The HCP must provide notice to the individual.
$O_{520.13}$: The CE must provide electronic notice to the individual.
$O_{520.14}$: The CE must provide a paper copy of the notice to the individual.
$O_{520.15}$: The CE must automatically provide electronic notice to the individual.

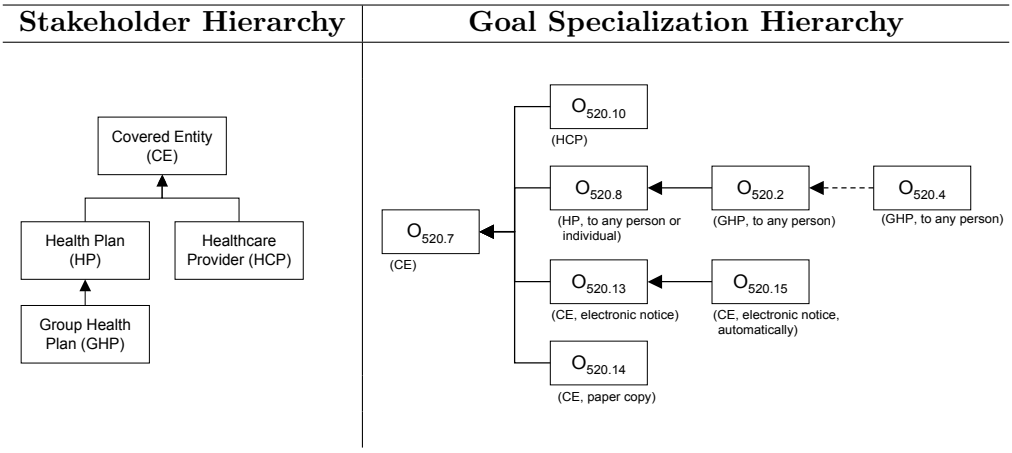

Figure 4.2: Organizing legal requirements by inferring goal specialization hierarchies

by an audio transducer..." indicates that some telecommunications products contain "audio transducers," which is a product component. These features and components must be acquired and formalized in conjunction with definitions to create a comprehensive product hierarchy.

Figure 4.3 presents a lower ontology expressed in the UML that was constructed from definitions in Accessibility Standards §1194.4 and the condition and purpose properties of extracted legal requirements from §1194.21-1194.41. Square boxes represent concepts described by nouns that were acquired from conditions, terms and kinds, whereas rounded boxes represent concepts described by activities that were acquired from purposes. Arrowhead lines lead from hyponyms to hypernyms (from sub-classes to super-classes) and diamondhead lines lead from meronyms to holonyms (from parts to wholes). The box with a dotted-line border at the bottom of the diagram denotes a placeholder concept, which serves to simplify the diagram. In each box, the concept name appears above paragraph references to the Accessibility Standards that contain requirements governing these products, product features and components.

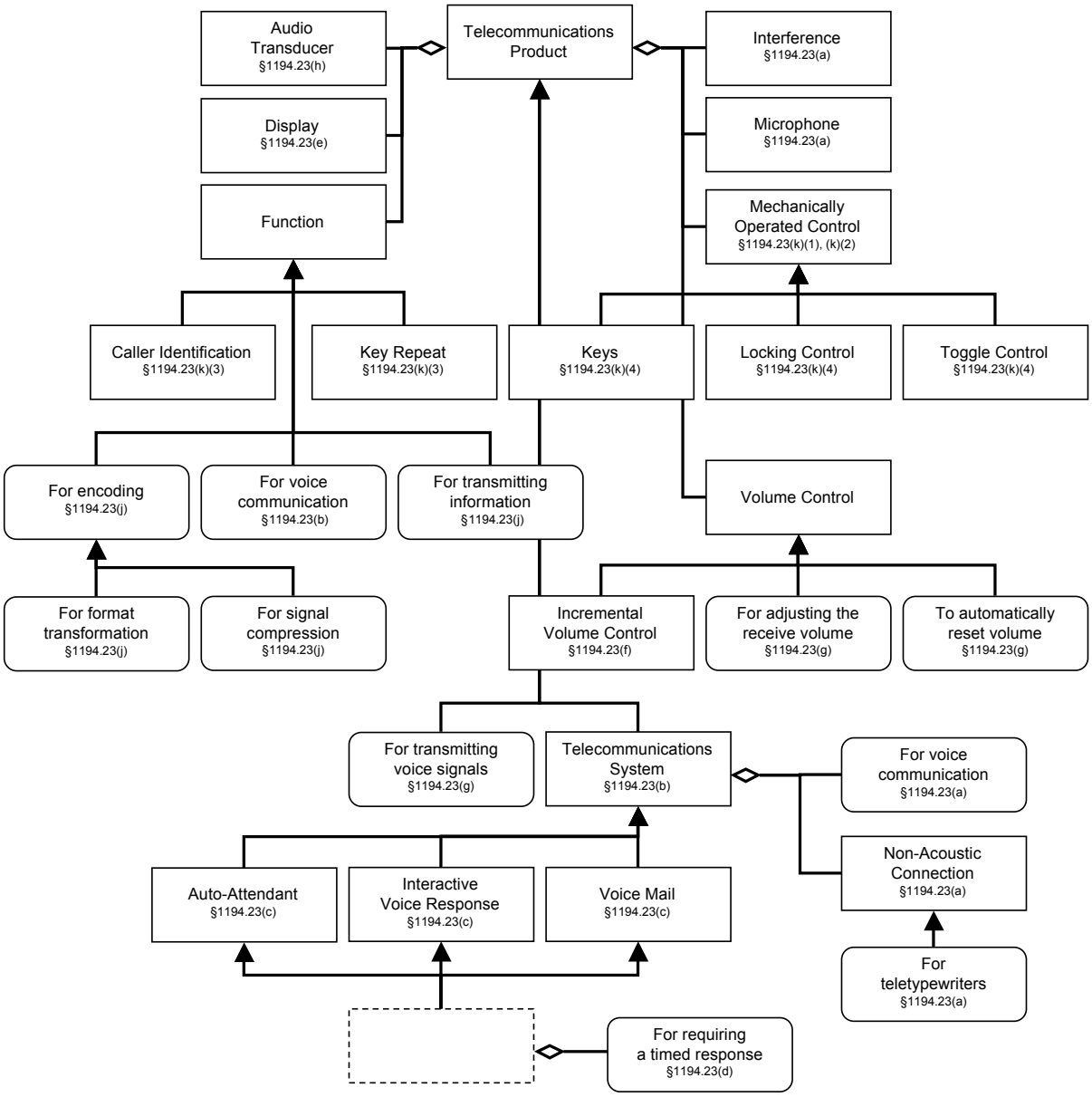

Figure 4.3: Product hierarchy acquired from the Accessibility study

4.3 The Effect of Constraints

Constraints are phrases that refine concepts in legal requirements to restrict the applicability of regulatory rules to only those situations that regulators believe are relevant. Gause and Weinberg compare constraints to boundaries that define the state space of a solution [69]. Concepts in the upper ontology, such as the subject, target, condition, purpose and instrument, are realized by words or phrases that constitute constraints on the applicability of regulatory rules. For software engineers, the correct design of systems governed by these regulations depends upon the satisfiability of constraints using available technology. For the purpose of this dissertation, a constraint is *satisfiable*, if there exists a hardware or software process that 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 satisfiability in terms of the environment. Because satisfying these constraints can introduce trade-offs, engineers may only be able to optimize for specific constraints and never be able to satisfy all constraints, all the time. Mylopolous et al. have termed this procedure *satisficing* in the context of high-level goals [124].

In the Privacy study, 861 constraints were identified as indicative of additional requirements that stakeholders must satisfy before accessing, using or disclosing information. The constraints were classified as *non-ephemeral*, if satisfiable by information that can be maintained across multiple transactions, such as disclosures of patient information; otherwise, they are classified as *ephemeral*, if heavily dependent upon circumstances specific to a single transaction. Table 4.4 presents a classification of 861 non-parameterized constraints that contains 235 non-ephemeral constraint 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 satisfiability. The non-parameterized constraints are catalogued [31] and discussed under the following sub-headings: stakeholder beliefs and determinations; contractual statements; and intended and inferred purposes.

Table 4.4: Empirical results: frequency and types of constraints

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

4.3.1 Beliefs and Determinations

A total of 431 constraints that were acquired during the Privacy study are satisfiable by stakeholder beliefs and determinations. They are classified into three non-disjoint, subsets based upon legal training, medical training or personal beliefs about circumstances that are required to satisfy these constraints. Each of these categories is separately discussed 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. To implement fully accountable transactions, analysts 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 4.5 illustrates three example constraints, some of which refer to specific laws while others refer to laws, in general.

Table 4.5: Example constraints on legal determinations

Reference	**Property**	**Value**
164.510(b)(2)(i)	Target	Who has lawful custody of an inmate or individual.
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 4.5, the constraint from 164.510(b)(2)(i) applies to a disclosure in which the covered entity (the subject) must decide if the recipient (the target) has lawful custody of an inmate or individual. The terms of this authorization are relevant to the custody of the inmate or individual at the time of access. At that time, a legal determination identifies which laws, if any, authorize the custody. 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, in advance, 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 [37].

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, whereas the remaining constraints should be satisfied by someone with appropriate knowledge and authority. Table 4.6 shows three 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 subject constraint from 164.510(b)(4) and the target constraint from 164.512(b)(1)(iv) are ephemeral because they must be individually satisfied for each transaction.

Table 4.6: Example constraints on medical determinations

Reference	**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.

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; individuals are victims or perpetrators of crimes; consent, or a lack of objection, to a disclosure is inferable from specific circumstances; and 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. 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 4.7 contains three example constraints that describe personal beliefs and determinations.

Table 4.7: Example constraints on personal beliefs

Reference	**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 individual is not present.

4.3.2 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 4.8 includes three example constraints that describe contractual statements.

Table 4.8: Example constraints on contractual statements

Reference	**Property**	**Value**
164.506(a)(1)	Subject	Who has obtained the consent of the individual for the disclosure.
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.

4.3.3 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 [10, 9, 39]. In traditional Role-Based Access Control (RBAC) systems [144], 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) often imply data purposes (e.g., actions for which data is used). In this study, purposes are stated with respect to the act of access or as a constraint on the subject, object or target properties. The purposes inferred from subject and target constraints are equivalent to roles because they describe actions performed by the affected stakeholders. The purposes expressed in an object constraint denote for which actions the information may be used. Table 4.9 provides three examples: one purpose stated on the object, one purpose stated on the act itself, and one purpose stated as a role (the target properties).

Purposes present an exceptional challenge to software engineers who intend to guarantee that data is only used for intended purposes. Intended purposes provide explicit motivation for limiting

Table 4.9: Example constraints on intended and inferred purposes

Reference	Property	Value
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.
164.512(h)	Target	Who is engaged in procurement, banking, or transplantation of cadaveric organs, eyes, or tissue.

retention, whereas inferred purposes provide insufficient cause for expiring data within a software system. In Table 4.9, the purpose described in the object and purpose properties both are intended purposes for which the data is to be used or disclosed. When the purpose is fulfilled, further retention of this data is likely unnecessary. For the inferred purposes in the target property, however, it is uncertain if other potential purposes are also intended for the data.

4.4 The Effect of Cross-references

Cross-references serve to coordinate knowledge expressed in different regions of a legal text. In general, these cross-references are coarse-grained, meaning they refer to broad regions (paragraphs, sections, parts, etc.) and include legal statements that are not relevant to the context that contains the the cross-reference.

The cross-reference analysis was conducted by identifying cross-reference natural language patterns and expressing these patterns using regular expressions. This approach increases analysis coverage and consistency by ensuring that every instance of a particular pattern is identified across the legal text and that each of these instances is consistent with the type of cross-reference. The cross-reference type corresponds to a programmatic operation that determines how the reference is interpreted. There are five distinct operations: (1) the stand-alone reference operator, indicated by a period "." maps each reference phrase to a single division in the legal document; (2) the list reference operator, indicated by a plus "+," maps two references to a list of references, inclusive (i.e., a list of paragraphs excluding the indices of sub-paragraphs); (3) the current division operator, indicated by an asterisk "*," maps to the division that contains the reference; (4) the top-down reference operator, indicated by a number, maps to the division at the corresponding level of nesting starting from the top-most division (e.g., 1 maps to a part, 2 maps to a sub-part, 3 maps to a section, etc.); and (5) the continuation operator, indicated by a forward slash "/," maps to the list of sub-divisions (e.g., a list of sub-paragraphs for a paragraph)

Table 4.10 presents the cross-references natural language patterns identified in the Practices study. The table includes a unique pattern ID, the programmatic operation (Op) described above, the regular expression that describes the pattern and the number of instances of each pattern in the

Practices, Safety and Accessibility studies. The regular expressions are applied to the frame-based markup using tool-support to automatically identify and retrieve the referenced legal requirements using the document model.

Table 4.10: Cross-reference natural language patterns from the multi-case study

ID	Op	Regular Expression	Practices	Safety	Accessibility
1	.	§((?:+).(?:+)(?:(?:.+?)+)?)	28	0	5
2	.	paragraphs ((?:.+?)+) or ((?:.+?)+) of this section	1	0	0
3	.	paragraphs ((?:.+?)+) and ((?:.+?)+) of this section	1	0	0
4	+	paragraphs ((?:.+?)) through ((?:.+?)+) of this section	1	0	0
5	.	paragraph ((?:.+?)+) or ((?:.+?)+) of this section	9	1	0
6	.	paragraph ((?:.+?)+) of this section	42	3	0
7	.	paragraph ((?:.+?)+)	0	2	0
8	.	part (+) of this subchapter	1	0	0
9	*	this paragraph	1	0	0
10	2	this section	9	0	0
11	1	this subpart	6	0	0
12	0	this part	0	1	0
13	/	as follows	2	1	0
14	/	the following requirements	4	0	0

The tool-supported analysis yields mappings between a legal requirement, which contains the cross-reference, and the referenced legal requirements (e.g., all the requirements in a referenced paragraph). In the Practices study, this result includes 1,720 mappings between legal requirements. These mappings were manually classified to determine whether the mapping was intended by the context of the reference (positive) or not (false-positive) and whether the mapping is a refinement, exception or continuation. Refinements consist of legal requirements that elaborate upon the referencing legal requirement and constitute 56.8% of the mappings in this analysis; these include concepts that are defined in other paragraphs. Exceptions restrict the applicability of a referencing legal requirement and constitute 35.5% of the mappings. Continuations, like refinements, elaborate on information in the referencing requirement through subsequent sub-divisions and constitute 7.7% of the mappings. Overall, 63.0% were false-positive mappings, 35.5% were positive mappings and 1.5% were self-referential (e.g., references to "this paragraph" include the referencing legal requirement). Among the positive mappings, 52.1% of the referenced legal requirements contain additional cross-references to other legal requirements. Figure 4.4 illustrates a graphical representation of this study, wherein each requirement (a circle node) has a directed edge to another requirement, only if the requirement is referenced by a cross-reference. External references to paragraphs or other laws

beyond the scope of this study are represented by rounded-corner, rectangular nodes.

