Managing Multi-Jurisdicational Requirements in a Computational Legal Landscape

Travis D. Breaux **David G. Gordon†**

March 11, 2011
CMU-ISR-11-102

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213

Abstract

Increasingly, information systems are becoming distributed and pervasive, enabling organizations to deliver services remotely to individuals and to share and store personal information, worldwide. However, system developers face significant challenges in identifying and managing the many laws that govern their services and products in this new multi-jurisdictional environment. To address this challenge, we apply the concept of a computational requirements document to multiple U.S. state regulations that share a common theme, data breach notification. The document is expressible using a formal requirements specification language (RSL), which allows document authors to codify, design, debug, analyze, trace, and visualize relationships among requirements from different policies and regulations. To measure gaps and overlaps between regulations, we applied previously validated requirements metrics. Our findings include a formalization of the legal landscape using operational constructs for high- and low-watermark practices, which correspond to high- and low standards of care, respectively. Business analysts and system developers can use these watermarks to reason about compliance trade-offs based on perceived businesses costs and risks. We discovered and validated these constructs using five U.S. state data breach notification laws that govern transactions of financial and health information of residents of these five states.

† Engineering and Public Policy

This research was supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, under the auspices of the Institute for Information Infrastructure Protection (I3P) research program.

Keywords: requirements specification; traceability; formal languages; legal requirements; standardization

1 Introduction

The Internet and wireless computing are enabling increasingly distributed and pervasive systems. In distributed systems, computer state, data and functions can be stored in multiple remote servers independent of the client's geographic location; in pervasive systems, the client is physically unbounded and can connect to the network from multiple locations. In both cases, software developers must respond to government and industry regulations that affect their product and service requirements. Because these regulations are bound to the geography in different ways (such as the area where the data is stored, where users access remote services, or where users are citizens), developers must contend with a multi-jurisdictional environment. In addition, new laws are enacted each year to improve information privacy and security, often in response to unanticipated and innovative uses of computer technology. This changing environment necessitates new theory to identify a process to achieve regulatory harmony.

In the United States, a prominent example includes the recent surge in state data breach notification laws, which have been empirically observed to reduce identity theft [30]. Collectively, these laws combine the act of notification across a data supply-chain with technical security controls targeted at different information types, business practices and consumers. The challenge for developers, especially in small businesses, is to distill regulations into actionable requirements that are traceable across their business practices. In this environment, we believe that existing approaches to governance, which consists of independently published, paper-based laws and policies, can no longer scale with rates of technology innovation. If an honest expectation of compliance is to be preserved in this new environment, regulations must be made accessible to policy makers, business analysts and software developers, alike.

We propose that regulators and industry can reach a coordinated solution wherein regulations are computational artifacts, dynamically linked across jurisdictions. These computational artifacts can integrate with industry standards to become more easily comparable and addressable in a manner that reflects the jurisdiction of the computer state, users' location, and the rate of technological change. To this end, we report our efforts to formalize a portion of the legal landscape using a requirements specification language (RSL) and apply previously validated metrics [3] to compare regulatory requirements using a gap analysis. Using the RSL and gap analysis results, we developed operational constructs for high- and low-watermarks to identify and resolve potential conflicts across multi-jurisdictional requirements and provide system developers with guidance on how to operationalize regulations. By making these potential conflicts salient, system developers can expressly consider the trade-offs based on business costs and risks through guided discussions with their legal advisers.

The remainder of the paper is organized as follows: in Section 2, we discuss related work; in Section 3, we introduce the RSL by example; in Section 4, we review our metrics for comparing requirements; in Section 5, we present our empirical case study design; in Section 6, we discuss our research findings, including prominent examples of watermark-motivated trade-offs; in Section 7, we discuss threats to validity; and in Section 8, we conclude with discussion and future work.

2 Related Work

Requirements engineering occurs in the early stages of modern software engineering, wherein terminology is to be grounded "in the reality of the environment for which a machine is to be built" [21]. As such, significant effort is invested into managing and analyzing natural language requirements and discovering new ways to formalize this informal domain. We now discuss related work in requirements engineering, artificial intelligence, and law.

Requirements specification languages (RSLs), including requirements modeling languages (RMLs), have a rich history in requirements and software engineering [23]. RSLs include informal, natural language descriptions to provide readers with context and elaboration, and formal descriptions, such as mathematical logic, to test assumptions across requirements using logical implications [13]. Goal-oriented languages, such as i* [36] and KAOS, and object-oriented notations, such as ADORA [18], include graphical notations to view relationships between entities, such as actors, actions and objects. Because of computational intractability and undecidability of using highly expressive logics, RSLs often formalize only a select class of requirements phenomena, e.g., using various temporal logics, including interval, real-time [9] or linear [14] temporal logic, or description logic [4]. Consequently, RSLs and RMLs may struggle with the balance between expressability and readability [13].

Unlike i*, KAOS, and ADORA, the RSL presented herein is designed for the policy domain by integrating formal expressions of document structure with semi-formal expressions of rights, permissions and obligations, which are required to express regulatory requirements [5]. The RSL emphasizes readability by requiring limited formalization of:

actor roles, constraints on those roles, and Boolean logic to express pre-conditions; definitions and their scope of applicability; and cross-references as typed relations between requirements. Finally, the RSL codifies the document structure (sections, paragraphs, and references) to ensure certain legal effects from cross-references are traceable and operational, which has been identified as a shortfall in current practice [22, 28, 34].

Studies to formalize laws have long been a topic of interest. Early work in the 1980's to encode laws in first-order logic began with a focus on decision support tools [1, 32], whereas a recent resurgence in formalization of privacy and security regulations have sought to test new theories as expressions of law [11, 27, 25]. In software requirements engineering, the emphasis is on requirements specification and analysis to develop tools for managing legal requirements. This work has emphasized methodology for encoding laws as rights, permissions, obligations [5], ownership and delegation [16] and techniques for formalizing the legal effects of cross-references, definitions, and exceptions in a comprehensive legal requirements management strategy [6]. Recent analysis of external cross-references emanating from the Health Information Portability and Accountability Act (HIPAA) shows the potential for conflicts between laws governing different industries [26].

Research to compare natural language has long focused on document-level comparisons. K-means cluster [19] and latent semantic indexing [10] have been applied to compare documents by examining term frequencies after cleaning the text by removing term suffixes, called stemming [33], punctuation, etc. Similar techniques have since been applied to requirements analysis to create traceability links between regulatory requirements and product requirements [8]. In a recent gap analysis between regulatory and product requirements, we discovered that significant domain knowledge is required to recognize semantic differences between requirements, i.e., subsumption, polysemy or synonymy [3]. While tools such as WordNet [12] are used in NLP to supplement domain knowledge for many problems, our research indicates that comparing requirements remains largely a manual process.

