Laws and regulations increasingly impose requirements on IT business practices and products to achieve societal goals such as privacy, safety, and accessibility. To meet these requirements, companies integrate their product development with an organizational infrastructure for managing compliance risks. The emphasis has often been on compliance with specific regulations, such as the 1996 Health Insurance Portability and Accountability Act (HIPAA) and the Sarbanes-Oxley Act of 2002. However, as the regulatory landscape for information becomes more complex, companies are adopting more programmatic ways to understand and integrate these requirements.

A key difficulty in responding to regulatory requirements emerges from the need to interpret legal language structures and syntax and translate them into domain-specific product specifications. Researchers at North Carolina State University developed a formal method to improve a company’s ability to systematically acquire legal requirements from policies and regulations. The frame-based requirements analysis method (FBRAM) gives developers a means to extract functional requirements that preserves the legal language from US regulatory documents. This lets FBRAM users demonstrate due diligence and good faith by tracing relevant words and phrases from their exact position in a legal text to an unambiguous property in a formal requirements specification. Over the past several years, FBRAM has been validated in three regulatory domains—information privacy, aviation safety, and information accessibility.1

As part of this process, NCSU and Cisco researchers recently teamed up to study product requirements developed to comply with US Section 508 regulations for electronic and IT accessibility.2 Our study used actual requirements implemented in Cisco products. It demonstrated the use of metrics to identify gaps between legal and product requirements as well as patterns that could help reduce ambiguities in legal compliance.

FBRAM-Based Accessibility Requirements

FBRAM builds on existing goal-based analysis methods by partitioning natural language phrases from requirements statements into categories that have a consistent meaning.1 This partitioning, called semantic parameterization, enables formal interpretations of legal requirements by resolving ambiguities that result from English grammatical structures, natural language context sensitivity, and legal cross-references.

The FBRAM abstract model comprises four method artifacts:

- a document model, representing the regulatory document structure (parts, sections, paragraphs, and so on);
- a reusable, domain-independent upper ontology, comprising statement- and phrase-level concepts;
natural language phrase heuristics, mapping concepts in the upper ontology to informal definitions and natural language patterns; and

a frame-based markup language, formalizing a context-free grammar that developers use to maintain traceability from the legal text to concepts in a lower ontology and to formal assertions.

FBRAM employs these artifacts in a process model (see Figure 1). The process generates legal requirements that organizations can analyze in a domain context and use to reason about compliance decisions.

Requirements are captured in a tabular format. For example, Figure 2 shows the FBRAM format for the following Section 508, Subpart B requirement:

When an image represents a program element, the information conveyed by the image must also be available in text.

Excerpt phrases are classified into properties. For example, “When…” indicates a precondition, and the modal phrase “must” indicates that the statement represents an “obligation.” The requirements are normalized using a standard sentence schema: [ACTION] [object] [instrument] [purpose], [condition]. For the example in Figure 2, this schema yields

NCSU O-27: MAKE AVAILABLE information conveyed by the image in text, when an image represents a program element.

where “O” is the requirement identifier for the “obligation” modality. Other modalities are “P” for “permission” and “R” for “refrainment.”

Cisco Accessibility Requirements
Cisco based its product requirements on an accessibility template developed by IBM and published online (www.ibm.com/able/guidelines). The Cisco Accessibility Team refined and contextualized the templates for Cisco product features.

In addition, the team worked with Inclusive Technologies, a company that provides domain expertise, market research, and training on accessibility related issues. The company connected Cisco with a relevant community of practice—that is, a community engaged in a collective learning process over time. These communities work to negotiate and refine explicit meaning from di-

Figure 1. The frame-based requirements analysis method (FBRAM) process model. The numbers in black circles represent a linear-temporal progression of composite procedures to extract formal requirements from US federal regulatory documents.

<table>
<thead>
<tr>
<th>Record Number: 27</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paragraph</td>
</tr>
<tr>
<td>1194.21(d)</td>
</tr>
<tr>
<td>1194.21(d)</td>
</tr>
<tr>
<td>1194.21(d)</td>
</tr>
<tr>
<td>1194.21(d)</td>
</tr>
<tr>
<td>1194.21(d)</td>
</tr>
</tbody>
</table>

Figure 2. Example frame-based requirement for Section 508 excerpt. The excerpt phrases are classified into properties that parameterize and itemize legal requirements.
verse opinion and formal knowledge from tacit experience. Because accessibility issues affect individuals differently, the accessibility community provided critical information for meeting or exceeding regulatory expectations for products.