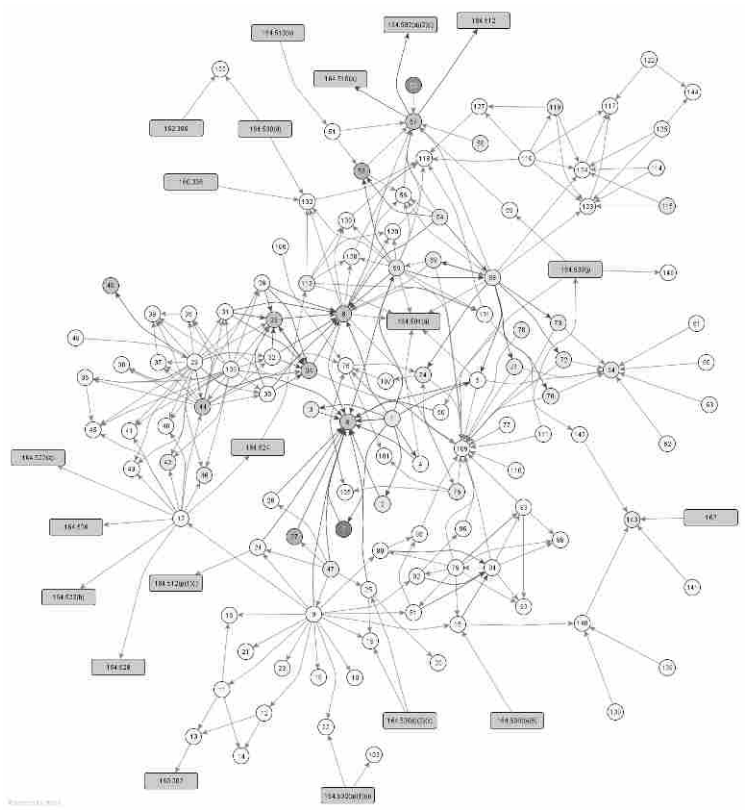

Figure 4.4: Cross-reference graph illustrating dependencies among legal requirements

The existing system of cross-referencing legal requirements in U.S. federal regulatory documents exhibits substantial ambiguity evidenced by a high proportion of false-positives and complex recursion exhibited by legal requirements that contain cross-references to other requirements with cross-references. To address this challenge, the FBRAM includes a technique to formalize cross-references that correspond to exceptions. In the Privacy study that yielded over 300 access control requirements, these cross-references produced over 12,205 mappings between acquired legal requirements. For example, consider the following exception (in italics) that appears in the legal statement from HIPAA §164.502(a):

> **HIPAA §164.502(a)**: A covered entity may not use or disclose protected health information *except as permitted or required by this subpart or by subpart C of part 160 of this subchapter.*

This exception corresponds to a mapping between two refrainments, to *not use* and *not disclose* protected health information, and the permitted and required uses and disclosures of subpart E of part 164 (this subpart) and subpart C of part 160. In subpart E, over 300 access-related requirements were identified to which this exception applies; subpart C of part 160 was not included in this analysis.

Figure 4.5 illustrates 12 of the 58 exceptions that were extracted in the Privacy study [31]. These 12 exceptions comprise 66 priorities between legal requirements that govern the use and disclosure of information. The boxes contain extracted rule numbers and brief descriptions of those requirements in parenthesis. White boxes represent permissions or "allow rules" whereas shaded boxes represent refrainments or "deny rules." The arrows denote a priority and lead from lower priority requirements to higher priority requirements. Higher priority requirements are the "exceptions" to the lower priority requirements. Requirements 1 and 2 are the lowest-priority refrainments relative to all other extracted requirements in the deny-first/ allow-later scenario depicted in the HIPAA Privacy Rule.

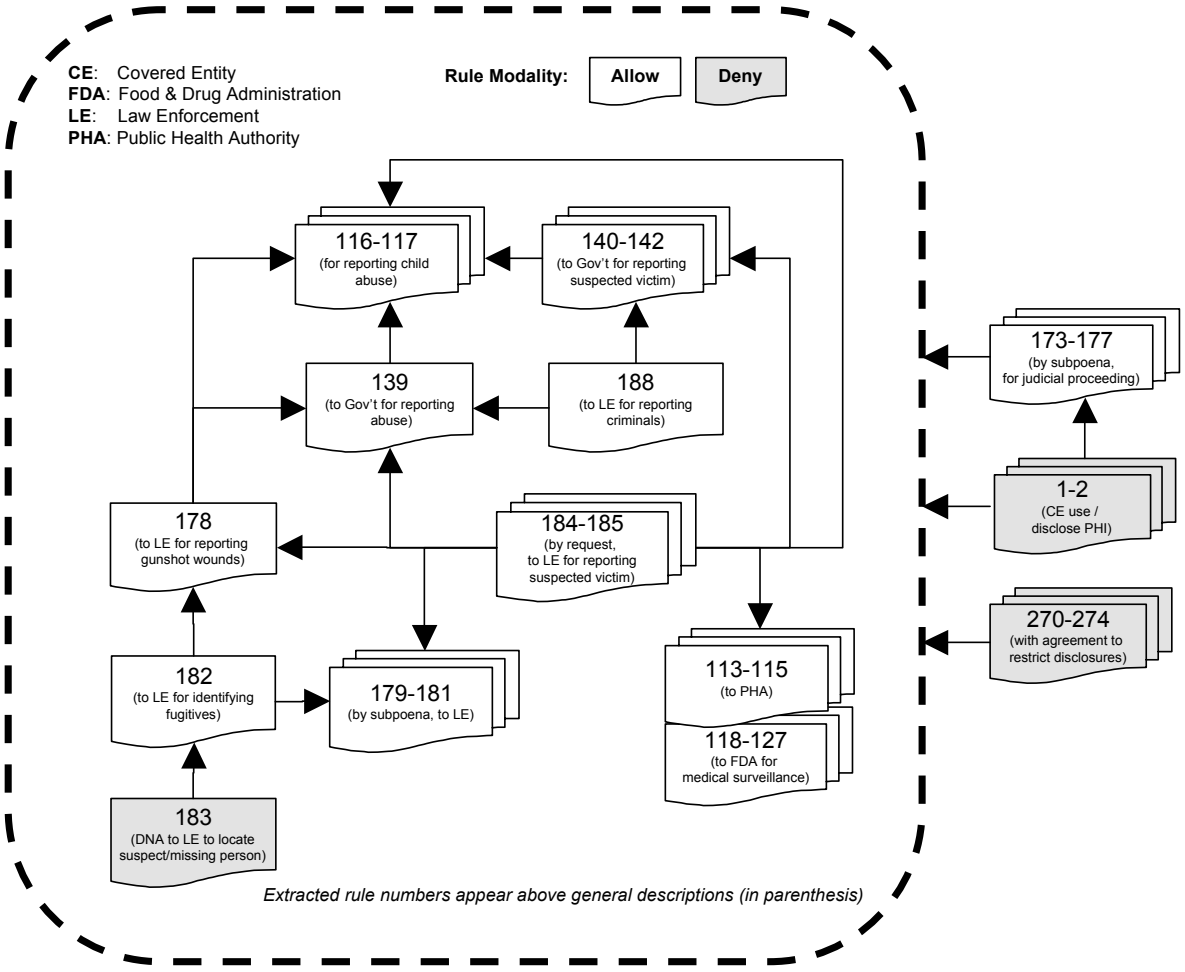

Figure 4.5: Priority hierarchy acquired from Privacy study

4.5 The Effect of Ambiguity

U.S. federal regulations contain intended and unintended ambiguity. Intended ambiguity includes are intended by law makers to be re-interpreted as business practices evolve and as capabilities to

comply with regulations change over time. For example, HIPAA §164.502(b)(1) states that an entity must make "reasonable" efforts to limit uses and disclosures of protected health information to the minimum necessary to complete a transaction. The word "reasonable" is an intended ambiguity: which subsets or health information are considered reasonable, (e.g., patient identifier, physician notes, list of medications, etc.) varies depending on the type of communities served, purpose of the transaction and the prevalence of relevant, existing technologies to apportion only relevant data.

Regulations also contain unintended ambiguities that are inherent to natural language syntax and semantics — or English. This dissertation addresses four types of ambiguity, which were introduced in Section 1.1.3 and are briefly summarized here:

> *Logical ambiguity* refers to English words that can be mapped to different logical interpretations. This includes aligning English conjunctions (and, or) with logical connectives, such as logical-and and logical-or.
>
> *Attributive ambiguity* is found in phrases that may be reasonably ascribed to more than one other phrase within a sentence. In part, this ambiguity is due to natural language context-sensitivity and multiple associativity in phrases corresponding logical connectives.
>
> *Referential ambiguity* occurs when a word or phrase has multiple meanings. This includes intensional and extensional polysemy [33], cross-references and words that have an anaphoric (backward-referencing) or cataphoric (forward-referencing) function, such as pronouns (this, that, they), noun phrases that use definite articles (the) and some adjectives (such).
>
> *Under-specification or omission* occurs when a word or phrase is missing from a sentence; this includes phrases that are implied by certain verbs.

Table 4.11 summarizes the number of occurrences of the four types of ambiguity identified during the multi-case study. The table presents the number of requirements (Requirements) per case study, the number of logical ambiguities corresponding to conjunctions (and) and disjunctions (or), attributive ambiguities and referential ambiguities corresponding to anaphora (backward-referencing), cataphora (forward-referencing), and cross-references and the number of under-specifications.

Table 4.11 highlights several noteworthy anomalies, which we now discuss. For example, the goal-mining process itemizes policy statements which likely removes most ambiguities because of inferences made by the analyst. These undocumented inferences may vary with the experience level of the analyst. In addition, the Practices study shows a large number of referential ambiguities that result from the high proportion of cross-references to statements. Because these paragraphs contain irrelevant information to the meaning of the cross-reference, there is also a high false-positive rate (63% on average in this study). For example, obligation $\mathbf{O}_{\mathsf{PP}.10}$ below that was acquired from HIPAA §164.520(b)(1) in this study refers to other legal requirements in the same paragraph from which this obligation was acquired. This referential ambiguity is measured by the number of legal requirements referenced by "this paragraph." Moreover, the referenced divisions in the legal text

Table 4.11: Empirical results: frequency of ambiguities

	Requirements	Logical (and)	Logical (or)	Attributive	Referential (anaphora)	Referential (cataphora)	Referential (cross-refs)	Under-specification
Goals	101	1	1	0	0	1	0	11
Facts	73	10	0	1	1	7	0	12
Practices	149	4	1	8	10	23	105	22
Safety	65	0	0	5	6	5	8	20
Accessibility	143	26	0	6	8	9	5	143

can themselves contain cross-references (51% on average in this study), which further compounds this type of ambiguity. See Section 4.4 for further discussion of the effect of cross-references.

> **$O_{PP.10}$** The covered entity must provide a notice that contains the elements required by this paragraph.

Lastly, the Access Standards in the Accessibility study contain legal requirements that closely approximate traditional requirements engineering practice. These legal requirements contain a high number of under-specifications, because this type of specification describes the functionality and behavior of machines without describing *who* or *what* interacts with this functionality. In this study, 64% of the under-specifications identified concern the subject of the action or *who* is required to implement a particular function or behavior. For example, the obligation $O_{1194.61}$ below from the Accessibility study describes a legal requirement for designing web pages. This requirement omits *by whom* the web pages shall be designed, because it presumes this requirement applies to everyone covered by this regulation.

> **$O_{1194.61}$:** Web pages *shall* be designed so that all information conveyed with color is also available without color, for example from context or markup.

4.6 Chapter Summary

The Frame-Based Requirements Analysis Method (FBRAM), which is formalized by the abstract model in Chapter 2, was developed and evaluated using a multi-case study conducted in three domains, information privacy, aviation safety and information accessibility. The multi-case study led to important insights for legal requirements acquisition, in general, including the various effects of facts, definitions, cross-references and ambiguity in coordinating legal requirements knowledge. Finally, this chapter illustrates how this knowledge can be formalized into product and stakeholder

hierarchies and priority hierarchies to aid engineers in prioritizing which legal requirements apply to their organizations and products.

Chapter 5

Findings of the Experiment

True works of art contain their own theory and give us the measurement according to which we should judge them.

Johann Wolfgang Von Goethe (1749–1832)

The experimental design described in Section 3.2 measures the completeness and consistency of legal requirements acquisition by comparing traditional requirements engineering practice with the Frame-Based Requirements Analysis Method (FBRAM). Completeness is measured by *recall*, which is the number of expected requirements acquired by participants divided by the total number of expected requirements. Consistency is measured by *precision*, which is the number of expected requirements acquired by participants divided by the total number of acquired requirements. The experiment has the following three conditions wherein participants analyze the sample legal text in Figure 3.2 to acquire the expected legal requirements in Section 3.2.1:

1. **Traditional Practice** consists of participants applying templates for the problem domain (see Table 3.3) and solution domain (see Table 3.4) to acquire legal requirements;

2. **Ontology** consists of participants applying the markup application procedure (see Figure 2.1) using only the upper ontology; and

3. **Ontology with Heuristics** consists of participants applying the markup application procedure (see Figure 2.1) using the upper ontology with phrase heuristics.