3 The Requirements Specification Language

In preparation to compare regulatory requirements across jurisdictions, we translate the original regulations into a canonical form using a requirements specification language (RSL). The RSL makes several assumptions about the domain of requirements. These assumptions were first observed in our study of regulations and thus they were incorporated into the RSL syntax and semantics described here. In the discussion that follows, we use the following excerpt from Arkansas Title 4, §110.105 to present the RSL:

> **4-110-105. Disclosure of security breaches.**
>
> **(a)(1)** Any person or business that acquires, owns, or licenses computerized data that includes personal information shall disclose any breach of the security of the system… to any resident of Arkansas…
>
> **(2)** The disclosure shall be made in the most expedient time and manner possible and without unreasonable delay, consistent with the legitimate needs of law enforcement as provided in subsection (c) of this section

Figure 1. Excerpt from the Arkansas (AR) Title 4, §110.105 of the Personal Information Protection Act

3.1. Document Model

The RSL is applied directly to original text, converting statements and phrases from the text into expressions in the RSL. Figure 2 shows the excerpt from Figure 1 expressed in the RSL: reserved keywords, special operators, and line numbers (found along the left side) appear in bold. The `DOCUMENT` keyword (see line 1) is used to assign a unique index to the specification. The `SCHEMA` keyword (see line 2) is followed by an expression consisting of *components* in curly brackets. Each component corresponds to a different reference level within the document model, beginning with the outermost component, in this case the title and chapter. References within the specification will be parsed using this schema. Line comments are indicated by the "//" operator. We use the ellipsis "…" to indicate omissions from the specification to simplify presentation in this paper.

```
1   DOCUMENT US-AR-4-110
2   SCHEMA{title:4}-{chapter:110}-{section:\d+}{par:
    \([a-z]\)}{par:\(\d+\)} //...
3   TITLE 4-110 Personal Information Protection Act
4
5   SECTION 4-110-105 Disclosure of security breaches
6   PAR (a)
7   PAR (1)
8   person!
9     |business!
10    & acquires, owns, or licenses computerized data that includes
        personal information
11    : shall disclose a breach of the security of the system to any
        resident
12  PAR (2)
13  disclosure!
14    : shall be made in the most expedient time and manner possible
        and without unreasonable delay
15    ANNOTATE timing requirements
16    REFINES (1)
17    EXCEPT (c)(1) #1
```

Figure 2. Excerpt from Arkansas 4-110-105 expressed in the RSL

The document model consists of sections and nested paragraphs, expressed in the RSL by the SECTION and PAR keywords, respectively. These keywords are followed by a reference and an optional title. For example, line 5 shows the section reference 4-110-105 followed by the section title from §105 in the excerpt in Figure 1; sub-paragraphs (a) and (1) follow on lines 6-7. The parser validates the references against the previously declared document schema and constructs an internal document model that is used with cross-references to lookup definitions and requirements.

3.2. Roles, Constraints and Requirements

Requirements consist of roles and constraints on a role, organized into first-order logical expression using operators "|" for logical-or (see line 9), and "&" for logical-and (see line 10). Associativity in logical expressions is inferred from the number of tabs before the logical operator: one less tab than the previous line is right associative; otherwise the logic is left associative. Roles are noun phrases that describe the actors or objects to whom the requirements apply and are followed by "!" (see lines 8-9); constraints on a role are phrases that begin with a verb (see line 10). For a role *R* and constraint *C*, we always assume the sentence "*R* who *C*" is valid and grammatically correct for the purpose of generating natural language from this formalization. Roles and constraints are part of the *pre-conditions* in a requirement. Next follows the requirement *clause*, preceded by a ":" and modal verb, such as "shall" to indicate an obligation (see lines 11 and 14). We identify these modal verbs using established phrase heuristics [5]. Finally, the analyst can write commands in the RSL to instruct the parser to perform special operations on rules. In Figure 2, the command keyword ANNOTATE (see line 15) indicates that the following text contains comma-separated annotations that should be linked to the requirement. Annotations can be used to group requirements by shared themes.

3.3. Relations and Cross-References

Requirements are related to each other through relations and cross-references. The RSL includes several commands by default for expressing relations and can accommodate more as needed. The default commands are:

- REFINES, with the inverse REFINED-BY, indicates that this requirement is a sub-process or quality attribute that describes how another requirement is fulfilled.
- EXCEPT, with the inverse EXCEPT-TO, indicates that this requirement has an exception (another requirement). If the pre-conditions of the exception are satisfied, then this requirement does not apply (it becomes an exclusion, e.g., *is not required*).
- FOLLOWS, with the inverse PRECEDES, indicates that this requirement is a *post-condition* to another requirement, e.g., this requirement is permitted, required, or prohibited after the other requirement is fulfilled.

In Figure 2, the command keyword `REFINES` (see line 16) indicates that this requirement refines the requirements in paragraph (1). This example is a quality attribute, because the refinement on line 13 elaborates the act on line 11 (to disclose), elaborating when the act must occur (expediently, without delay). Generally, quality attributes describe the act or an object in the act of another requirement. The command keyword `EXCEPT` (see line 17) indicates this requirement has one exception in paragraph (c)(1): the first requirement.

Relative references in these commands are expanded by the parser using a simple algorithm: in Figure 2 for example, starting from the source paragraph (2), the parser ascends the document model checking the document schema for a *descending match* rooted at the current paragraph. Thus, the first check is for a matching sibling paragraph: in this case, the index (a)(1) is a match. References may be either: an index to a singular paragraph; a paragraph range separated by the "--" operator; the ".." operator, which matches the parent paragraph; or the "." operator, which matches the current paragraph. References followed by a "*" operator refer to the paragraph and all sub-paragraphs (i.e., transitive closure). Rule selection is done in three ways: a) by default, references select all rules within the referenced paragraph(s); b) singular paragraph references followed by the ordinality operator "#" and a number *n* will identify the *n^th^* rule in that paragraph (see line 17); and c) references followed by a comma-separated list of annotations will find rules that share those annotations (e.g., all "permissions" or all "timing requirements"). Using the last mechanism, document authors can organize requirements around aspects or themes shared across a system and index requirements accordingly.

3.4. Definitions and Exemptions

Definitions describe the actors and objects in the environment of the system. They can be used to organize roles and constraints into a single term-of-art, which allows document authors to substitute the term for repeated logic across requirements. For requirements with complex pre-conditions, we found this simplification to make reading the specification much easier. Because regulations govern multiple industries and systems, it is also important to coordinate and reuse definitions across separate regulations. Consider the following RSL specification in Figure 3, acquired from Nevada Chapter 603A, Security of Personal Information, §215(5), which describes definitions related to security measures for businesses who collect payment cards.

```
1   PAR 5.
2   INCLUDE 603A.215.5* 603A.215*
3   PAR (a)
4   data storage device
5     = device
6     & stores information or data from any electronic or optical
        medium
7     < computers
8       | cellular telephones
9   // ...
10  PAR (c)
11  facsimile
12    = electronic transmission between two dedicated fax machines
        using Group 3 or Group 4 digital formats...
13    ~ onward transmission to a third device after protocol
        conversion, including, but not limited to, any data storage
        device
14  PAR (d)
15  INCLUDE EXTERNAL NV-205.602 603A.215* "payment card"
```

Figure 3. Excerpt from Nevada §603A.215(5)(c)

In Figure 3, paragraph (a) on lines 3-9 contains a definition for *data storage device*, indicated by the "=" operator. Definitions are expressed similar to pre-conditions and can use the logical operators for logical-and and logical-or, in addition to the operator "<", which means "includes" and precedes examples or sub-classes (see line 7), and the operator "~", which means "excludes" (see line 13). The parser assumes definitions apply to the paragraph in which they occur, unless instructed using the `INCLUDE` keyword, followed by two references: the source location of the definitions, and the target section or paragraph to which the definitions will apply. The instruction in Figure 3, line 2 tells the parser to apply all the definitions from paragraph (5) and all sub-paragraphs (indicated by the "*") to §215. In contrast, the `INCLUDE EXTERNAL` instruction on line 15 instructs the parser to lookup the definition "payment card" by finding a regulatory document indexed by NV-205.602, and to apply this definition to §215. This second usage enables reuse of definitions from and across multiple regulations.