It also helped Cisco monitor changes in regulatory interpretations. Some of these changes reflect sociopolitical priorities—for example, the different enforcement policies of different presidential administrations. Other changes coincide with new technology developments.

The accessibility community of practice helped Cisco monitor changes in regulatory interpretations. Some of these changes reflect sociopolitical priorities; others coincide with new technology developments.

For example, the accessibility standards requiring the capability to disable scripts in Web pages originated in difficulties that electronic screen readers for the visually impaired had in parsing these scripts. Today’s screen readers have overcome this challenge, so scripts are typically not disabled. However, as Ajax and Web 2.0 take hold, they pose a similar technological challenge, so this requirement is becoming relevant again.

Communities of practice can help industry account for such changes in prioritizing regulatory compliance.

**Comparative Case Study Design**

We employed an exploratory, single-case study focused primarily on identifying the kinds of gaps that existed between the NCSU and Cisco requirements and the Section 508 accessibility standards. We defined a “gap” as a requirement in either the NCSU or Cisco requirements set that didn’t appear in the other set or as a paragraph in the accessibility standards that wasn’t cross-referenced to an NCSU or Cisco requirement.

### Metric-Based Comparisons

We identified gaps empirically in a fine-grained comparison that required investigators to think critically about subtle semantic differences or similarities. To capture the investigators’ critical thinking in a logical assertion, we had them use qualitative metrics to rationalize the comparison of two requirements. First, they applied three *statement metrics* to a requirement pair, evaluating them at a general level as follows:

- **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.

Next, they used *phrase metrics* to compare discrete phrases in two requirements and further clarify the similarity or difference in meaning:

- **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 from the “phrase in B.”

For each applicable phrase metric, investigators created a corresponding assertion that included the original phrases justifying the metric’s application. They documented the assertions, which were preserved for traceability and later review by other experts.

### Pattern-Based Alignments

The assertions represent nominal measurements that we further analyzed using pattern-based inferences to identify propositions that link the data to our findings. These patterns consist of constant features (the assertion type) and the manner by which assertions of the same type coordinate variable features in the observed phenomena (the requirements statements and their phrases). For example, developers can use the PR-1 (Refined Concept) metric to identify reasonable interpretations for the legal concept “accessibility features.”

Applying the metrics to the NCSU and Cisco requirements resulted in 635 total statement
Accessibility and Section 508

U.S. laws governing accessibility to electronic information by individuals with disabilities have evolved over the past two decades to keep pace with technology.

The primary impetus was Section 508 of the 1986 US Rehabilitation Act amendments, which required manufacturers to develop guidelines “to ensure that handicapped individuals may use electronic office equipment with or without special peripherals.”

Although this legal phrasing was in step with an office automation tradition, it lacked sufficient detail to address the many nuances of complex information systems that US citizens would start using in the 1990s. The 1998 Rehabilitation Act amendments increased Section 508’s reach by extending access to “electronic and information technology” in general and by requiring reporting, complaint, and enforcement mechanisms to improve accountability.

In December 2000, the US Architectural and Transportation Barriers Compliance Board published the final regulatory rule, “Electronic and Information Technology Accessibility Standards,” for compliance with Section 508.1 Thereafter, technology companies seeking government contracts had to align their products with legal requirements.

Section 508 contains four subparts. Our study focused on Subpart B, “Technical Standards,” which defines rules governing six product classifications: software applications and operating systems, Web-based intranet and Internet information and applications, telecommunications products, video and multimedia products, self-contained/closed products, and desktop and portable computers.

Reference

mappings, of which 354 were refinements (S-R), 31 were generalizations (S-G), and 250 were similar actions (S-E). There were 38 NCSU and 19 Cisco requirements that didn’t map between the sets. However, every paragraph in the Section 508 standards aligned with at least one NCSU and one Cisco requirement.

The study also yielded 555 phrase assertions: 183 refined concepts (P-R1), 50 new constraints (P-R2), 134 generalized concepts (P-G1), 132 missing constraints (P-G2), and 56 modality changes (P-M) among the Cisco requirements.

Compliance Challenges
Details on the validation of the eight metrics and the gap analysis results are available in the study paper.2 Here, we summarize general findings relevant to specifying products.

First, we found that none of the Cisco requirements were true refinements, such that if the S-R statement metric applied, then only P-R phrase metrics applied. Nor were they true generalizations, such that if the S-G statement metric applied, only P-G phrase metrics applied.