Table 5.1 shows the experimental results, including the recall, precision and variance for the three experimental conditions. Based on these results, we reject the null hypothesis $H_{1.0}$ that states the markup application procedure exhibits "no effect" on completeness and accept the alternate hypothesis $H_{1.1}$ that the markup application procedure using only the ontology yields significantly greater coverage than traditional practice ($p = 0.001$). In addition, we reject the null hypothesis $H_{2.0}$ that states the markup application procedure exhibits "no effect" on consistency and accept the alternate hypothesis $H_{2.1}$ that the markup application procedure using only the ontology yields significantly greater consistency than tradition practice ($p = 0.001$).

Regarding the two ontology conditions, with and without heuristics, we observe the ontology with heuristics only yields greater coverage that is weakly significant ($p = 0.063 < 0.100$) over using only the ontology, we do not accept hypothesis $H_{1.2}$. This means the heuristics do not significantly decrease the number of false-negatives or missed legal requirements and that the ontology alone is adequate to increase coverage over traditional practice. Alternatively, we do accept hypothesis $H_{2.2}$ that states the markup application procedure using the ontology with heuristics yields significantly greater consistency ($p = 0.030 < 0.050$). This result means the heuristics significantly decrease the number of false-positives by improving the participant's ability to assign the correct legal concept to each legal requirement. Together, these results suggest a need to standardize legal terminology and phrases to support legal requirements acquisition.

Table 5.1: Precision and recall for the three experimental conditions

Condition	**Precision**	**Variance**	**Recall**	**Variance**
Traditional Practice	.131	.071	.106	.054
Ontology	.692	.057	.610	.080
Ontology with Heuristics	.843	.049	.762	.089

Figures 5.1 and 5.2 show the distributions for precision and recall for all three conditions. In statistics, the *skew* describes the asymmetry of a distribution, wherein a rightward skew has the mass of the distribution to the left with relatively fewer observations to the right. In Figures 5.1 and 5.2 , we observe a significant rightward skew in precision and recall for traditional practice, indicating that participants generally were unable to consistently acquire complete legal requirements from the text. The nominal variance that explains this result is discussed in Section 5.2.

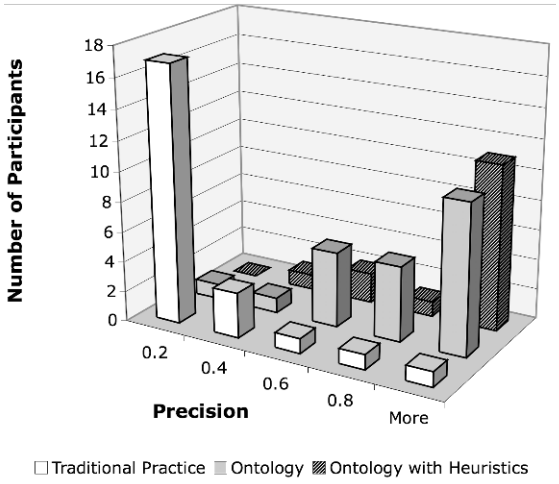

Figure 5.1: Precision distribution for all three conditions

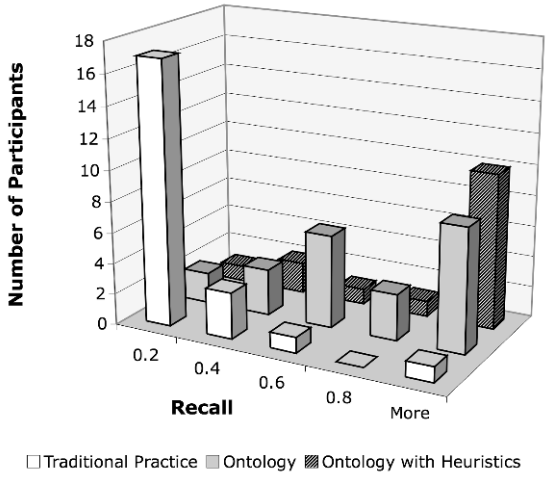

Figure 5.2: Recall distribution for all three conditions

The details behind the experimental findings reported in Table 5.1 are now discussed for the three conditions. The experimental procedure described in Section 3.2.4 requires participants to study a

tutorial, answer questions on a tutorial competency test and then perform an exercise to acquire legal requirements. The following sections review the results from the tutorial competency test and the exercise and propose possible explanations for variance within the experimental findings. Afterwards, demographic and performance measures from the experiment are discussed for the three conditions.

5.1 Findings from Traditional Practice

Findings from the traditional practice condition illustrate important strengths and limitations of this approach in the domain of legal requirements acquisition. As this section illustrates, participants generally agree on whether a legal requirement falls within the problem or solution domain with notable exceptions. In addition, participants make categorical inferences, such as integrating constraints from cross-references and viewing requirements from alternate stakeholder viewpoints, which cause notable differences between acquired and expected requirements. These findings are now discussed by analyzing the results from the tutorial competency test and the legal requirements acquisition exercise.

Twenty-three participants were given a short tutorial on traditional practice followed by a tutorial competency test in which participants were asked to classify a stratified sample of legal requirements as either in the problem domain, solution domain or none. These requirements were selected to emphasize phenomena in the problem and solution domains, separately and in combination with one another (see Figure 3.1 in Section 3.2.4). The test score corresponds to the proportion of participants who agree with the majority opinion for each classification. Figure 5.3 presents the test score distribution for the traditional practice tutorial competency test results, which illustrates a normal distribution with a mean score of 56.3%.

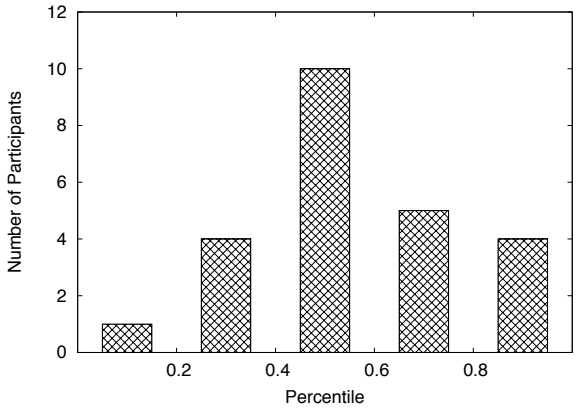

Figure 5.3: Test score distribution for traditional practice

The test results are itemized according to each question in Table 5.2, wherein the number of participants who classified each requirement by the problem domain (**P**), solution domain (**S**)

or none (**N**) are shown. The score per question is also shown with the majority classification designated by the colored table cells. Table 5.2 illustrates that most participants (>50%) agree with the majority opinion for seven of twelve requirements; for six of these seven requirements, twice as many participants agree with the majority opinion. The majority also agree that six out of seven requirements with human or organizational subjects (e.g., the individual, covered entity, inmate, etc.) are in the problem domain, whereas all three requirements with system subjects (e.g., products, web forms, controls and keys, etc.) are in the solution domain. Overall, participants agree on average with the solution domain (71.0%) more often than the problem domain (51.2%). On average, participants also believe 18.5% of the time that the legal requirements are not requirements in either the problem or solution domains. This proportion is especially discouraging because it indicates a potential disregard for the impact that certain legal requirements have on software systems.

Table 5.2 also illustrates when participants exhibit conflicting interpretations about when a requirement is in the problem or solution domain. If the requirement is missing the subject (the actor performing the action), participants may ascribe the action to a human, organization or system at their discretion. This observation applies to requirements Q_4 and Q_9 that were missing the subject and classified near equally (9-9 and 9-8, respectively) in the problem and solution domains. Similarly, participants classified Q_6 near equally (9-11) in the problem and solution domains, and this requirement has a human or organizational subject (the covered entity), but an action ascribable to a system function (provide access). Alternatively, legal requirements can be administrative in nature and describe product procurement (i.e., the problem of finding products to solve problems); an observation that coincides with participants near equally (11-10) classifying requirement Q_8 in the problem domain and none (neither solution or problem domains).

The traditional practice exercise consists of participants reading the legal text in Figure 3.2 and specifying requirements using templates that describe the problem domain (see Tables 3.3) and solution domain (see Table 3.4). Section 3.2.1 describes the method to calculate the number of expected requirements acquired by participants in the traditional practice condition. This method employs qualitative metrics described in Appendix E for acquiring nominal measurements in the form of alignments among the requirements acquired by participants and the expected requirements, which are presented in Section 3.2.1. The qualitative metrics yield measurements that describe semantic gaps between two requirements. Therefore, an acquired requirement can align with an expected requirement but still differ by an important concept expressed by a word or phrase. For an acquired requirement to be expected, there must be no measurable differences observed using the qualitative metrics.

Participants in this condition acquired 96 requirements, of which 89 requirements were aligned with the expected requirements but only 17 requirements were expected. Figure 5.4 shows the distribution for the number of requirements acquired by participants, the number aligned with the expected requirements and the number expected. These participants exhibited a mean precision of .131 with variance .071 and mean recall of .106 with variance .054. Figure 5.5 shows the distribution

Table 5.2: Tutorial competency test results for traditional practice

ID	Requirement	P	S	N	Score
Q_1	The individual has a right of access to their protected health information.	12	6	5	.522
Q_2	The covered entity may use protected health information.	14	3	6	.609
Q_3	The individual retains the right to obtain a copy of the privacy notice.	18	3	2	.783
Q_4	The required statements may be altered to reflect the fact that the notice covers more than one covered entity.	9	9	5	.391
Q_5	The product must allow people to interrupt, pause and restart the audio.	5	18	0	.783
Q_6	The covered entity need only produce the protected health information once in response to a request for access.	9	11	3	.391
Q_7	The web form shall allow people using assistive technology to access the information.	5	17	1	.739
Q_8	Federal agencies cannot claim a product is not commercially available because no product meets all the standards.	11	2	10	.478
Q_9	A material change to any term of the privacy notice may not be implemented prior to the effective date of the notice.	9	8	6	.391
Q_{10}	Controls and keys shall not require tight grasping, pinching or twisting of the wrist.	9	14	0	.609
Q_{11}	An inmate does not have a right to a privacy notice.	10	5	8	.435
Q_{12}	A covered entity is not required to agree to a restriction on disclosures.	14	4	5	.609

for the precision and recall for legal requirements acquisition using traditional practice.

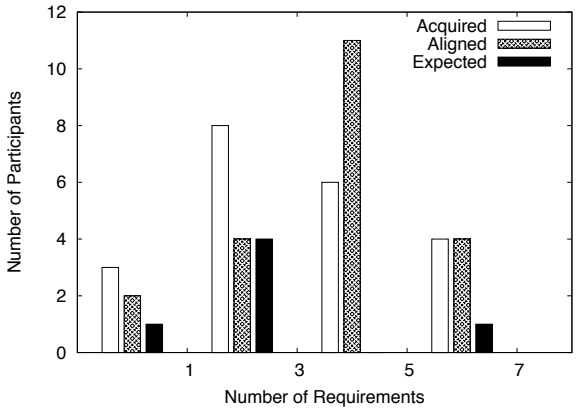

Figure 5.4: Requirements distribution for traditional practice

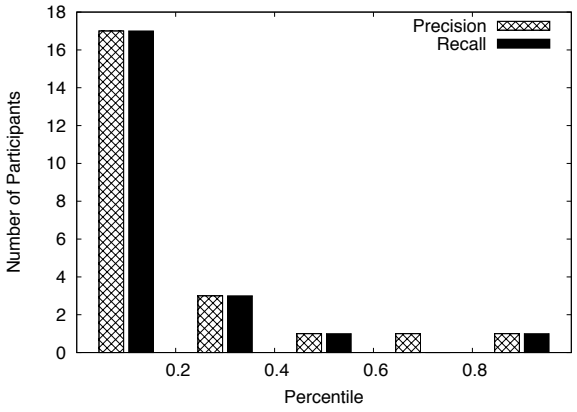

Figure 5.5: Precision and recall distribution for traditional practice

5.2 Nominal Variance in Traditional Practice

The measurements acquired using the qualitative metrics are used to explain the variance observed in the 72 aligned but unexpected requirements. Four scenarios are observed within these measurements: (1) the participant acquired one requirement that aligns with two different legal requirements; (2) the participant acquired two requirements that align with only one legal requirement; (3) the participant omitted relevant information from a requirement; and (4) the participant acquired a requirement with a different modality (e.g., by inferring the opposing viewpoint in a transaction as discussed in Section 2.5.3). These four scenarios are now discussed in the following sections.

5.2.1 Constraint Integration and Case-splitting

Constraint integration occurs when an analyst infers additional constraints on a requirement, for example, by applying knowledge from other sources including other sentences and paragraphs in the legal text. The experimental results indicate that constraint integration occurred 32 times based on analysis of measurements made using the P-G2 (new constraint) metric. For example, the following obligation $O_{T117.R1}$ and corresponding P-G2 measurements were acquired under the traditional practice condition:

> **$O_{T117.R1}$:** The individual shall be able to know the uses and disclosures of protected health information, if the individual is not an inmate nor has a group health plan that: provides health benefits solely through an insurance contract with a health insurance issuer or HMO

and does not create or receive protected health information other than summary health information as defined in 164.504(a) or information on whether an individual is participating in the group health plan, or is enrolled in or has disenrolled from a health insurance issuer or HMO offered by the plan.

P-G2: Includes "if the individual is not an inmate"

P-G2: Includes "nor has a group health plan that: ..."

The obligation $O_{T117.R1}$ generally aligns with the expected requirement P_1, which states "an individual has a right to adequate notice" (see the complete list of expected requirements in Section 3.2.1). The two P-G2 measurements for obligation $O_{T117.R1}$ align with separate expected requirements E_7 and E_6, respectively, and appear in subsequent paragraphs §164.520(a)(2)(iii) and §164.520(a)(3) in the legal text (see Figure 3.2), respectively. Thus, we may assume the participant specified obligation $O_{T117.R1}$ by inferring additional constraints from these two exclusions E_7 and E_6.

Case-splitting occurs when an analyst creates separate requirements from a single legal statement, for example, by identifying English conjunctions (and, or) that separate constraints and creating separate requirements for each constraint. Case-splitting occurred 12 times based on analysis of measurements made using the statement metrics. For example, the following two obligations $O_{T240.R1}$ and $O_{T240.R2}$ were mapped to the expected requirement P_1, which contains the phrase "notice of the uses and disclosures of protected health information that may be made by the covered entity, and of the individual's rights and the covered entity's legal duties with respect to protected health information." We may assume the participant distinguished two cases: (1) uses and disclosures; and (2) individual's rights and the covered entity's legal duties. In so doing, the participant created two separate notification requirements.