The RSL parser cross-links definitions to requirements by matching terms-of-art in definitions with phrases in requirements pre-conditions and clauses. Recall from Figure 3 the definitions for terms *data storage device* (line 4) and *facsimile* (line 11) and the imported term payment card (line 15) from another law, NV §205.602. The instructions `INCLUDE` (lines 2 and 15) orchestrate these definitions by applying them to all sub-paragraphs in §603A.215, which in turn instructs the parser to link each term to each matching phrase in the pre-conditions and clauses for all requirements contained therein. This includes other definitions, such as the phrase on line 13 that excludes "data storage device" from the onward transmission of a facsimile. Figure 4 illustrates the implication these definitions have on requirements in paragraphs §603A.215(1) and (2): the underlined phrases correspond to those phrases that match the terms-of-art from Figure 3 as determined by the parser.

Both *when to apply* a prescription and *the extent of* the prescription can be computationally adjusted by relaxing or tightening definitions using the includes "<" and excludes "~" operators, respectively. For example, if we redefine *payment card* to exclude *gift card*, then the scope of when to apply the requirement to comply with the PCI DSS standard (on line 6) would be further restricted to omit the case of gift cards. Alternatively, if *data storage device* were redefined to include *USB drives*, then the extent of the prohibition on moving such devices (on line 16) would be extended to include this interpretation. The ability to shape *when to apply* and *the extent of* prescriptions using the RSL can enable regulators and businesses to evolve the conditionality of regulations as new technologies emerge over time.

```
1   SECTION 603A.215
2   PAR 1.
3   data collector!
4     & doing business in this State
5     & accepts a payment card in connection with a sale of goods or
        services
6     : shall comply with the current version of the Payment Card
        Industry (PCI) Data Security Standard...
7   PAR 2.
8   data collector!
9     & doing business in this State
10    EXCEPT 1.
11  PAR (a)
12    & does not use encryption to ensure the security of electronic
        transmission
13    : shall not transfer any personal information through an
        electronic, non-voice transmission other than a facsimile to
        a person outside of the secure system of the data collector
14  PAR (b)
15    & does not use encryption to ensure the security of the
        information
16    : shall not move any data storage device containing personal
        information beyond the logical or physical controls of the
        data collector or its data storage contractor
```

Figure 4. Excerpt from Nevada §603A.215(1) and (2)

Whereas definitions shape terms used in pre-conditions and clauses of requirements, exemptions fine-tune what is excluded from pre-conditions and clauses. Figure 5 shows a description of the role "telecommunications provider" (in the RSL) with a role constraint on line 4. The `EXEMPT` keyword instructs the parser to exclude this role and constraint from all rules in §215 and all sub-paragraphs therein. While such an exemption could be stated in a definition using the excludes operator "~", exemptions provide a mechanism to tighten meanings across a document cross-section, unbounded by a single term-of-art or definition.

```
1   PAR 4.
2   PAR (a)
3   telecommunications provider!
4     & acting solely in the role of conveying the communications of
        other persons, regardless of the mode of conveyance used,
    including, without limitation (1) optical, wire line and
    wireless facilities; (2) analog transmission; and (3)
    digital subscriber line transmission, voice over Internet
    protocol and other digital transmission technology
  EXEMPT 603A.215 *
```

Figure 5. Excerpt from Nevada §603A.215(4)(a)

Figure 6 illustrates a high-level architecture for how constraints, expressed as definitions and exemptions, are traced by the parser to requirements. The arrows route constraints through parser instructions as follows: the INCLUDE EXTERNAL instruction imports (in purple) the *payment card* definition from another regulation, NV 205.602, into NV 603A.215(5)(d). The INCLUDE instruction maps (in blue) the definitions from 603A.215(5), including any imported definitions, onto 603A.215; this mapping includes the inner link from *data storage device* to *facsimile*, and the outer links to requirements in 603A.215(1) and (2). Finally, the exemption from 603A.215(4)(a) is mapped (in red) onto the requirements 603A.215 to specifically exclude interpretations that may be implied by the definitions.

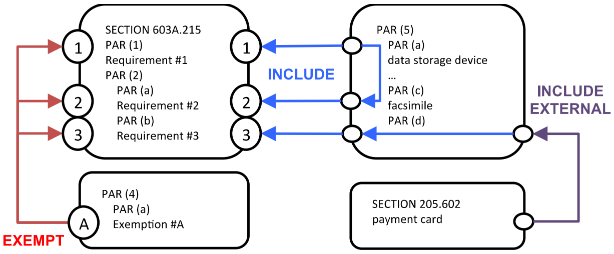


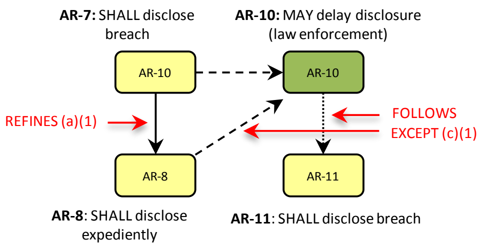
Figure 6. Summarizing the Effects of Conditionality

3.5. Tool Support and Generated Artifacts

The RSL is complemented by an automated parsing tool, which checks the language for syntax errors, such as malformed or unassociated logical expressions, and semantic errors, such as incorrect references, empty relations that refer to no rules, unreferenced definitions, and cycles among relations of the same type, e.g., REFINES, EXCEPT, FOLLOWS. The parser also handles pre- and post-clause continuations [5], wherein one or more roles and constraints apply to rules in sub- or parent paragraphs, respectively. Lastly, the parser annotates the requirements using phrase heuristics that indicate the modality of the clause [5], for example, "may" indicates a "permissions" annotation, or "shall" indicates an "obligations" annotation. Annotations are used to sort and reference requirements.

The parser constructs a model from the RSL, which is exported to other formats, such as the eXtensible Markup Language[1] (XML), HyperText Markup Language[2] (HTML) and the Graph Markup Language[3] (GraphML). Each format offers a different perspective: the HTML allows users to browse the specification by clicking hyperlinks, viewing definitions and referenced rules *in context* of a single rule; the GraphML allows users to visualize relationships across multiple requirements; and the XML enables data inter-operability and exchange with other tools. Figure 4 shows a graph generated from the RSL example in Figure 1: text labels include a unique requirement identifier (e.g., AR-7), followed by the roles in parentheses and the requirement clause (abbreviated in this figure). Nodes are colored by whether they are permissions (green), obligations (yellow), prohibitions (red) and exclusions (blue) based on annotations. Directed edges represent relations and point to referenced rules as follows: solid edges are REFINES, dashed edges are EXCEPT, and dotted edges are FOLLOWS relations.