$O_{T240.R1}$: The individual shall be able to receive adequate notices of the uses and disclosures of health information by the covered entity.

$O_{T240.R2}$: The individual shall be able to receive the individual's rights and the covered entity's legal duties with respect to protected health information.

5.2.2 Under-specification and Omission

Under-specification and omission occurs when an analyst excludes an explicit constraint from a requirement. Omissions were observed in the results 62 times based on analysis of measurements made using the P-G2 (missing constraint) metric. Similarly, generalizations wherein the analyst uses a more general word or phrase than the word or phrase contained in the legal text occurred 43 times based on analysis made using the P-G1 (generalized concept) metric. The most frequent omission across the 89 aligned requirements occurred 17 times and consists of the following exception to the expected requirement P_1:

P-G2: Excludes exception "Except as provided by paragraph (a)(2) or (3) of this section"

In general, exceptions to legal requirements can be complex and require special care to ensure that requirements are correctly applied to a given situation. In Section 4.4, the impact of exceptions on prioritizing legal requirements is further discussed. Evidence that 65.2% of participants omit such exceptions during traditional practice further supports the need for specific methods to identify and appropriately interpret these critical phrases.

5.2.3 Changes in Modality

Changes in modality are measured using the P-M metric and characterize three patterns in the experimental results from traditional practice: (1) balancing rights and obligations, which occurred in 32 of 89 requirements; (2) stronger modalities (e.g., changing an exclusion to a refrainment), which occurred in 8 of 89 requirements; and (3) conflicts (e.g., changing an implied obligation to a permission), which occurred in 5 of 89 requirements. Table 5.3 presents the valid changes in modality when acquiring a legal requirement expressed in a legal text. As illustrated by the table, a legal permission may be changed to a permission, obligation or refrainment but not an exclusion, because an exclusion means the act is not necessarily permitted. A legal exclusion, however, can be changed to any modality, assuming this does not conflict with another law. In government, for example, legal permissions may be changed to refrainments under budgetary pressure, whereas, legal exclusions may be changed to permissions as an expansion of authority. We now discuss the three types of changes observed in the experimental results.

Table 5.3: Validity of changing modality of legal requirements

Legal Requirement	**Permission**	**Obligation**	**Refrainment**	**Exclusion**
Permission	True	True	True	False
Obligation	False	True	False	False
Refrainment	False	False	True	False
Exclusion	True	True	True	True

Balancing rights and obligations is a technique formalized by Breaux et al. [36], which is desirable for covered entities who need to decide which requirements affect their organization. For example, obligation $O_{\mathsf{T114.R4}}$ requires a health insurer to conditionally provide notice to an individual, which balances with the right of an individual to conditionally receive notice expressed in expected requirement P_3 using the "health insurer" viewpoint. Because these requirements communicate the same information from different viewpoints, they are deemed equivalent for this purpose of this experiment. In general, for the purposes of calculating precision and recall, acquired requirements that balance with expected requirements are deemed equivalent if they do not exhibit nominal variance measured by the remaining phrase-level metrics (P-R, P-G).

$O_{\mathsf{T114.R4}}$ balances with P_3: If an individual is enrolled in a group health plan and receives

benefits through a health insurer, the health insurer shall provide a privacy notice to the individual.

Several modality changes lead to interpretations that "strengthen" the modality of a requirement, including: (1) an exclusion is changed to a permission, obligation or refrainment; and (2) a permission is changed to an obligation or a refrainment. These interpretations are not necessarily conflicts, because an organization can presumably permit, require or prohibit actions that the law does not require or prohibit and an organization can require or prohibit actions that the law permits. For example, refrainment $R_{T105.R3}$, which demonstrates the first case with an obligation to not permit, states that inmates are prohibited from receiving a privacy notice. This interpretation is not inconsistent with expected requirement E_7, which states that inmates do not have a right to receive such notices but does not exclude the possibility of providing privacy notices to inmates.

$R_{T105.R3}$ further restricts E_7: An inmate shall not have a right to notice of privacy practices.

Several modality changes lead to conflicts with legal requirements, including: (1) a permission, obligation or refrainment is changed to an exclusion; (2) an obligation is changed to a permission or a refrainment; and (3) a refrainment is changed to a permission or obligation. Obligation $O_{T249.R2}$, which demonstrates the second case, states that group health plans are required to be permitted to provide individuals with a privacy notice. This implementation conflicts with P_2, which requires group health plans to provide such notices. In the acquired obligation, however, group health plans only have permission but not the obligation to provide such notices.

$O_{T249.R2}$ conflicts with P_2: The group health plan shall be able to make notice to the individual if the individual does not receive health benefits under the group health plan through an insurance contract with a health insurance issuer or HMO.

5.3 Findings from the Markup Application Procedure

Findings from participants performing the markup application procedure demonstrate the relative strength of the ontology over traditional practice in both completeness and consistency and the measurable improvement obtained in consistency by using the phrase heuristics. As discussed in this section, participants generally exhibit strong agreement on which legal statements are permissions and obligations but have difficulty distinguishing exclusions and refrainments from each other. These findings are now discussed by analyzing the results from the tutorial competency test and the legal requirements acquisition exercise.

Forty-seven participants were given a short tutorial on the upper ontology followed by a tutorial competency test in which participants were asked to classify a stratified sample of legal requirements as either a permission, obligation, refrainment, exclusion or none. These requirements were selected to emphasize each of these four modalities using unique phrase heuristics (see Figure 3.1 in Section 3.2.4). The test score corresponds to the proportion of participants who agree with the

classification determined by the corresponding phrase heuristic (see Table 2.2 in Section 2.3), which has been shown to be consistent and unambiguous across multiple case studies. Figure 5.6 presents the test score distribution for the ontology only and ontology with heuristics conditions, which have a normal distribution with mean scores of 62.7% and 64.1%, respectively.

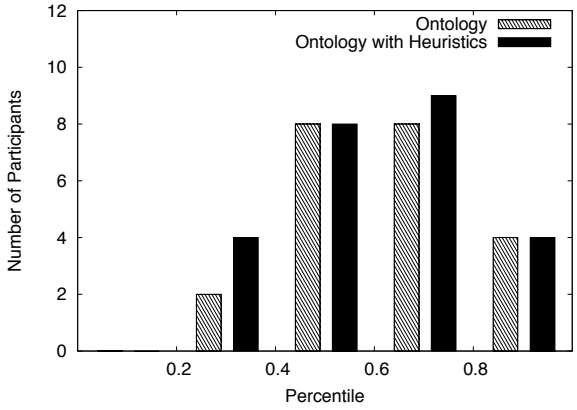

Figure 5.6: Test score distribution for two ontology conditions

The test results are itemized by question in Table 5.4, wherein the number of participants who classified each requirement as a permission (**P**), obligation (**O**), refrainment (**R**), exclusion (**E**) or none (**N**) are shown. The score per question is also shown with the correct classification designated by the colored table cells. Table 5.4 illustrates that most participants (>50%) agree with the phrase heuristics for ten of twelve requirements; for eight of these ten requirements, twice as many participants agree with the phrase heuristics. Overall, participants agreed on average with the permission (77.8%) and obligation (68.2%) concepts more often than they agreed with the refrainment (46.8%) and exclusion (32.0%) concepts. On average, participants believe 8.3% of the time that the legal requirements are not permissions, obligations, refrainments or exclusions. This proportion is significantly ($p = 0.030$) less than the 18.5% of the time that participants in the traditional practice condition classified these twelve requirements as neither in the problem or solution domains.

Table 5.4 also illustrates when participants exhibit conflicting interpretations about when a requirement is classified by certain upper ontology concepts. Two requirements, Q_1 (right to receive access) and Q_3 (right to obtain the notice), describe permitted transactions that imply obligations for the other party in the transaction. Notably, 21.3% of participants classified these two requirements as obligations. Similarly, obligation Q_7 contains the modal phrase "shall allow," which implies a permission. This obligation was classified as a permission by 30.0% of participants. However, obligation Q_5, which has the similar phrase "must allow", was overwhelmingly classified as an obligation by 87.2% of participants as compared to the 55.3% who classified Q_7 as an obligation.

Disagreement between when to apply the refrainment and exclusion concepts to legal statements

Table 5.4: Tutorial competency test results for upper ontology concepts

ID	Requirement	P	O	R	E	N	Score
Q_1	The individual *has a right* of access to their protected health information.	36	10	0	0	1	.766
Q_2	The covered entity *may* use protected health information.	44	1	1	0	1	.936
Q_3	The individual *retains the right* to obtain a copy of the privacy notice.	35	10	0	1	1	.745
Q_4	The required statements *may* be altered to reflect the fact that the notice covers more than one covered entity.	31	1	2	1	12	.660
Q_5	The product *must* allow people to interrupt, pause and restart the audio.	3	41	1	0	2	.872
Q_6	The covered entity *need only* produce the protected health information once in response to a request for access.	6	29	4	3	5	.617
Q_7	The web form *shall* allow people using assistive technology to access the information.	14	26	1	1	5	.553
Q_8	Federal agencies *cannot* claim a product is not commercially available because no product meets all the standards.	1	2	22	16	6	.468
Q_9	A material change to any term of the privacy notice *may not* be implemented prior to the effective date of the notice.	6	6	25	5	5	.532
Q_{10}	Controls and keys *shall not* require tight grasping, pinching or twisting of the wrist.	2	11	19	11	4	.404
Q_{11}	An inmate *does not have a right* to a privacy notice.	1	1	18	25	2	.532
Q_{12}	A covered entity *is not required* to agree to a restriction on disclosures.	3	1	12	28	3	.596

is evidenced in the classifications of five requirements. Refrainments Q_8 and Q_{10} exhibit a close disagreement (22-16 and 19-11, respectively), with most agreeing with the phrase heuristics and a 23-34% minority applying the exclusion concept instead of the refrainment concept. Conversely, exclusions Q_{11} and Q_{12} exhibit a close disagreement (18-25 and 12-28, respectively), with most agreeing with the phrase heuristics and a 26-38% minority applying the refrainment concept instead of the exclusion concept.

The ontology only and ontology with heuristics conditions used the markup application procedure (see Figure 2.1) to identify legal requirements from the legal text (see Figure 3.2). Section 3.2.1 describes the method to calculate the number of expected requirements acquired by the markup application procedure. This method is automated with tool support based upon the machine-parseable markup language. Among the 47 participants who completed the tutorial competency test, the markup application procedure was completed by 21 and 15 participants in the ontology

only and ontology with heuristics conditions, respectively. The ontology only condition exhibited a 4.5% non-response rate as compared to the higher 40.0% non-response rate in the ontology with heuristics conditions. Ninety percent of the non-respondents in the ontology with heuristics condition were undergraduate students.

Participants in the ontology only condition acquired 128 requirements, of which 98 requirements were expected, whereas participants in the ontology with heuristics condition acquired 92 requirements, of which 80 requirements were expected. Figures 5.7 and 5.8 show the distribution for the number of requirements acquired by participants that were expected for the ontology only and ontology with heuristics conditions, respectively. The participants in the ontology only condition exhibited a mean precision of 69.2% with variance 5.7% and mean recall of 61.0% with variance 8.0%. The participants in the ontology with heuristics condition exhibited a mean precision of 84.3% with variance 4.9% and mean recall of 76.2% with variance 8.9%. Figures 5.9 and 5.10 show the distribution for the precision and recall for the ontology only and ontology with heuristics conditions, respectively.

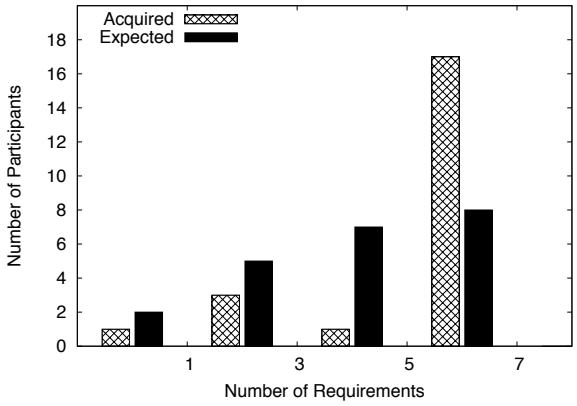

Figure 5.7: Requirements distribution for ontology

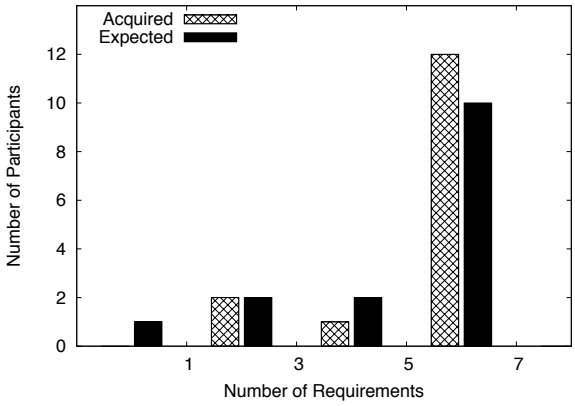

Figure 5.8: Requirements distribution for ontology with heuristics

5.4 Participant Demographics and Context

The participant population drew from baccalaureate and graduate students who were enrolled in a software engineering course at North Carolina State University. Each of the students had received a lesson in requirements engineering before participating in the experiment. The undergraduate students were working on a semester-long project to develop a HIPAA-compliant electronic medical records system. However, the system requirements and design specification did not describe, implement or otherwise support the legal requirements in the sample legal text that was used in this experiment (see Figure 3.2).