[1] http://www.w3.org/XML/
[2] http://www.w3.org/html/
[3] http://graphml.graphdrawing.org/


AR-7: SHALL disclose breach
AR-10: MAY delay disclosure (law enforcement)
AR-10
AR-10
REFINES (a)(1)
FOLLOWS
EXCEPT (c)(1)
AR-8
AR-11
AR-8: SHALL disclose expediently
AR-11: SHALL disclose breach

Excerpt from Arkansas §110.105 expressed in GraphML

We foresee importing our models into other requirements tools that support open requirements exchange formats, such as the Requirements Interchange Format[4] (RIF) and User Requirements Notation (URN).

4 Metrics for Performing Gap Analysis

Regulations from multiple jurisdictions contain potential conflicts due to differences in the administrative hierarchy (federal, state and local jurisdictions) and requirements coverage (*who* is required to do *what*, *when*). To measure coverage gaps in natural language requirements, Breaux et al. developed a set of statement and phrase-level metrics that an analyst can apply to rationalize and document similarities and differences between two natural language requirements [3]. Unlike software quality metrics that yield numerical measurements [20], our metrics yield nominal measurements in the form of logical assertions. These metrics were validated in an empirical case study, wherein investigators performed a gap analysis between CISCO product requirements and the U.S. Access Standards (Section 508) that govern access to information by individuals with disabilities. For comparing two requirements A and B, the metrics used in this paper are:

Metric S-E (Equivalent): Requirements A and B are equivalent, with some portions of the requirements describing the same or a similar action.

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."

The process for applying these metrics to statements encoded in the RSL proceeds by: (1) identifying near-equivalent statement pairs A, B and recording a logical assertion S-E(A, B); and (2), comparing phrases between statements A, B and recording logical assertions P-G1(A, B, p_A, p_B) or P-G2(A, B, p_A) for some phrase p_A in statement A and some phrase p_B in statement B. The metrics P-G1 and P-R1 are symmetric, as are the metrics P-G2 and P-R2, based on which document the analyst begins with. The metric P-M yields an assertion P-M(A, B, p_A, p_B), wherein the phrases correspond to modal phrases, such as *may*, *must*, and *shall not*, and determine whether the requirement is a right, obligation, or prohibition [5]. For example, consider the definitions for person and business from the regulations MA and MD, respectively, shown in Figure 8 and expressed in the RSL.

MA Definition of Person:	**MD Definition of Business:**

`person`
`  = a natural person`
`
	corporation`
	`
		association`
		`
			partnership`
			`
				legal entity`	`business`
					`  = sole proprietorship`
					`
						corporation`
						`
							association`
							`
								partnership`
								`
									business entity`
									`      & whether or not organized`
									`        to operate at a profit`

Figure 7. Related stakeholder definitions in MA and MD

[4] http://www.omg.org/spec/ReqIF/

The analyst would compare these definitions by assigning the statement-level and phrase-level metrics to yield the following measures in Table 1. For definitions, the S-E measure presumes the analyst believes the term person and business are at least partially synonymous, with the remaining measures used to itemize the differences.

TABLE I. MEASURES COMPARING DEFINITIONS FROM MA §93H(1)(A) AND MD §14-3501(B)(1).

Stmt. A	Stmt. B	Metric	Measure
MA-D-1	MD-D-7	S-E	
MA-D-1	MD-D-7	P-G2	a natural person
MA-D-1	MD-D-7	P-G1	legal entity *generalizes* business entity and whether or not organized to operate at a profit
MA-D-1	MD-D-7	P-R2	sole proprietorship

To compare requirements, the metrics are applied by separately comparing the requirement clauses and the pre-conditions between two requirements. Consider the following requirements MA-20 and MD-10, which describe an obligation to send a security breach notification. This example includes the obligated actor in parenthesis.

MA-20: (Person or Business) shall provide notice, as soon as practicable and without unreasonable delay, to the owner or licensor

MD-10: (Business) shall notify the owner or licensee of the personal information of a breach of the security…

Table 2 presents select measures acquired by applying the metrics to requirements MA-20 and MD-10. Notably, the P-G2 and P-R2 metrics capture an important difference: under MA §93H(3)(a), the business must notify licensor or upstream data providers, whereas under MD §14-3504(c)(1), the business must notify licensee or downstream data providers. As shown in Figure 6 in Section 3.4, definitions transfer constraints to multiple requirements. In the case of MA-20 and MD-10, the definitions for person and business, shown in Figure 8, also apply to these two requirements, respectively. Furthermore, the measures acquired from comparing these definitions, shown in Table 1, are transferred to these target requirements as descriptions of the difference in coverage (elaborating *who* must comply).

TABLE II. MEASURES COMPARING REQUIREMENTS FROM MA §93H(3)(A) AND MD §14-3504(C)(1).

Stmt. A	Stmt. B	Metric	Measure
MA-20	MD-10	S-E	
MA-20	MD-10	P-G2	licensor
MA-20	MD-10	P-R2	as soon as practicable and without unreasonable delay
MA-20	MD-10	P-R2	licensee

5 Research Methodology

We now describe our case study research method [35] used to compare multi-jurisdictional requirements from repeated observations of natural language expressions in regulatory documents. The method includes our selection criteria, the translation process, units of analysis, and analysis procedure.

This paper only presents preliminary results towards our goal to observe variation in regulations across multiple jurisdictions with the aim of understanding how regulations introduce complexity into system requirements. To observe this variation, we selected a single theme (data breach notification) to limit the effects of dissimilarity while we build new theory to reconcile differences and potential conflicts. In the United States, this theme represents the recent enactment of 46 state and territorial laws from 2002-2009, each governing personal information about state residents. For distributed and pervasive systems, variations in these laws require businesses to reconcile different legally required practices for customers of different states. The laws we selected in this study are as follows:

- **AR**: *Personal Information Protection Act*, Arkansas Chapter 14.110, enacted 2005.
- **MA**: *Security Breaches*, Massachusetts Chapter 93H, enacted 2007.
- **MD**: *Personal Information Protection Act*, Maryland Subtitle 14-35, enacted 2008.
- **NV**: *Security of Personal Information*, Nevada Chapter 603A, enacted 2006.
- **WI**: *Notice of Unauthorized Access to Personal Information*, Wisconsin Chapter 134.98, enacted 2006.

We down selected from 46 laws to 5 laws using two criteria: first, we invited suggestions from a legal expert with seven years of privacy and security law expertise to highlight industrial challenges, resulting in AR, MA, MD, NV, and lastly, we included the State of Wisconsin, because it uniquely covers biometric information, including fingerprints, voice prints, retinal images and unique physical characteristics.

Our translation process was conducted by two investigators (the authors) separately translating each statement within each law using the RSL. The process includes a general classification of each statement, as a definition, requirement, or exemption, and writing an expression in the language to characterize the statement. Definitions were identified by common phrases, such as "*x* means *y*", where a term *x* has the logical definition *y*. Requirements were identified using the phrase heuristics identified by Breaux et al. [53], which were extended during this study. Comments were used in the translation to capture questions, issues and other discrepancies. We maintained a *caveats list* of translation strategies that reflect unusual cases and how the parser should treat such cases, and a *proposed changes list* of requirements with examples for new language constructs. As a new construct was introduced, we reviewed each law to update the translation to reflect the new construct to ensure consistency across the entire dataset.

The units of analysis correspond to the translated requirements, definitions, exemptions, and relations between requirements, in addition to the measures produced by the gap analysis. The RSL acts as a natural filter, capturing only what it can express, which is a threat to validity discussed in Section 7. After the translation, we analyze the units of analysis to identify propositions that link the units to our findings through pattern-based inferences [7]. These patterns consist of constant features (the types of relations and metrics) and the manner by which these constant features structure variable features in the observable phenomena (the different requirements in the relations and the phrase-level measures). We explain the different patterns in our research findings in Section 6.

In the analysis procedure, we first compare similar definitions, which either have the same term (two definitions for "breach of security") or share similar constraints (two definitions, one for "person" and the other for "business" both include "corporation" as a kind or sub-type). We applied the phrase-level metrics to the definitions to identify the dissimilar sub-types and constraints on those types. Second, we compared the requirements by applying the metrics from Section 3 to the requirements clauses and pre-conditions. We analyzed the measures as follows: for two requirements clauses measured using the S-E metric, we applied the phrase-level metrics to distinguish the differences in terms of *who* is permitted, required or prohibited to do *what*. Next, we consider the dissimilarity between these two requirements in terms of the relations (e.g., does one requirement have an exception not observed in the other, different refinements, or pre- or post-conditions). We call these two types of comparisons intra- and inter-dissimilarity, respectively. We now discuss our research findings, including the patterns observed through our analysis.

6 Research Findings

The translation of the five laws by two investigators (the authors) required an average of 2.86 minutes per statement with the first document requiring an average of 2.75 hours or 4.23 minutes per statement, which includes the time to discover the RSL; the longest document consisting of 49 statements required an average of 1.5 hours. Each investigator spent an average total of 9 hours to encode the five laws. Figures 9 and 10 present summary statistics for the units of analysis encoded in the RSL. Recall these laws cover the same theme (data breach notification). We observed the number of definitions did not vary greatly and that the number of exemptions was a matter of writing style; neither definitions nor exemptions are proportional to the number of requirements in this dataset.

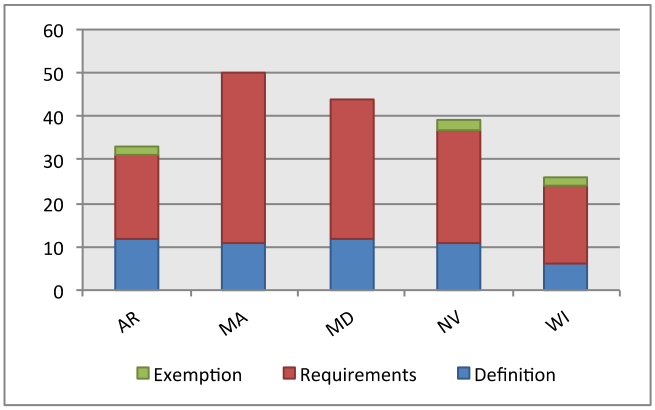


Figure 8. Summary Units of Analysis– Statements

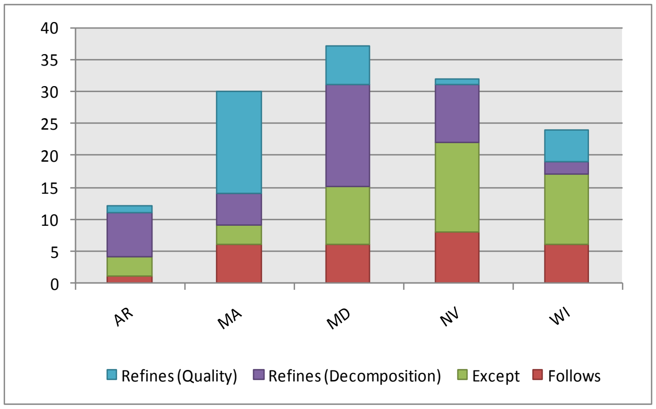


Figure 9. Summary Units of Analysis – References

The references reported in Figure 10 originate from multiple sources, including: *anaphora*, which is indicated by determiners (such) and pronouns (this); *case-splitting*, which is indicated by English conjunctions (and, or) separating verb clauses that are headed by a modal phrase (must, may, shall); and direct references to sections and paragraph that may be anaphoric (*this* section, *this* paragraph) or indexed by paragraph number, such as "paragraph (a)." Because operationalized references in the RSL are more precise, we determined that the RSL reduces ambiguity by eliminating false-positives for these five laws from the set of referenced requirements in another paragraph or section introduced by cross-references.

Our analysis of statements, relations, and measures acquired from the gap analysis yielded several observations. These observations include patterns of dissimilarity, heuristics for reconciling differences and for discovering a legal landscape, and variations in document writing styles that affected our method.

6.1. Patterns of Dissimilarity

When an organization is subject to multiple regulations governing similar business practices, it is inevitable that the requirements may overlap either completely (identical requirements) or a partially (the requirements are related but differ by one or more constraints). Identical requirements, identified by the S-E metric, without any observed phrase measures, pose no issue; complying with one requirement is compatible with complying with the other. However, when the overlap is partial, the differences between each requirement must be reconciled in order to achieve compliance with both regulations. At this juncture, we outline various differences between requirements and demonstrate (by example) how they may be reconciled. These differences occur "in-the-small" and "in-the-large", which we define respectively to be:

Intra-dissimilarity: differences *within* two requirements from two different documents, as determined using phrase-level metrics

Inter-dissimilarity: differences *among* two requirements, as determined by analyzing dissimilar `REFINES`, `EXCEPT`, and PRECEDES relations to other requirements

An organization must address and reconcile these types of differences before integrating multi-jurisdictional requirements into their systems, policies, and procedures. Normally, this integration is a difficult procedure due to lack of traceability; however, the RSL and gap analysis offer an improved method for identifying, displaying, and addressing these differences, as evidenced by the following examples of intra-dissimilarity. Consider Figure 11, which shows two requirements: MD-7 from Maryland §14.3504(b)(2) and NV-9 from Nevada §603A.220(1).

MD-7

```
business!
  & concludes the investigation
  : shall notify the individual
```

NV-9

```
data collector!
   : shall disclose the breach to the resident…
```

Figure 10. Maryland and Nevada disclosure details (RSL)

MD-7 and NV-9 both obligate the entity to notify the individual of a data breach, but their pre-conditions differ significantly: MD-7 requires that the entity conduct an investigation into the breach, whereas NV-9 requires no such investigation. As it is unlikely that this investigation would interfere with the notification proposed by Nevada, an entity could achieve compliance with both regulations by conducting the investigation as if the precondition were present in both obligations.

Regulatory requirements may contain thresholds to limit the scope of an obligation. These thresholds can vary across states, for example, consider MD-18 from Maryland §14-3504(e) and AR-14 from Arkansas §110.105(e)(3).

MD-18: if the (business) demonstrates that the cost of providing notice would exceed $100,000, or that the affected class of individuals to be notified exceeds 175,000, they may give notification by substitute notice

AR-14: if the (person or business) demonstrates that the cost of providing notice would exceed $250,000, or that the affected class of individuals to be notified exceeds 500,000, they may provide substitute notice

Both Maryland and Arkansas provide the option of substitute notice when the standard notification methods would be prohibitively complex or expensive. However, the levels at which substitute notice become available differ for each state. Due to these "hard" requirements, reconciliation into a single requirement is not possible without suffering the loss of significant information. In cases such as these the optimal decision is to keep the requirements separate and satisfy each individually.