Table 5.5 presents the demographics for the participant populations in the three conditions:

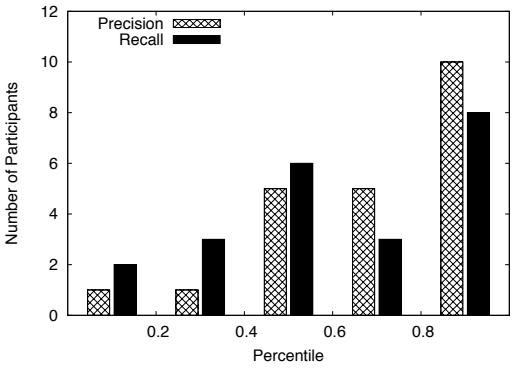

Figure 5.9: Precision and recall distribution for ontology

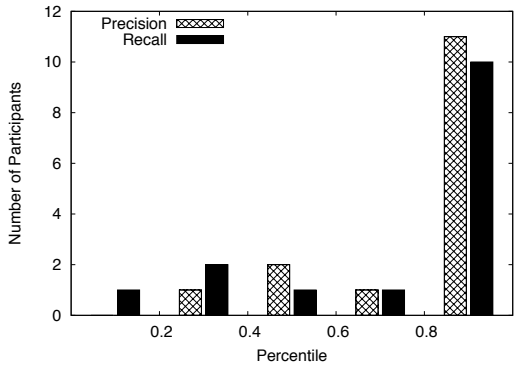

Figure 5.10: Precision and recall distribution for ontology with heuristics

traditional practice (Tradition), ontology (Ontology) and ontology with heuristics (Heuristics). The table presents the number of participants, the proportions of those students who learned English as a first language (English L1), who are studying for a bachelors, masters or Ph.D. degree, the mean age, ratio of females to males, and the number of years of industry experience, including internships. At the finish of the exercise, the ontology condition had one non-respondent and the ontology with heuristics condition had ten non-respondents. After removing these non-respondents from the demographics, there was a weakly significant difference in the proportion of participants pursuing a bachelors degree ($p = 0.076 < 0.100$) and a significant difference in the proportion of participants pursuing a masters degree ($p = 0.038 < 0.050$); no other differences were significant. Further study is needed to understand the effects, if any, of these educational differences on the reported findings.

Table 5.5: Demographics of participants in the three experimental conditions

	Tradition	**Ontology**	**Heuristics**
Number of Participants	24	22	25
Exercise Non-respondents	0	1	10
English L1	16	16	18
Pursuing Bachelors	18	17	16
Pursuing Masters	4	2	6
Pursuing Ph.D.	1	3	2
Age	22.6	22.5	23.7
Female/Male	3/19	3/19	7/17
Years of Industry Experience	1.27	2.43	1.60

Time of day has been found to affect human performance on a variety of cognitive tasks, with performance generally improving as the day progresses [66]. Others have studied time of day effects and proposed that morningness/ eveningness personality bias may cause certain individuals to

perform better earlier or later in the day, respectively, depending on their nature [83]. Figure 5.11 summarizes the time of day that participants enrolled in the experiment and began performing steps within the experimental procedure. The figure shows that participant enrollment across the three conditions is generally dispersed to reduce the probability that time of day effects would disproportionately affect any one condition. Figure 5.12 shows the mean time expended by participants in each of the three conditions; the y-error bars correspond to a 95% confidence interval. The figure shows only minor differences (e.g., under two minutes) in mean time to complete the tutorial and procedure steps, with no other significant differences between conditions.

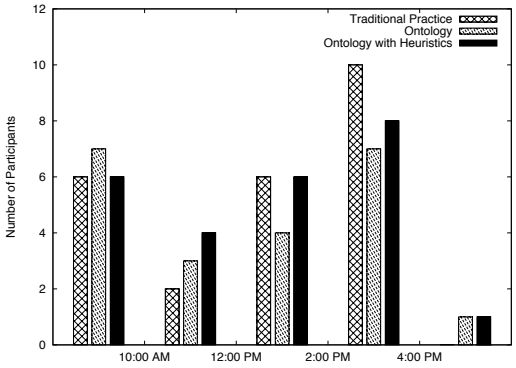

Figure 5.11: Participant enrollment, time of day

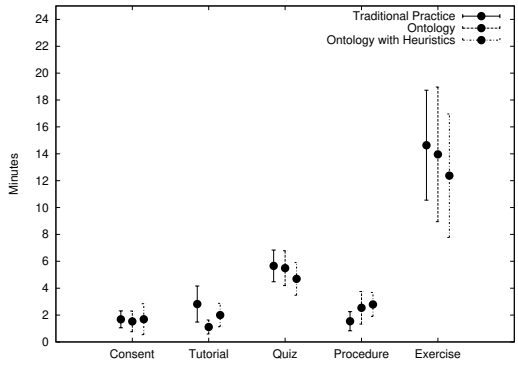

Figure 5.12: Participant time expenditure

5.5 Chapter Summary

The Frame-Based Requirements Analysis Method (FBRAM), which is formalized by the abstract model in Chapter 2, was evaluated using a human subject experiment in which participants used traditional practice or the FBRAM markup application procedure to acquire legal requirements. The experiment measured the effect of this practice and procedure on precision and recall in legal requirements acquisition. The experimental results indicate that using the ontology significantly improves precision and recall over traditional practice, whereas the phrase heuristics significantly improves precision, but not recall, over using only the ontology. These results show that traditional practice, as interpreted within the experimental design in Section 3.2, is not sufficient for legal requirements acquisition and that ontology concepts and phrase heuristics similar to those in the FBRAM can greatly improve the quality of acquired requirements. Finally, the experimental results also illustrate the nominal variance that leads engineers to create legal requirements that are inconsistent with the law.

Chapter 6

Conclusion

To map out a course of action and follow it to an end requires courage.
Ralph Waldo Emerson (1803–1882)

The challenges faced by software engineers and lawmakers in the development of legally compliant information systems are international, domain-dependent and span legislative, law enforcement, judicial and software development processes. This dissertation only narrowly examines part of this scope, the acquisition of legal requirements from U.S. federal regulations, in order to provide a basis for exploring this broader research topic. The approach in this dissertation has several contributions, including:

- A validated, systematic method consisting of a reusable document model, upper ontology, phrase heuristics and markup language for annotating and acquiring legal requirements from U.S. federal regulations with traceability;
- Four new methods for prioritizing legal requirements: balancing rights and obligations, stakeholder and product hierarchies, requirements specialization hierarchies and priority hierarchies based on explicit exceptions; and
- A suite of tools that support the method to help analysts apply the markup, detect and resolve several types of ambiguity and present frame-based and normalized legal requirements.

The empirical case study results include several techniques for identifying, classifying and interpreting legal ambiguity. The aim of these techniques is not to show which interpretation is correct, but to provide engineers and lawmakers a means to identify different, potentially conflicting interpretations of legal requirements to enable discussions about which interpretation is preferable in a given context. Because the result of applying these techniques can be formalized (e.g., the stakeholder and product hierarchy, the priority hierarchy, the cross-reference to requirement traceability matrix, etc.), it is evident that U.S. federal regulations have an implied information architecture that can be used to reduce ambiguity in the delivery of legal requirements to governed stakeholders. Whereas the case studies were conducted to acquire a grounded theory of legal requirements

acquisition, the experiments served to assess the extent to which other analysts could apply the methodology to acquire equivalent requirements. The experimental results show that the ontology, a collection of legal requirement concepts, significantly improves legal requirements coverage over traditional practice, whereas the phrase heuristics provide only weakly significant improvement over using only the ontology. However, the experimental results show that the phrase heuristics significantly improve the correct assignment of ontology concepts to legal requirements, as opposed to using only the ontology. Together, these results indicate a need for both a standard legal ontology and standard phrase heuristics to improve legal requirements acquisition. The experimental results also reveal that analysts have minor difficulty in distinguishing permissions and obligations, but that they have extensive difficulty in distinguishing between refrainments (an actor is prohibited from acting) and exclusions (an actor is not permitted, required or prohibited from acting). This result indicates that law and policy makers should define and adhere to a standard set of terminology, including phrase heuristics, to reduce this disagreement and increase the likelihood of legally appropriate behavior. This section now concludes with a discussion of the limitations of this research and future work to extend this research and to consider related problems in the area of legally compliant information systems.

6.1 Limitations

This dissertation has a limited scope and focus. The limitations discussed herein include threats to the validity of the multi-case study, remaining elements in the legal context not addressed by the Frame-Based Requirements Analysis Method (FBRAM), and gaps observed between legal requirements acquired by the FBRAM and product requirements acquired using traditional practice and expert knowledge. These gaps are relevant to requirements engineers, software developers and system administrators who will need to consider additional challenges during design, deployment, configuration and maintenance of legally compliant information systems.

6.1.1 Threats to Validity

The quality of case study research is evaluated by identifying and addressing, to the greatest extent possible, *threats to validity*. The following types of validity are discussed herein:

1. *Construct validity* is the correctness of operational measures used to collect data, build theory and report findings from the data [165], and the extent to which an observed measurement fits a theoretical construct [148].

2. *Internal validity* is the extent to which measured variables cause observable effects in the data [165].

3. *External validity* determines the scope of environmental phenomena or domain boundaries to which the theory and findings generalize [165].

4. *Reliability* describes the consistency of the theory to explain environmental phenomena and the repeatability of the operational procedures for collecting data [165].

The case study research methodology to obtain grounded theory sought to identify and thwart threats to external validity, construct validity and reliability. Internal validity is relevant only when building explanatory theory that explains underlying causes and not descriptive theories such as the abstract model presented in Chapter 2 [148, 165].

To improve construct validity, each case study was conducted by multiple researchers over multiple systematic passes using the following case study protocol to: (1) validate existing theory, by (a) identifying limitations in existing theory; and (b) reconciling contradictions between existing theory and each new dataset; and (2) expand existing theory, by observing and correctly describing all new phenomena in each case. Limitations occur when the theory does not describe some new phenomena. For example, an observed limitation in the abstract model occurred when the upper ontology did not contain a concept to describe "states" and "quality" of being, whereas these phenomena were observed in the Accessibility Standards legal text. Contradictions occur when new phenomena are contrary to an existing theoretical construct. For example, an observed contradiction occurred when the concept "condition" was assumed to mean "pre-condition," a state or event that occurs prior to some other state or event, but was used to describe durations and post-conditions, which describe states or events that occur during or after some other state or event, respectively. This contradiction is resolved by broadening the definition to cover multiple phenomena or restricting the definition to one phenomena and introducing new concepts to cover the remaining, previously unobserved phenomena.

The external validity of grounded theory is limited to the scope of the dataset by design. Consequently, grounded theory is inherently conservative about claims of scalability or generalizability to other domains. The case study results were acquired by analyzing policy goals, a consumer fact sheet and U.S. federal regulations written in the English language. To improve external validity, the researchers conducted multiple case studies that examine purposefully selected cases from which additional observations are obtained. Each additional case serves to incrementally expand the scope (i.e., variety and frequency of change) of domain phenomena that the theory claims to describe or explain. For example, although the findings of this research may generalize to U.S. and foreign state laws and international treaties, including laws written in other languages, this validity of such a claim requires additional work to analyze such laws.

Finally, to improve reliability in grounded theory, all changes to grounded theory in subsequent cases, including those to address limitations and contradictions, were validated by reviewing prior cases to ensure that the new theory continues to accurately represent the previously studied datasets.

6.1.2 Reconstructing Legal Contexts

The Frame-Based Requirements Analysis Method (FBRAM) enables software engineers to itemize legal requirements to systematically manage the complexity of legal language. This itemization is necessary to analyze legal requirements for expressed and implied rights and obligations. The FBRAM also captures explicit temporal constraints, such as pre- and post-conditions and exceptions, that are necessary to describe important business processes. However, legal texts, and U.S. federal regulations in particular, also contain implicit temporal queues that can be used to connect legal requirements into complex business processes. To study this relationship, we conducted a case study to model legal requirements using the Business Process Modeling Notation, a recent Object Modeling Group (OMG) standard with broad industry support [35].

Figure 6.1.2 illustrates a business process diagram expressed in BPMN that was constructed using the FBRAM and legal requirements from the HIPAA Privacy Rule §164.528 [35]. In BPMN, *activities* are represented by round-cornered rectangles and describe a unit of work. Activities are connected by flows, which describe the movement of information, called *message flows* that are represented by arrows with dotted lines, and the movement of time, called *sequence flows* that are represented by arrows with solid lines. *Gateways*, represented by diamonds, connect and control sequence flows using conditionally branch logic. The activities associated with a single participant (e.g., an individual) are grouped into a *pool* represented by a rectangular container spanning the horizontal length of the diagram. Pools are divided into one or more *swimlanes* that proceed chronologically from left to right and contain activities performed by the associated participant. Sequence flows pass between swimlanes, but only message flows pass between pools. Figure 6.1.2 shows only one swimlane for each participant.

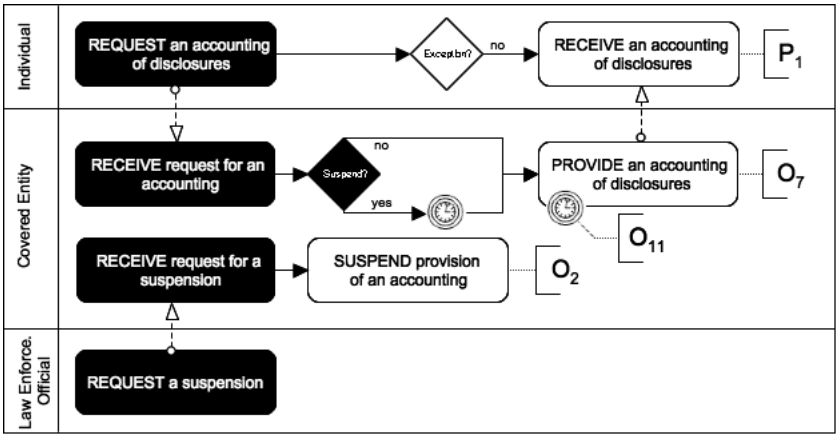

Figure 6.1: Business Process Model Acquired from HIPAA Privacy Rule

The model in Figure 6.1.2 demonstrates how explicit legal requirements (round-cornered boxes with white backgrounds) are balanced by implied legal requirements (round-cornered boxes with black backgrounds) that were discovered using the FBRAM. Separate obligations describe temporal constraints that affect when the covered entity must provide an accounting of disclosures of

protected health information to individuals (usually within 60 days). These temporal constraints consist of deadlines and temporary suspensions, depicted in Figure 6.1.2 using a gateway and timers. To coordinate these deadlines and suspensions, the analyst must identify which legal requirements are affected by these temporal constraints; a procedure described in research that extends this dissertation conducted by Breaux and Powers [35]. Although the FBRAM provides concepts in the upper ontology for capturing temporal constraints as conditions, exceptions and durations, the extended research by Breaux and Powers illustrates that legal contexts expressed in regulations require additional work to coordinate these concepts in the construction of legally compliant business processes.