In an effort to reduce overhead or maintain simplicity, some organizations may only wish to adopt the “common thread” between requirements – that is, what remains when the differences between the two are excluded. While possible, this practice comes with inherent risk. As shown in Figure 12, requirement WI-3 from Wisconsin 134.98(2)(b) and AR-7 from Arkansas §110.105(a)(1) have a conflict over when notice should be issued – whether an organization “reasonably believes” or “knows” that personal information has been acquired. Presumably, “knows” requires stronger evidence than “reasonably believes.” Thus, an organization choosing “knows” will less frequently need to apply this requirement. Differences at all levels – even those of only two or three words – can have significant impacts on implementation, organizations must maintain traceability from the choices they face and their decisions to implement those choices in practice.

WI-3

```
entity!
   & knows that personal information … has been
   acquired…
      : shall make reasonable efforts to notify…
```

AR-7

```
person!
   | business!
   & reasonably believes the personal
   information… was unencrypted and acquired…
   : …shall disclose any breach… to any resident
of Arkansas
```

Figure 11. Wisconsin and Arkansas disclosure details (RSL)

In addition to intra-dissimilarity observed in phrase-level measures, inter-dissimilarity appears in the presence or absence of relations to other requirements. The following example demonstrates how relations used to link requirements – REFINES, EXCEPT, PRECEDES – result in inter-dissimilarity. In Figure 13, requirement AR-3 from Arkansas §110.104(a) and NV-4 from Nevada §603A.210(1) are compared using the S-E measure (shown by a double solid line).

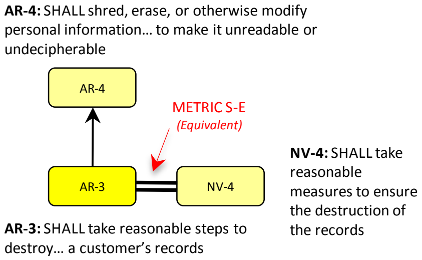

AR-4: SHALL shred, erase, or otherwise modify personal information… to make it unreadable or undecipherable
AR-4
METRIC S-E
(Equivalent)
NV-4: SHALL take reasonable measures to ensure the destruction of the records
AR-3
NV-4
AR-3: SHALL take reasonable steps to destroy… a customer’s records

Figure 12. Excerpt from Arkansas and Nevada Comparison

Both AR-3 and NV-4 require that customer records no longer in use must be destroyed; however, Arkansas further constraints the solution space through a REFINES relation to indicate methods of destruction that must be used to make personal information unreadable or undecipherable (the solid arrow to AR-4 in Figure 13). If applying AR-4 to Nevada state resident’s personal information is unlikely to add significant burden, the organization can adopt AR-4 as a standard for destroying data under both regulations. In most cases, the presence of a REFINES on one requirement but not the other can be handled by duplicating the additional refinement(s) across the equivalency, establishing these additions as a standard to be followed for both jurisdictions. Figure 14 presents a more complex example in which three parallel equivalencies are identified: AR-7 and NV-9, which require disclosing data breaches to state residents; AR-8 and NV-10, which require the disclosure to occur expeditiously; and AR-10 and NV-12, which permit a delay by law enforcement.

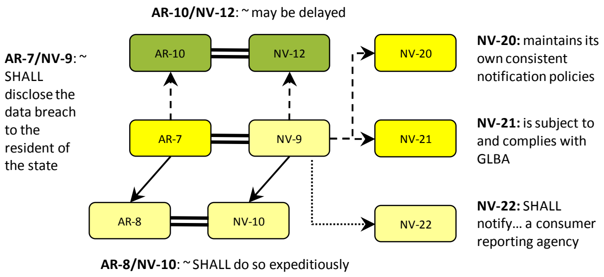

AR-10/NV-12: ~ may be delayed
AR-7/NV-9: ~ SHALL disclose the data breach to the resident of the state
AR-10
NV-12
NV-20
NV-20: maintains its own consistent notification policies
AR-7
NV-9
NV-21
NV-21: is subject to and complies with GLBA
AR-8
NV-10
NV-22
NV-22: SHALL notify… a consumer reporting agency
AR-8/NV-10: ~ SHALL do so expeditiously

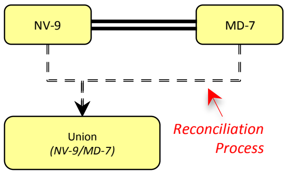
Figure 13. Excerpt from Arkansas and Nevada Comparison

Apart from these three equivalencies, Nevada has additional requirements linked by `EXCEPT` (the dashed arrows to NV-20, NV-21) and PRECEDES (the dotted arrow to NV-22). The exceptions provide alternative notification mechanisms (comparable internal policies or procedures or compliance under the GLBA). The post-condition NV-22 requires additional notification to consumer reporting agencies. Because exceptions can halt the discharge of an obligation, the presence of exceptions in one regulation and not another at these equivalencies can cause conflicts. The post-condition, however, is an additional obligation that extends the requirements of the organization, thus they can be treated in the same fashion as `REFINES` relations.

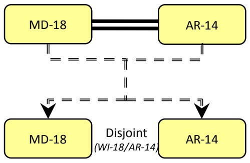
6.2. The Legal Landscape and Positioning

The patterns of dissimilarity illustrate potential conflicts between two regulatory documents at a very atomic level: as binary comparisons between single requirements. We analyzed the seemingly vast number of comparisons that can be made, and discovered three heuristics for reconciling differences, which appear in Table 3. Our discussion in Section 6.1 presents situations in which these heuristics can be used to resolve potential conflicts or differences between requirements.

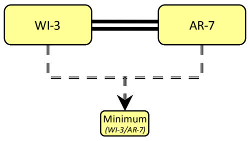
TABLE III. HEURISTICS FOR RECONCILING REGULATORY DIFFERENCES



Union Requirement: merge expectations (adhering to both if not purely equivalent, or the greater of the two in the case of inclusion)



Disjoint Requirement: employ practices that allow adherence to each within its respected jurisdiction



Minimum Requirement: determine the floor or lowest common standard

We believe these heuristics can be applied to potential conflicts across regulatory requirements to discover a legal landscape. The landscape consists of choices that system designers must consider in the context of their products and services, business practices, internal policies, preferences, and risk profiles. The borders of the landscape are defined by different standards of care for a finite set of requirements across multiple regulations. A *low watermark standard* is a standard of care that satisfies the minimum requirements by making the fewest decisions in the reconciliation of differences between requirements and occurs when two requirements are precisely equivalent (because there is no requirement from which to presume a higher standard in the finite set of requirements). A *high watermark standard* is a standard set in which an organization proposes to achieve compliance by the "union" or the "disjoint" separation of differences between requirements. The low watermark standard results in the abandonment of relevant details: usually refinements measured by the P-R1 or P-R2 metrics. Alternatively, the high watermark standard seeks to maintain these details in order to achieve or exceed compliance.

TABLE IV. QUALITIES OF WATERMARKS

	High Watermark		**Low Watermark**
Decisions	***Union***	***Disjoint***	***Minimum***
Compliant	Yes	Yes	No
Source of Cost	Exceeds Standards	Administrative and Logistical	---
Risk	Low	Low	High

Achieving a high watermark will incur costs beyond those necessary to satisfy the requirements themselves. If dissimilar requirements are reconciled through the use of unions, additional resources will likely be needed given that the covered entities (in this case, additional states) will have increased in number. If the two requirements are kept disjoint, we anticipate the need for additional resources (overhead) to maintain separate practices or processes. However, while both of these approaches to dissimilarity resolution result in higher costs, they take on less risk than adhering to the low watermark standard, which fails to achieve full compliance.

6.3. Variation Among Practices

While our heuristics offer guidance in reconciling differences, some documents contain inconsistent styles that inhibit uniform processing and interpretation based on our method, which we now discuss. Examples include MA §93H, which retains constraints on what *may*, *must*, or *must not* be done within definitions as opposed to moving these constraints into rules. Another example includes NV §603A, which lacks an overarching goal to lend direction and context to the document and under which other requirements can be linked as refinements, exceptions, etc. We now discuss examples of these inconsistencies that we observed during our study and how they affected our findings.

Within our documents set, we found common practice was to define notice gradually across multiple requirements, leveraging preconditions to add or remove constraints on the notice, such as the permission (or prohibition) for notice to be given through an organization's website. The approach taken in MA §93H(1)(a) retains many of these constraints in the definition of notice (Figure 15). For example, the definition describes three kinds of notice: written, electronic or substitute. Other regulations have expressed these kinds as permissions to provide these notices, which are refinements upon the obligation to provide notice. For the application of our method, analysts must ensure they compare requirements to definitions to capture these potential overlaps and conflicts.

> "Notice" shall include:—
>
> (i) written notice;
>
> (ii) electronic notice, if notice provided is consistent with the provisions regarding electronic records and signatures set forth in § 7001 (c) of Title 15 of the United States Code; and chapter 110G; or
>
> (iii) substitute notice, if the person or agency required to provide notice demonstrates that the cost of providing written notice will exceed $250,000, or that the affected class of Massachusetts residents to be notified exceeds 500,000 residents, or that the person or agency does not have sufficient contact information to provide notice.

Figure 14. MA §93H(1)(a) Notice Excerpt

Safe harbors are important regulatory mechanisms that encourage organizations to accept some known outcome or costs in the face of uncertainty. Safe harbors can be conveyed in many ways in the original text and analysts must be aware of these different formats. Using the RSL, safe harbors can be encoded as exemptions and deference to standards, exclusions (is not required to), and "lynchpin" conditions, which, when satisfied or not satisfied cause portions of the regulation to not apply to an organization, their practices or products. All of these safe harbor strategies can be found in NV §603A. To illustrate, consider Figure 16: NV §603A.215 shows deference to another standard, the Payment Card Industry (PCI) Data Security Standard (DSS).

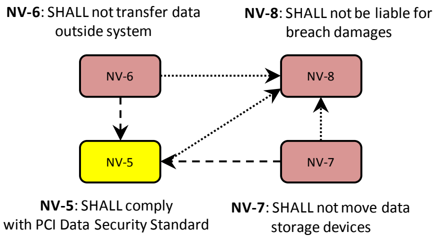

NV-6: SHALL not transfer data outside system
NV-8: SHALL not be liable for breach damages
NV-6
NV-8
NV-5
NV-7
NV-5: SHALL comply with PCI Data Security Standard
NV-7: SHALL not move data storage devices

Figure 15. Nevada §603A Safe Harbors (GraphML)

In general, NV §603A.215 contains several requirements that apply to data collectors that accept the payment cards for their services. These requirements are considerable and include restrictions on the use of data storage devices (NV-7) as well as a prohibition against transmitting data over non-secure media (NV-6). An exemption that applies to this section (shown previously in Figure 5) excludes telecommunications providers from these requirements. In this case, the safe harbor is encoded using the EXEMPT keyword in the language.

Deference to standards is another technique used for providing safe harbors, and occurs when requirements are removed or satisfied through compliance with another (often external) standard. In Figure 16, the prohibitions (in red) NV-6 and NV-7 do not apply (via the EXCEPT relation, shown by the dotted line arrows), if the data collector exercises this exception by choosing to comply PCI-DSS.

Similar to the deference to standards practice, an exclusion occurs when an organization satisfies a requirement within a document that serves to satisfy another obligation. For example, NV-20 (Figure 14) states that an entity that maintains and follows its own notification procedures that are consistent with the timing of notices specified in the document shall be deemed in compliance with the section regarding the type and delivery method of notice.

Lastly, perhaps the most obscure type of safe harbor is what we call a "lynchpin" condition. These conditions occur in definitions and requirements and, if satisfied, cause the requirement to which they apply to not apply. In addition, all refinements, exceptions, and some post-conditions linked to such requirements "drop out" as a consequence of their dependence on these drop-outs. In Figure 17, a number of requirements can be traced back to NV-9. These requirements elaborate NV-9 in a number of ways, including how the notice must be provided, the types of acceptable notice, and what actions follow the notification.

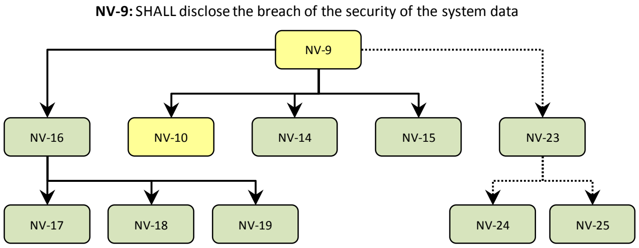

NV-9: SHALL disclose the breach of the security of the system data
NV-9
NV-16
NV-10
NV-14
NV-15
NV-23
NV-17
NV-18
NV-19
NV-24
NV-25

Figure 16. "Lynchpin" Condition in Nevada §603A

However, NV-9 has a precondition that restricts the requirement to a breach of *unencrypted* system data. Provided that an organization has encrypted their data within Nevada's definition of encryption, the entire requirements tree no longer applies to these organizations.

7 Threats to Validity

In grounded analysis, multiple analysts derive theoretical constructs from a dataset to describe or explain the data and the constructs are assumed to only generalize to that dataset [17]. Recall from Section 4 that we selected regulations that share a theme (data breach notification), thus our theory may not be externally valid in other regulated domains, such as medical devices or aviation, which may require new language constructs. However, to challenge our assumptions, we validated the schema notation and document model by visually inspecting data breach notification laws in all 46 U.S. states and territories, two U.S. Federal regulations (HIPAA Privacy Rule and Access Standards), the European Union Directive 95/46/EC and a Canadian law (PIPEDA). We found the schema and document model to be sufficiently robust to model these regulatory documents and express their cross-references.

Construct validity is the correctness of operational measures used to collect data, build theory and report findings [35]. To improve construct validity, we maintained a *caveats list* of translation strategies that reflect unusual cases and how the parser should treat such cases, and a *proposed changes list* of requirements with examples for new language constructs. As a new construct was introduced into the language, we reviewed each law to update the translation to reflect the new construct to ensure consistency across the translated datasets. In addition, we developed analytic tools using the parser and a research database to collect all the statistics reported in this paper.