6.1.3 Legal and Product Requirements Gaps

As discussed in the introduction, legal requirements are written broadly to govern multiple industries and anticipate a variety of business practices. This phenomena is realized along a requirements spectrum from system-independent stakeholder rights and obligations to system-specific product requirements. To further study this spectrum, we conducted a case study to identify gaps between Section 508 legal requirements, which govern access to information by individuals with disabilities, and real-world system requirements in use by CISCO Systems, Inc., a major manufacturer of telecommunications and network management products for the Internet [32]. In this study, we performed a *gap analysis*, which in corporate governance means a comparison between current performance and expected or desirable performance [118]. The gap analysis yielded thousands of logical assertions that formalize the types of gaps between 141 legal and 84 product requirements; these assertions were analyzed using a technique called *pattern-matching* [40]. The results include several requirements refinement patterns that constitute reasonable explanations for the types of gaps between legal and product requirements. Moreover, practitioners can use these refinement patterns to develop new product requirements from existing legal requirements in other domains. The patterns are organized in the following sub-sections under three themes [32]:

- *Clarification* – removing ambiguity by introducing relevant domain knowledge;
- *Innovation* – introducing novel product requirements to support regulatory goals; and
- *Simplification* – avoiding or removing regulatory complexity to simplify compliance.

While engineers can use these patterns to refine legal requirements into product requirements based on their unique business context, this study illustrates that more work is needed, in addition to legal requirements acquisition, in order to design compliant information systems under Section 508.

6.2 Future Work

During the study of legal requirements acquisition, several broad challenges were identified that pertain to developing legally compliant information systems. A few of these challenges are summarized here:

- *Identifying, prioritizing and integrating relevant laws, treaties and jurisdictions.* Jurisdiction affects information systems in complex ways. For example, information obtained from residents in one jurisdiction (e.g., the European Union) and maintained in another jurisdiction (e.g., the United States) may be governed by laws from both jurisdictions under international treaties. In addition, national laws can supersede state and local laws or vice versa. How can software engineers identify, prioritize, integrate and otherwise manage these complex relationships between laws and jurisdictions? Similarly, what must software engineers do to introduce existing products and systems into new jurisdictions?

- *Coordinating requirements change and the evolution of law.* Laws change over time, often to address changes in societal values or to become current with technological innovation. For example, U.S. information practices in the 1980's were predominantly paper-based and corresponding laws could not accurately foretell the volumes of electronic data that would be shared after the 1990's. Similarly, privacy and security regulations have been dramatically updated to address new social and technological challenges introduced by innovation through the Internet. How can software engineers coordinate changes in the law with innovation in technology to ensure their information systems are compliant with the latest legal requirements without stifling innovation?

- *Requirements verification; documenting and auditing evidence of compliance.* Because laws are typically written to accommodate marketplace diversity and innovation, it is difficult for regulators to be precise about how legal compliance should be measured. For example, regulators of information security law often rely on industry security standards and best practices to inform the auditing process. What types of formal verification should software engineers employ or what form of documentation should be maintained and retained for how long to demonstrate acceptable degrees of compliance assurance and system certification?

Future work includes empirical case studies and experiments to develop new methods and tools to help regulators, system developers and auditors address these complex challenges.

Bibliography

[1] A. Agresti and B. Finlay. *Statistical Methods for the Social Sciences, 3rd ed.* Prentice Hall, Upper Saddle River, New Jersey, 1997.

[2] L.E. Allen and C.S. Saxon. Computer aided normalizing and unpacking: Some interesting machine-processable transformations of legal rules. In *Computing Power and Legal Reasoning*, pages 495–572, St. Paul, Minnesota, 1984. West Publishing Company.

[3] L.E. Allen and C.S. Saxon. Better language, better thought, better communication: the a-hohfeld language for legal analysis. In *Proc. 5th International Conference on Artificial Intelligence and Law*, pages 219–228, College Park, Maryland, 1995. ACM Press.

[4] A.I. Antón. Goal-based requirements analysis. In *Proc. IEEE 2nd International Requirements Engineering Conference*, pages 136–144, York, England, 1996. IEEE Computer Society.

[5] A.I. Antón. *Goal Identification and Refinement in the Specification of Software-Based Information Systems*. PhD thesis, Georgia Institute of Technology, 1996.

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

[7] 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 and Privacy*, 2(2):36–45, 2004.

[8] A.I. Antón, E. Liang, and R.A. Rodenstein. A web-based requirements analysis tool. In *Proc. 5th Workshop on Enabling Technologies, Infrastructure for Collaborative Enterprises*, pages 238–243, Stanford, California, 1996. IEEE Computer Society.

[9] P. Ashley, S. Hada, G. Karjoth, and M. Schunter. E-p3p privacy policies and privacy authorization. In *Proc. ACM Workshop on Privacy Electronic Society*, pages 103–109, Alexandria, Virginia, 2002. ACM Press.

[10] P. Ashley, C. Powers, and M. Schunter. From privacy promises to privacy management: A new approach for enforcing privacy throughout the enterprise. In *Proc. New Security Paradigms Workshop*, pages 43–50, Virginia Beach, Virginia, 2002. ACM Press.

[11] F. Baader, D. Calvanese, D. McGuiness, D. Nardi, and P.F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge, England, 2002.

[12] A.K. Bandara, E.C. Lupu, J. Moffett, and A. Russo. A goal-based approach to policy refinement. In *Proc. IEEE 5th Workshop on Policies for Distributed Systems and Networks*, pages 229–239, Yorktown Heights, New York, 2004. IEEE Computer Society.

[13] D.L. Baumer and J.C. Poindexter. *Legal Environment of Business: In the Information Age*. McGraw-Hill Irwin, New York, New York, 2004.

[14] T. J. M. Bench-Capon. Deep models, normative reasoning and legal expert systems. In *Proc. 2nd International Conference on Artificial Intelligence and Law*, pages 37–45, Vancouver, British Columbia, Canada, 1989. ACM Press.

[15] T. J. M. Bench-Capon, G. O. Robinson, T. W. Routen, and M. J. Sergot. Logic programming for large scale applications in law: A formalisation of supplementary benefit legislation. In *Proc. 1st International Conference on Artificial Intelligence and Law*, pages 190–198, Boston, Massachusetts, 1987. ACM Press.

[16] D.M. Berry and E. Kamsties. Syntactically dangerous all and plural specifications. *IEEE Software*, 22(1):55–57, 2005.

[17] C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, and C. Soria. Automatic semantics extraction in law documents. In *Proc. 10th International Conference on Artificial Intelligence and Law*, pages 133–140, Bologna, Italy, 2005. ACM Press.

[18] C. Biagioli, P. Mariani, and D. Tiscornia. ESPLEX: A rule and conceptual model for representing statutes. In *Proc. 1st International Conference on Artificial Intelligence and Law*, pages 240–251, Boston, Massachusetts, 1987. ACM Press.

[19] P.V. Biron and A. Malhotra. XML Schema Part 2: Datatypes Second Edition. W3C Recommendation, October 2004.

[20] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995.

[21] G. Boella and L. van der Torre. Permissions and obligations in hierarchical normative systems. In *Proc. 9th International Conference on Artificial Intelligence and Law*, pages 109–118, Edinburgh, Scotland, 2003. ACM Press.

[22] S. Borkin. The HIPAA final security standards and ISO/IEC 17799. In *Collect. Information Security Reading Room*. SANS Institute, July 2003.

[23] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, F. Yergeau, and J. Cowan. Extensible Markup Language (XML) 1.1 (Second Edition). W3C Recommendation, August 2006.

[24] T.D. Breaux. Exercising due diligence in legal requirements acquisition: A tool-supported, frame-based approach. In *Proc. IEEE 17th International Requirements Engineering Conference*, Atlanta, Georgia, 2009.

[25] T.D. Breaux and A.I. Antón an E.H. Spafford. A distributed requirements management framework for compliance and accountability. *Elsevier Computers and Security*, 28(1–2):8–17, 2009.

[26] T.D. Breaux and A.I. Antón. Analyzing goal semantics for rights, permissions and obligations. In *Proc. IEEE 13th International Requirements Engineering Conference*, pages 177–186, Paris, France, 2005. IEEE Computer Society.

[27] T.D. Breaux and A.I. Antón. Deriving semantic models from privacy policies. In *Proc. IEEE 6th International Workshop on Policies for Distributed Systems and Networks*, pages 67–76, Stockholm, Sweden, 2005. IEEE Computer Society.

[28] T.D. Breaux and A.I. Antón. Mining rule semantics to understand legislative compliance. In *Proc. ACM Workshop on Privacy in Electronic Society*, pages 51–54, Alexandria, Virginia, 2005. ACM Press.

[29] T.D. Breaux and A.I. Antón. Impalpable constraints: Framing requirements for formal methods. Technical Report Technical Report TR-2006-06, Department of Computer Science, North Carolina State University, Raleigh, North Carolina, February 2007.

[30] T.D. Breaux and A.I. Antón. A systematic method for acquiring regulatory requirements: A frame-based approach. In *Proc. 6th International Workshop on Requirements for High Assurance Systems*, Dehli, India, September 2007. Software Engineering Institute (SEI).

[31] T.D. Breaux and A.I. Antón. Analyzing regulatory rules for privacy and security requirements. *IEEE Transactions on Software Engineering*, 34(1):5–20, 2008.

[32] T.D. Breaux, A.I. Antón, K. Boucher, and M. Dorfman. Legal requirements, compliance and practice: An industry case study in accessibility. In *Proc. IEEE 16th International Requirements Engeering Conference*, pages 43–52, Barcelona, Spain, 2008. IEEE Computer Society.

[33] T.D. Breaux, A.I. Antón, and J. Doyle. Semantic parameterization: A process for modeling domain descriptions. *ACM Transactions on Software Engineering and Methodology*, 18(2):5, 2008.

[34] T.D. Breaux, A.I. Antón, Clare-Marie Karat, and John Karat. Enforceabiltiy vs. accountability in electronic policies. In *Proc. IEEE 7th International Workshop on Policies for*

Distributed Systems and Networks, pages 227–230, London, Ontario, 2006. IEEE Computer Society.

[35] T.D. Breaux and C. Powers. Early studies in acquiring evidentiary, reusable business process models from laws. In *Proc. 6th International Conference on Information Technology: New Generations*, Las, Vegas, 2009. IEEE Computer Society.

[36] T.D. Breaux, M.W. Vail, and A.I. Antón. Towards compliance: Extracting rights and obligations to align requirements with regulations. In *Proc. IEEE 14th International Requirements Engineering Conference*, pages 46–55, Minneapolis, Minnesota, 2006. IEEE Computer Society.

[37] Bureau of Labor Statistics, U.S. Department of Labor, Career Guide to Industries, 2008-09 Edition, Health Care.

[38] Bureau of Transportation Statistics, U.S. Department of Transportation, Air Carriers: T-100 Domestic Market (All Carriers), 2007-2008.

[39] J-W. Byon, E. Bertino, and N. Li. Purpose-based access control of complex data for privacy protection. In *10th ACM Symposium on Access Control Models and Technologies*, pages 102–110, Stockholm, Sweden, 2005. ACM Press.

[40] D.T. Campbell. *Pattern matching as an essential in distal knowing*, pages 81–106. Holt, Rinehart, Winston, New York, New York, 1966.

[41] P.P-S. Chen. English sentence structure and entity-relationship diagrams. *Information Sciences*, 29(2-3):127–149, 1983.

[42] L. Cholvy. Checking regulation consistency by using sol-resolution. In *Proc. 7th International Conference on Artificial Intelligence and Law*, pages 73–79, Oslo, Norway, 1999. ACM Press.

[43] CIO Insight. 2005 regulatory compliance survey, May 2005.

[44] R.L. Cobleigh, G.S. Avrunin, and L.A. Clarke. User guidance for creating precise and accessible property specifications. In *Proc. ACM SIGSOFT 14th International Symposium on Foundations of Software Engineering*, pages 208–218, Portland, Oregon, 2005.

[45] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.

[46] L. Cranor, B. Dobbs, S. Egelman, G. Hogben, J. Humphrey, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle, M. Schunter, D.A. Stampley, and R. Wenning. The Platform for Privacy Preferences 1.1. (P3P1.1) Specification. W3C Working Group Note, November 2006.

[47] A. Cregan, R. Schwitter, and T. Meyer. Sydney OWL Syntax: Towards a controlled natural language syntax for OWL 1.1. In *Proc. 3rd OWL: Experiences and Directions Workshop*, volume 258, Tilburg, Netherlands, 2007. CEUR-Workshop Proceedings.

[48] J.W. Creswell. *Research Design: Qualitative, Quantitative and Mixed Methods Approaches, 2nd ed.* Sage Publications, 2003.

[49] CSO Magazine. Chief security officers reveal business continuity, resiliency and disaster recovery as the top business concern in 2006: In stark contrast, CSO Magazine survey finds CSOs investing in compliance, not recovery, March 2006.

[50] L.M. Cysneiros and J.C.S.P. Leite. Nonfunctional requirements: From elicitation to conceptual models. *IEEE Transactions on Knowledge and Data Engineering*, 30(5):328–350, 2004.

[51] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20:3–50, 1993.

[52] C. Denger, D.M. Berry, and E. Kamsties. Higher quality requirements specifications through natural language patterns. In *Proc. IEEE International Conference on Software: Science, Technology and Engineering*, pages 80–90, Herzelia, Israel, 2003. IEEE Computer Society.

[53] N. Dinesh, A. Joshi, I. Lee, and B. Webber. Extracting formal specifications from natural language regulatory documents. In *Proc. 5th International Workshop on Inference in Computational Semantics*, Buxton, England, 2006.

[54] Ernst and Young. Global information security survey 2005: Report on the widening gap, 2005.

[55] Ernst and Young. Global information security survey 2006: Achieving success in a globalized world, 2006.

[56] Ernst and Young. 10th annual global information security survey: Achieving a balance of risk and performance, 2007.

[57] C.B. Farrell. Choicepoint settles data security breach charges; to pay $10 million in civil penalties and $5 million for customer redress. Technical Report FTC File No. 052-3069, Office of Public Affairs, U.S. Federal Trade Commission, 2006.