Internal validity is the extent to which measured variables cause observable effects within the data [35]. Our results show that writing styles can positively or negatively impact our methodology, requiring analysts to look beyond the present context to identify dissimilarities between requirements.

Reliability describes the consistency of the theory to describe or explain environmental phenomena over repeated observations [35]. To improve reliability, both investigators (the authors) separately translated the datasets into the RSL and compared their results afterwards to identify alternate modes of expression and language caveats. For the metrics, the investigators compared a subset of their statement equivalencies (S-E measures in the gap analysis) by document pair (e.g. NV-AR, WI-MD, etc.) and determined an initial agreement or "overlap" of over 85%.

8 Discussion and Summary

In this paper, we present the results of comparing five regulatory documents using a requirements specification language (RSL) for codifying legal requirements and qualitative metrics to identifying gaps between requirements. We found the time required to translate the regulations into the RSL well worth the ability to debug and analyze the RSL-generated requirements using the metrics. While regulations were not originally written for this type of technical analysis, we believe our analysis can be used to improve the construction of these documents to reach a broader, more participatory audience throughout industry and academia by allowing participation to focus on alternative regulatory structures and the logical implications of those structures.

In Section 6, we show how measures of the RSL-encoded requirements can be used to identify patterns of dissimilarity. In addition, we presented heuristics for analysts to use to reconcile potential conflicts between requirements from different jurisdictions. We believe system designers can use the heuristics to select requirements that position their products in better position to comply with multiple jurisdictions. These selections may be based on costs to design in alternatives based on conflicting requirements, or to choose a common standard that elevates products to a higher standard.

Acknowledgment

This research was supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, under the auspices of the Institute for Information Infrastructure Protection (I3P) research program.

References

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

[2] D. Bourcier, P. Mazzega, "Toward measures of complexity in legal systems." *Int'l Conf. AI & Law*, 2007, pp. 211-215.

[3] T.D. Breaux, A.I. Antón, K. Boucher, M. Dorfman, "Legal requirements, compliance and practice: an industry case study in accessibility." *IEEE 16th Int'l Req'ts Engr. Conf.*, pp. 43-52, 2008.

[4] T.D. Breaux, A.I. Antón, J. Doyle, "Semantic parameterization: a process for modeling domain descriptions."*ACM Trans. Soft. Engr. Method.*, 18(2): 5, 2008.

[5] T.D. Breaux, M.W. Vail, A.I. Antón. "Towards compliance: extracting rights and obligations to align requirements with regulations."*IEEE 14th Int'lReq'tsEngr.Conf.*, 2006, pp. 49-58.

[6] T.D. Breaux, *Legal requirements acquisition for the specification of legally compliance informaiton systems*, North Carolina State Univetsity, Ph.D. thesis, 2009.

[7] D.T. Campbell, "Pattern matching as an essential indistal knowing,"The Psychology of Egon Brunswick.Holt, Rinehart, Winston.pp.81-106, 1966.

[8] J. Cleland-Huang, A. Czauderna, M. Gibiec, J. Emenecker. "A machine learning approach for tracing regulatory codes to product specific requirements." *IEEE/ACM 32nd Int'l Conf. Soft. Engr.*, pp. 155-164, 2010.

[9] A. Dardenne, S. Fickas, A. van Lamsweerde. "Goal–directed requirements acquisition," *Sci. Comp. Prog.*, 20:3-50, 1993.

[10] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman. "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, 41(6): 391-407, 1990.

[11] H. DeYoung, D. Garg, L. Jia, D. Kaynar, A. Datta, "Experiences in the logical specification of the HIPAA and GLBA privacy laws." ACM Workshop on Privacy in Electornic Society, pp. 73-82, 2010.

[12] C. Fellbaum, *WordNet: An electronic lexical database*. MIT Press, 1998.

[13] M.D. Fraser, K. Kumar, V.K. Vaishnavi, "Informal and formal requirements specification languages: bridging the gap." *IEEE Trans. Soft. Engr.*, 17(5):454-466, 1991.

[14] A. Fuxman, L. Liu, J. Mylopoulos, M. Pistore, M. Roveri, P. Traverso. "Specifying and analyzing early requirements in Tropos." *Req'ts Engr. Journal*, 9(2): 132-150, 2004.

[15] B. Garner, *Black's Law Dictionary*, 9th ed, West, 2009.

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

[17] B. Glaser, A. Strauss. The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine Transaction, 1967.

[18] M. Glinz, S. Berner, S. Joos. "Object-oriented modeling with ADORA."*Info. Sys*. 27: 425-444, 2002.

[19] J.A. Hartigan, M.A. Wong. "A K-means clustering algorithm" *Applied Statistics*, 28(1): 100-8, 1979.

[20] IEEE Std. 1061-1998 – Standard for a Software QualityMetrics Methodology

[21] M. Jackson. "The world and the machine." *17th IEEE Int'l Conf. Soft. Engr.,* pp. 283–292, 1995.

[22] M. Lauritsen, T.F. Gordon, "Toward a general theory of document modeling."*Int'l Conf. AI & Law*, 2009, 202-211.

[23] A.A. Levene, G.P. Mullery, "An investigation of requirement specification languages: theory and practice."*IEEE Computer*, 15(5):50-59, 1982.

[24] A.K. Massey, A.I. Anton, "Triage for legal requirements," NCSU Technical Report #TR-2010-22, October 11, 2010.

[25] J. Maxwell, A.I. Anton, "Developing production rule models to aid in acquiring requirements from legal texts." *IEEE 17th Int'l Req'ts Engr. Conf.,* 2009, pp. 101-110.

[26] J. Maxwell, A.I. Anton, "Discoverying conflicting software requirements by analyzing legal cross-references," In Submission: *ACM/IEEE Int'l Soft. Engr. Conf*., 2011.

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

[28] J. Martinek, J. Cybulka, "Dynamics of legal provisions and its representation." *Int'l Conf. AI & Law*, 2005, pp. 20-24.

[29] J. Mylopoulos, A. Borgida, M. Jarke, M. Koubarakis. "Telos: representing knowledge about information systems," *ACM Trans. on Info. Sys.*, 8(4):325-362, 1990.

[30] S. Romanosky, R. Telang, A. Acquisti. "Do data breach disclosure laws reduce identity theft?" *Workshop on the Economics of Information Security (WEIS)*, June 25-28, 2008.

[31] I. Rubinstein, "Privacy and Regulatory Innovation: Moving Beyond Voluntary Codes." (In Press) *I/S: A Journal of Law and Policy for the Information Society*, April, 2011.

[32] 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):370–386,1986.

[33] M.F. Porter."An algorithm for suffix stripping."*Program*, 14(3):130–137, 1980.

[34] R. Winkels, A. Boer, E. de Maat, T. van Engers, M. Breebaart, H. Melger. "Constructing a semantic network for legal content," *Int'l Conf. AI & Law*, 2005, pp. 125-132.

[35] R.K. Yin. *Case study research*, 4th ed. In Applied Social Research Methods Series, v.5. Sage Publications, 2008.

[36] E. Yu. "Modeling organizations for information systems requirements engineering." *Int'l Symp. Req'ts Engr.*, 1993, pp. 34-41.