[58] Federal Aviation Administration, U.S. Department of Transportation, Extended Operations (ETOPS) of Multi-Engine Airplanes; Final Rule, 14 CFR Parts 1, 21, 25, 33, 121, and 135, *Federal Register*, 72(9): 1808-1887, January 16, 2007.

[59] Federal Aviation Administration, U.S. Department of Transportation, Extended Operations (ETOPS) of Multi-Engine Airplanes; Final Rule, correction, 14 CFR Parts 1, 121, and 135, *Federal Register*, 72(31): 7346-7348, February 15, 2007.

[60] Federal Aviation Administration, U.S. Department of Transportation, Extended Operations (ETOPS) of Multi-Engine Airplanes; Final Rule, correction, 14 CFR Parts 121 and 135, *Federal Register*, 72(90): 26540-26542, May 10, 2007.

[61] Federal Aviation Administration, U.S. Department of Transportation, Extended Operations (ETOPS) of Multi-Engine Airplanes; Final Rule, delay of compliance dates, 14 CFR 135, *Federal Register*, 73(32): 8796-8798, February 15, 2008.

[62] Federal Aviation Administration, U.S. Department of Transportation, Extended Operations (ETOPS) of Multi-Engine Airplanes; Final Rule, immediately adopted, 14 CFR 121 and 135, *Federal Register*, 73(116): 33879-33882, June 16, 2008.

[63] C.J. Fillmore. The case for case. In E. Bach and R.T. Harms, editors, *Universals in Linguistic Theory*. Holt, Rhinehart and Winston, New York, New York, 1968.

[64] R.A. Fisher. *Design of Experiments, 9th ed.* Hafner Press, London, England, 1971.

[65] J.L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[66] L. Fowler. *Time of Day Effects on Cognitive Processing and Brain Temperature.* PhD thesis, Georgia State University, 2000.

[67] A. Fuxman, L. Liu, J. Mylopoulos, M. Pistore, M. Roveri, and P. Traverso. Specifying and analyzing early requirements in tropos. *Requirements Engineering*, 9(2):132–50, 2004.

[68] B.A. Garner, editor. *Blacks Law Dictionary, 8th ed.* Thompson West, St. Paul, Minnesota, 2004.

[69] D.C. Gause and G.M. Weinberg. *Exploring Requirements: Quality Before Design*. Dorset House Publishing, New York, New York, 1989.

[70] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Modeling security requirements through ownership, permissions and delegation. In *Proc. IEEE 13th International Requirements Engineering Conference*, pages 167–176, Paris, France, 2005. IEEE Computer Society.

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

[72] L. Goldin and D.M. Berry. Abstfinder, a prototype abstraction finder for natural language text for use in requirements elicitation: design, methodology, and evaluation. In *Proc. 1st International Requirements Engineering Conference*, pages 84–93, Colorado Springs, Colorado, 1994. IEEE Computer Society.

[73] L. Goldin and D.M. Berry. AbstFinder: A prototype natural language text abstraction finder for use in requirements elicitation. *Automated Software Engineering*, 4(4):375–412, 1997.

[74] O. Gotel and A. Finkelstein. Contribution structures. In *Proc. 2nd IEEE International Symposium on Requirements Engineering*, pages 100–107, San Diego, California, 1995. IEEE Computer Society.

[75] O. Gotel and A. Finkelstein. Extended requirements traceability: results of an industrial case study. In *Proc. 3rd IEEE International Symposium on Requirements Engineering*, pages 169–178, Annapolis, Maryland, 1997. IEEE Computer Society.

[76] Object Management Group. OMG Unified Modeling Language (OMG UML), Infrastructure Version 2.1.2, November 2007.

[77] J. Gruber. *Lexical structure in syntax and semantics*. North Holland, New York, New York, 1976.

[78] R. Harwell, E. Aslaksen, I. Hooks, R. Mengot, and K. Ptack. *What Is A Requirement?*, pages 23–29. IEEE Computer Society, Washington, D.C., 1997.

[79] J. Hash, P. Bowen, A. Johnson, C.D. Smith, and D.I. Steinberg. An introductory resource guide for implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule. Technical Report Special Publication 800-66, U.S. National Institute of Standards and Technology, March 2005.

[80] W.N. Hohfeld. Some fundamental legal conceptions as applied in judicial reasoning. *The Yale Law Journal*, 23(1):16–59, 1913.

[81] W.N. Hohfeld. Fundamental legal conceptions as applied in judicial reasoning. *The Yale Law Journal*, 26(8):710–770, 1917.

[82] C.J. Hoofnagle and D.J. Solove. Re: Request for investigation into data broker products for compliance with the FCRA. Technical report, Electronic Privacy Information Center, Washington, D.C., 2004.

[83] J. A. Horne, C. G. Brass, and A. N. Petitt. Circadian performance differences between morning and evening 'types'. *Ergonomics*, 23(1):29–36, 1980.

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

[85] E. Hull, K. Jackson, and J. Dick. *Requirements Engineering, 2nd ed.* Springer, London, England, 2005.

[86] IEEE Std. 1061-1998, IEEE Standard for a Software Quality Metrics Methodology.

[87] IEEE Std. 830-1993, IEEE Recommended Practice for Software Requirements Specifications.

[88] Information Technology Governance Institute, Control Objectives for Information and related Technology (COBIT), Version 4.1, 2007.

[89] ISO/IEC 14977:1996. Information technology – Syntactic metalanguage – Extended BNF, 1996.

[90] ISO/IEC 15408:2005. Information technology – Security techniques – Evaluation criteria for IT security, 2005.

[91] ISO/IEC 17799:2005. Information technology - Security techniques - Code of practice for information security management, 2005.

[92] M. Jackson. *Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices*. Addison-Wesley, New York, New York, 1995.

[93] M. Jackson. The world and the machine. In *Proc. 17th IEEE International Conference on Software Engineering*, pages 283–292, Seattle, Washington, 1995. IEEE Computer Society.

[94] M. Jackson and P. Zave. Domain descriptions. In *Proc. IEEE 1st International Symposium on Requirements Engineering*, pages 56–64, San Diego, California, 1993. IEEE Computer Society.

[95] L. Kagal. *A Policy-Based Approach to Governing Autonomous Behavior in Distributed Environments*. PhD thesis, University of Maryland Baltimore County, 2004.

[96] L. Kagal, T. Finin, and A. Joshi. A policy language for a pervasive computing environment. In *Proc. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, pages 63–74, Lake Como, Italy, 2003. IEEE Computer Society.

[97] H. Kaindl. How to identify binary relationships for domain models. In *Proc. IEEE 18th International Conference on Software Engineering*, pages 28–36, Berlin, Germany, 1996. IEEE Computer Society.

[98] K. Kaljurand and N.E. Fuchs. Verbalizing OWL in Attempto Controlled English. In *Proc. 3rd OWL: Experiences and Directions Workshop*, volume 258, Tilburg, Netherlands, 2007. CEUR-Workshop Proceedings.

[99] E. Kamsties. *Understanding Ambiguity in Requirements Engineering*, pages 245–266. Springer, The Netherlands, 2006.

[100] P. Keevil. Representing the building regulations in frame-based format. In *Proc. IEE Colloquium on Knowledge-Based Approaches to Automation in Construction*, pages 4/1–4/3, London, England, 1995. Institute of Engineering and Technology.

[101] S.L. Kerrigan. *A Software Infrastructure for Regulatory Information Management and Compliance Assistance*. PhD thesis, Stanford University, 2003.

[102] S.L. Kerrigan and K.H. Law. Logic-based regulation compliance-assistance. In *Proc. 9th International Conference on Artificial Intelligence and Law*, pages 126–135, Edinburgh, Scotland, 2003. ACM Press.

[103] N. Kiyavitskaya, N. Zeni, T.D. Breaux, A.I. Antón, J. Gordy, L. Mich, and J. Mylopoulos. Automating the extraction of rights and obligations from regulations. In *Proc. International Conference on Conceptual Modeling*, pages 154–168, Barcelona, Spain, 2008. Springer.

[104] N. Kiyavitskaya, N. Zeni, L. Mich, T.D. Breaux, A.I. Antón, and J. Mylopoulos. Extracting rights and obligations from regulations: Towards a tool-supported process. In *Proc. IEEE/ACM 22nd International Conference on Automated Software Engineering*, pages 429–432, Atlanta, Georgia, 2007. IEEE Computer Society.

[105] T. Koch, C. Krell, and B. Kramer. Policy definition language for automated management of distributed systems. In *Proc. IEEE 2nd International Workshop on Systems Management*, pages 55–64, Toronto, Ontario, 1996. IEEE Computer Society.

[106] S. Konrad and B.H.C. Cheng. Real-time specification patterns. In *Proc. IEEE 27th International Conference on Software Engineering*, pages 372–381, St. Louis, Missouri, 2005. IEEE Computer Society.

[107] S-W. Lee, R. Gandhi, D. Muthurajan, D. Yavagal, and G-J. Ahn. Building problem domain ontology from security requirements in regulatory documents. In *Proc. International Workshop Software Engineering Secure Systems*, pages 43–50, Shanghai, China, 2006. IEEE Computer Society.

[108] S. Liaskos, A. Lapouchnian, Y. Yu, E. Yu, and J. Mylopoulos. On goal variability acquisition and analysis. In *Proc. IEEE 14th International Requirements Engineering Conference*, pages 76–85, Minneapolos, Minnesota, 2006. IEEE Computer Society.

[109] D. Lipkin, J. Marsh, H. Thompson, N. Walsh, and S. Zilles. eXtensible Schema Language Transformations (XSLT) Version 1.0. W3C Recommendation, November 1999.

[110] J. Lobo, R. Bhatia, and S. Naqvi. A policy description language. In *Proc. 16th National Conference on Artificial Intelligence and 11th Innovative Applications of Artificial Intelligence Conference*, pages 291–298, Orlando, Florida, 1999. American Association for Artificial Intelligence.

[111] E. Lupu, M. Sloman, N. Dulay, and N. Damianou. Ponder: Realising enterprise viewpoint concepts. In *Proc. 4th International Conference on Enterprise Distributed Object Computing*, pages 66–75, Makuhari, Japan, 2000. IEEE Computer Society.

[112] J. Martinek and J. Cybulka. Dynamics of legal provisions and its representation. In *Proc. 10th International Conference on Artificial Intelligence and Law*, pages 20–24, Bologna, Italy, 2005. ACM Press.

[113] M.J. May, C.A. Gunter, and I. Lee. Privacy APIs: Access control techniques to analyze and verify legal privacy policies. In *Proc. IEEE 19th Computer Security Foundations Workshop*, pages 85–97, Venice, Italy, 2006. IEEE Computer Society.

[114] L.T. McCarty. The TAXMAN project: Towards a cognitive theory of legal argument. In *Proc. Advanced Workshop on Computer Science and Law*, pages 23–43, Swansea, Wales, September 1979.

[115] D.L. McGuinness and F. van Harmelen. OWL Web Ontology Language. W3C Recommendation, February 2004.

[116] Rebecca T. Mercuri. The HIPAA-potamus in health care data security. *Communications of the ACM*, 47(7):25–28, 2004.

[117] M. Mernik, J. Heering, and A.M. Sloane. When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344, 2005.

[118] M. Shirom M.I. Harrison. *Organizational Diagnosis and Assessment: Bridging Theory and Practice*. Sage Publications, 1999.

[119] M. Minsky. A framework for representing knowledge. In P. Wilson, editor, *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill, 1975.

[120] N.H. Minsky and V. Ungureanu. Law-governed interaction: a coordination and control mechanism for heterogeneous distributed systems. *ACM Transactions Software Engineering and Methodology*, 9(3):273–305, 2000.

[121] Jonathan D. Moffett and Morris S. Sloman. The representation of policies as system objects. In *Proc. Conference on Organizational Computing Systems*, pages 171–184, Atlanta, Georgia, 1991. ACM Press.

[122] T. Moses. eXtensible Access Control Markup Language (XACML) Version 2.0. Oasis Standard, February 2005.

[123] J. Mylopoulos, L. Chung, and B. Nixon. Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Transactions on Software Engineering*, 18(6):483–497, 1992.

[124] J. Mylopoulos, L. Chung, and E. Yu. From object-oriented to goal-oriented requirements analysis. *Communications of the ACM*, 42(1):31–37, 1999.

[125] Office of Civil Rights, U.S. Department of Health and Human Services, Standards for Privacy of Individually Identifiable Health Information; Final rule, 45 CFR Part 160, Part 164, *Federal Register*, 65(250): 82462-82829, December 28, 2000.

[126] Office of Civil Rights, U.S. Department of Health and Human Services, Standards for Privacy of Individually Identifiable Health Information; Technical corrections to final rule, 45 CFR Part 160, Part 164, *Federal Register*, 65(251): 82944, December 29, 2000.

[127] Office of Civil Rights, U.S. Department of Health and Human Services, Standards for Privacy of Individually Identifiable Health Information; Correction of effective and compliance dates, 45 CFR Part 160, Part 164; final rule. *Federal Register*, 66(38):12434, February 26, 2001.

[128] Office of Civil Rights, U.S. Department of Health and Human Services, Standards for Privacy of Individually Identifiable Health Information; Correction of effective and compliance dates, 45 CFR Part 160, Part 164; final rule. *Federal Register*, 67(157):53182-53273, August 14, 2002.

[129] Office of Civil Rights, U.S. Department of Health and Human Services, Fact Sheet: Protecting the Privacy of Patients Health Information, April 14, 2003.

[130] Office of Management and Budget, Office of Information and Regulatory Affairs, Stimulating Smarter Regulation: 2002 Report to Congress on the Costs and Benefits of Regulations and Unfunded Mandates on State, Local and Tribal Entities, December 2002.

[131] Office of the Federal Register, U.S. National Archives and Records Administration, Federal Register Document Drafting Handbook, October 1998.

[132] A. Ohnishi and C. Potts. Grounding scenarios in frame-based action semantics. In *Proc. 7th Workshop on Requirements Foundations for Software Quality*, pages 177–182, Interlaken, Switzerland, 2001.

[133] P.N. Otto, A.I. Antón, and D.L. Baumer. The choicepoint dilemma: How data brokers should handle the privacy of personal information. *IEEE Security and Privacy*, 5(5):15–23, 2007.

[134] S.P. Overmyer, B. Lavoie, and O. Rambow. Conceptual modeling through linguistic analysis using LIDA. In *Proc. IEEE 23rd International Conference on Software Engineering*, pages 401–410, Toronto, Ontario, 2001. IEEE Computer Society.

[135] J. Park and R. Sandhu. The UCON$_{ABC}$ usage control model. *ACM Transactions on Information and System Security*, 7(1):128–174, 2004.

[136] S. Pemberton, D. Austin, J. Axelsson, T. elik, D. Dominiak, H. Elenbaas, B. Epperson, M. Ishikawa, S. Matsui, S. McCarron, A. Navarro, S. Peruvemba, R. Relyea, S. Schnitzenbaumer, and P. Stark. XHTML 1.0 The Extensible HyperText Markup Language (Second Edition). W3C Recommendation, August 2002.

[137] J.R. Pleis and M. Lethbridge-ejku. Summary health statistics for U.S. adults: national health interview survey, 2005. *Vital and Health Statistics*, 10(232), 2006.

[138] K.R. Popper. *The Logic of Scientific Inquiry*. Taylor and Francis, London, England, 2002.

[139] C. Potts, K. Takahashi, and A. I. Antón. Inquiry-based requirements analysis. *IEEE Software*, 11(2):21–32, 1994.

[140] C. Powers, Personal communication, December 2, 2005.

[141] Radio Technical Commission for Aerospace, DO-178b, Software Considerations for Airborne Systems and Equipment Certification, December 1, 1992.

[142] Radio Technical Commission for Aerospace, DO-248b, Final Report for Clarification of DO-178B "Software Considerations in Airborne Systems and Equipment Certification," October 12, 2001.

[143] H.B. Reubenstein and R.C. Waters. Requirements apprentice: Automated assistance for requriements acquisition. *IEEE Transactions on Software Engineering*, 17(3):226–240, 1991.

[144] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

[145] R.C. Schank and R.P. Abelson. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Discovery*. Lawrence Erlbaum Assoc., Hillsdale, New Jersey, 1977.

[146] N. Schwarz and H-J. Hippler. *Response Alternatives: The Impact of Their Choice and Presentation Order*, pages 41–56. John Wiley and Sons, Ltd., West Sussex, England, 1991.

[147] M.J. Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond, and H.T. Cory. The British Nationality Act as a logic program. *Communications of the ACM*, 29(5):370–386, May 1986.

[148] W.R. Shadish, T.D. Cook, and D.T. Campbell. *Experimental and Quasi-experimental Designs for Generalized Causal Inference*. Houghton-Mifflin Company, Boston, Massachusetts, 2002.

[149] D.M. Sherman. A Prolog model of the Income Tax Act of Canada. In *Proc. 1st International Conference on Artificial Intelligence and Law*, pages 127–136, Boston, Massachusetts, 1987. ACM Press.

[150] M. Sipser. *Regular Languages*, pages 31–90. Course Technology, Boston, Massachusetts, 2005.

[151] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, 2007.

[152] R.L. Smith, G.S. Avrunin, L.A. Clarke, and L.J. Osterweil. PROPEL: An approach supporting property elucidation. In *Proc. IEEE 24th International Conference on Software Engineering*, pages 11–21, Orlando, Florida, 2002. IEEE Computer Society.

[153] I. Sommerville and P. Sawyer. *Requirements Engineering: A Good Practice Guide*. John Wiley and Sons, Ltd., West Sussex, England, 1997.

[154] R.K. Stamper. LEGOL: Modelling legal rules by computer. In *Proc. Advanced Workshop on Computer Science and Law*, pages 45–71, Swansea, United Kingdom, September 1979.

[155] H.S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn. XML Schema Part 1: Structures Second Edition. W3C Recommendation, October 2004.

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

[157] U.S. Access Board, Electronic and Information Technology Accessibility Standards; Final rule, 36 CFR 1194, *Federal Register*, 65(246):80500-80528, December 21, 2000.

[158] T.M. van Engers and M.R. Boekenoogen. Improving legal quality: an application report. In *Proc. 9th International Conference on Artificial Intelligence and Law*, pages 284–292, Edinburgh, Scotland, 2003. ACM Press.

[159] T.M. van Engers, I.R. Gerrits, M.R. Boekenoogen, I.E. Glassée, and P. Kordelaar. POWER: Using UML/OCL for modeling legislation - an application report. In *Proc. International Conference on Artificial Intelligence and Law*, pages 157–167, St. Louis, Missouri, 2001. ACM Press.

[160] K.S. Wasson. A case study in systematic improvement of language for requirements. In *Proc. IEEE 14th International Requirements Engineering Conference*, pages 6–15, Minneapolis, Minnesota, 2006. IEEE Computer Society.

[161] K.S. Wasson, J.C. Knight, E.A. Strunk, and S.R. Travis. Tools supporting the communication of critical domain knowledge in high-consequence systems development. In *Proc. 22nd International Conference on Computer Safety, Reliability and Security*, volume 2788, pages 317–330, Berlin, Germany, 2003. Springer.

[162] R.J. Wieringa. *Requirements Engineering: Frameworks for Understanding*. John Wiley and Sons, Ltd., West Sussex, England, 1996.

[163] W. Wilkinson. The office for civil rights and health care privacy. In *Proc. 12th National HIPAA Summit*, Washington, D.C., 2006. Health Care Conference Administrators.

[164] W.M. Wilson, L.H. Rosenberg, and L.E. Hyatt. Automated analysis of requirement specifications. In *Proc. IEEE 19th International Conference on Software Engineering*, pages 161–171, Boston, Massachusetts, 1997. IEEE Computer Society.

[165] R.K. Yin. Case study research, 3rd ed. In *Applied Social Research Methods Series, v.5*. Sage Publications, 2003.

[166] E. Yu. Towards modeling and reasoning support for early-phase requirements engineering. In *Proc. IEEE 3rd International Symposium on Requirements Engineering*, pages 226–235, Annapolis, Maryland, 1997. IEEE Computer Society.

[167] P. Zave and M. Jackson. Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology*, 6(1):1–30, 1997.

APPENDICES

Appendix A

Acts of United States Congress

Administrative Procedures Act (APA) of 1941, 5 U.S.C. Chapter 5, governs United States federal and independent agencies in the acts of rulemaking and judicial oversight.

Fair Credit Reporting Act (FCRA), 15 U.S.C. §1681, governs credit reporting procedures in the United States.

Federal Aviation Act of 1958, 49 U.S.C. §44701, etc., establishes the U.S. Federal Aviation Administration and rulemaking authority.

Health Insurance Portability and Accountability Act (HIPAA) of 1996, 42 U.S.C. §1301, etc., governs the security and privacy of electronic medical records in the United States.

Rehabilitation Act Amendments of 1986 and 1998, 29 U.S.C. §794d, govern the accessibility of federal information by individuals with disabilities in the United States.

Appendix B

The Frame-based Markup Grammar

The frame markup language is formalized by a context-free grammar (CFG), presented below in Extended Bachus-Naur–Form (EBNF) [89] with regular expression operators *, +, and ? to denote lists of zero or more, one or more, or zero or one, respectively. The frame markup is used to extract legal requirements from regulations, as discussed in Section 2.4. The capitalized words and brackets are terminal symbols: in practice, AND is ampersand (&), OR is vertical bar (|), COPY is slash (/), CUT is backslash (\), PASTE is asterisk (*), HASH is the number sign (#), NUMBER is one or more digits 0-9 and TEXT is any printable character except for [,], {, }, &, |.

```
<s>         := <frame>*
<frame>     := (<pattern> | <value> | TEXT)+
<pattern>   := { <slot> }
<value>     := [ <slot> ]
<slot>      := <op>? <frame> <alt>* | PASTE NUMBER
<alt>       := AND <slot> | OR <slot>
<op>        := HASH TYPE | (COPY | CUT) NUMBER
```

Appendix C

The Document Model XML Schema

The document model is formalized and expressed in the W3C eXtensible Schema Language (XSL) []. As discussed in Section 2.1, the document model is used by an engineer to divide a legal text into divisions: each division has at most one division index that is used to reference the division throughout the document and at most one title that describes the content of the division. The string values assigned to the division index and title will appear in the actual document text. The XML Schema for the document model follows:

```
<?xml version="1.0"?>
<schema targetNamespace="fbram" xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:fbram="fbram" elementFormDefault="qualified">

    <element name="document" type="fbram:Document" />
    <complexType name="Document" mixed="true">
        <annotation>
            <documentation xml:lang="en">
                documents contain one or more divisions mixed with text
            </documentation>
        </annotation>
        <sequence>
            <element name="division" type="fbram:Division"
                minOccurs="0" maxOccurs="unbounded" />
        </sequence>
    </complexType>
    <complexType name="Division" mixed="true">
        <annotation>
            <documentation xml:lang="en">
                divisions contain one or more subdivisions mixed with text
                and at most one division index and title
```

```
            </documentation>
        </annotation>
        <sequence>
            <element name="index" type="string" minOccurs="0" maxOccurs="1" />
            <element name="title" type="string" minOccurs="0" maxOccurs="1" />
            <element name="division" type="fbram:Division"
                minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
</schema>
```

Appendix D

Transaction and Delegation Verbs

Table D.1 contains the transaction and delegation verbs identified in the Practices, Safety and Accessibility case studies.

Table D.1: Transaction and Delegation Verbs

Practices	**Safety**	**Accessibility**
append	accept	adjust
condition	compare	avail
disclose	report	equip
discuss	submit	provide
give		reduce
include		synchronize
inform		
link		
make		
notify		
obtain		
post		
provide		
refer		
transmit		

Appendix E

Qualitative Requirements Metrics

The requirements comparison metrics direct the investigators focus to evaluate subtle differences in meaning between two requirements. Unlike software quality metrics that yield numerical measurements [86], these metrics yield nominal measurements in the form of logical assertions. The metrics are introduced with examples in an exploratory case study on Section 508 [32]. The metrics are presented below in the context of comparing two arbitrary requirements, A and B.

E.1 Statement Metrics

The statement metrics are used to compare two requirement statements, avoiding a detailed evaluation of differences between specific phrases or words in those statements. The metrics are:

> **Metric S-G (Goal):** Requirement A describes "why" Requirement B should be implemented.
>
> **Metric S-R (Refinement):** Requirement A describes "how" Requirement B should be implemented.
>
> **Metric S-E (Equivalent):** Requirements A and B are equivalent, with some portions of the requirements describing the same or a similar action.
>
> **Metric S-C (Conflict):** Requirement A and B potentially or directly conflict with each other.

Metrics S-G and S-R are similar to the concept of *goal refinement* in goal-oriented requirements acquisition [51]. Table E.1 presents an example from the Section 508 case study that shows the above statement metrics applied to obligation NCSU O-29 (Requirement A) and three Cisco requirements (Requirements B). The original text from Section 508 §1194.21(f) in the Accessibility Standards from which NCSU O-29 was acquired appears below:

> **1194.21(f):** Textual information shall be provided through operating system functions for displaying text.

NCSU O-29: PROVIDE textual information through operating system functions for displaying text.

Cisco SW-50.11: Provide text in a manner compatible with assistive technology.

Cisco SW-50.11 (M2): Draw text using the standard function calls; this applies to text in the client area as well as text in custom controls

Cisco SW-50.11 (M3): Use standard functions to copy or erase text and graphics.

Table E.1: Example application of statement-level metrics

NCSU Requirement	Cisco Requirement	Metric
O-29	SW-50.11	S-G
O-29	SW-50.11 (M2)	S-E
O-29	SW-50.11 (M3)	S-R

E.2 Phrase Metrics

The phrase metrics are used to compare discrete phrases in two requirements. These metrics are used in conjunction with one of the above statement metrics to further clarify the similarity or difference in a requirements meaning. They are:

Metric P-G1 (Generalized Concept): The "phrase in B" describes a more general concept than the "phrase in A."

Metric P-G2 (Missing Constraint): The "phrase in A" is missing from Requirement B.

Metric P-R1 (Refined Concept): The "phrase in B" describes a more refined concept than the "phrase in A."

Metric P-R2 (New Constraint): The "phrase in B" is missing from Requirement A.

Metric P-M (Modality Change): The "phrase in A" has a different modality than the "phrase in B."

For each applicable phrase metric, the investigator creates a corresponding assertion that includes the original phrases from both requirements that justify the metrics application. The assertions are documented and preserved for traceability and later reviewed by other investigators. Table E.2 presents the assertions A_1-A_4 created from comparing the obligation NCSU O-29 (Requirement A) with Cisco requirement SW-50.11 (M2) (Requirement B):

The statement metrics S-R/S-G and phrase metrics P-R1/P-G1 and P-R2/P-G2 are symmetric pairs. To illustrate, consider the assertion A_5 (below), which is symmetric with A_2 in Table E.2 and which would result from inversely comparing Cisco SW-50.11 (M2) to NCSU O-29:

Table E.2: Example application of phrase-level metrics

Metric	Assertion
P-G1	A_1: Generalizes from "through operating system functions for displaying text" to "using the standard function calls"
P-R1	A_2: Refines from "textual information" to "text"
P-R1	A_3: Refines from "provide" to "draw"
P-R2	A_4: Refines to include "this applies to text in the client area as well as text in custom controls"

A_5 **(P-G1):** Generalizes from "text" to "textual information"

The four statement metrics were empirically validated using two inter-rater reliability statistics for nominal data: Cohens Kappa for two raters [45] and Fleiss Kappa for multiple raters [65]. The Cohen and Fleiss Kappa statistics are a value [0,1] and measure the actual observed agreement among raters (P) excluding the expected agreement (P_e), if agreement were due strictly to chance, and is expressed by the formula: $\kappa = (P - P_e) \div (1 - P_e)$ [45, 65]. For Fleiss Kappa, we used a stratified sample of legal and product requirements and four raters who are all graduate students enrolled in a requirements engineering course. We observed a 61.2% probability that the agreement among four raters occurred beyond what is expected by chance. For Cohens Kappa and the same stratified sample, we observed that the "expert" and three of the four raters were in agreement 70-80% beyond what is expected by chance. Different factors, including the survey instrument, number of categories and number of raters, influence the Kappa statistics. We could not evaluate the phrase metrics using the Kappa statistics, however, because these metrics violate the statistics assumption that categories are mutually exclusive (i.e., a requirement pair may be classified by multiple phrase metrics).

Index