This finding demonstrates a key challenge that developers and regulators face. Specifically, contextualizing product requirements for a domain provides guidance that developers need, but it also introduces new language that simultaneously generalizes and refines regulatory concepts and terms. In this process, developers might overlook nuances in legal language that affect compliance requirements. They might also generate unintended requirements that are costly and can stifle innovation.

Product-Focused Regulations
In contrast to a privacy standard such as HIPAA, which is written and organized with stakeholders as the primary focus, the Section 508 accessibility standards are organized primarily according to products (see the “Accessibility and Section 508” sidebar). This product focus adds to the inherent ambiguity of legal language. For example, only 34 percent of the NCSU requirements explicitly name the requirement’s grammatical subject. This means developers must determine whether the requirement applies to a product and its functions or stakeholders and their actions. For example, §1194.21(c) requires an input focus “that moves among interactive interface elements as the input focus changes,” but it doesn’t indicate what subject is changing focus—the product or the user.

Another observation that coincides with the product focus is the organization of standards around product classes as opposed to product features. For example, the accessibility standards include three product classes: software applications (§1194.21), Web applications (§1193.22), and telecommunications products.
For innovative products, such as voice over IP using a Web-based interface, product requirements must address each class. The S-E metric identified 55 redundancies across Section 508 product classes.

Compliance Patterns

We also identified several compliance patterns by sorting the investigators’ logical assertions according to the metrics. We believe these patterns can help regulators write clearer regulations while also improving developers’ compliance coverage and innovation management within a legal framework.

For example, legal requirements present preconditions either as phrases that follow condition keywords (if, when) or as classes of products or services to which requirements apply. S-E requirement pairs that include either the P-G2 or P-G1 phrase metric can identify these conditions. Developers can use this pattern to simplify compliance by removing preconditions or generalizing the terms of permissions and obligations. Applying this pattern incurs the potential trade-off costs of exceeding legal requirements. On the other hand, it might also better achieve the intent of the societal goal or better protect the developer and company when their products innovate in ways that regulators can’t foresee.

Another compliance pattern emerges from “terms of art,” which US laws and regulations define as precise phrases having specialized meanings in specific subject areas. Government laws and regulations intentionally define these terms broadly to support marketplace variabilities. Grounding these terms in the domain variants that practitioners use can help align legal and product requirements. For example, the term of art “telecommunications,” which appeared in Section 508 in the late 1980s, was later refined by the variant “voice over IP” in the late 1990s. In our study, the phrase metric P-R1 frequently coincided with these variants. In any case, developers should list important terms of art in their domain and maintain them for their product requirements.

Refining the legal language of broad societal goals into product requirement specifications is a complex, dynamic process that inherently generates gaps in meaning. Identifying these gaps gives developers a more complete, comprehensible view of the compliance landscape. We found working with a community of practice necessary to contextualize legal compliance in practice. A formal method to acquire legal requirements, such as FBRAM, ensures traceability, thus improving how companies demonstrate due diligence and good faith under the rule of law.

Acknowledgments

This article is based on a paper published with detailed study results in the Proceedings of the 16th IEEE International Requirements Engineering Conference. The work was supported by the IBM PhD Fellowship (RTP CAS) and US National Science Foundation grants 032–5269 and 043–0166.

References


Travis D. Breaux is a research staff member at the Institute for Defense Analyses in Alexandria, Virginia. His research interests include requirements engineering for software systems that are regulated by policies and laws. Breaux has a PhD in computer science from North Carolina State University, where he performed the work reported in this article. Contact him at tbreaux@ida.org.

Annie I. Antón is a professor in North Carolina State University’s Computer Science Department and a member of its Cyber Defense Lab. Her research focuses on methods and tools to support the specification of complete, correct behavior of software systems used in environments that pose loss risks as a consequence of failures and misuse. Antón has a PhD in computer science from the Georgia Institute of Technology. Contact her at aianton@ncsu.edu.

Kent Boucher is an accessibility technical lead with Cisco Systems. His research interests include universal design, software engineering, and cognitive psychology. Boucher has a BS in quantitative analysis of psychology from Northwest Missouri State University. Contact him at kboucher@cisco.com.

Merlin Dorfman is a quality systems staff engineer with Cisco Systems until his retirement on 24 September 2009. His research interests include software process improvement, software system engineering, and software requirements engineering and management. Dorfman has a PhD in aerospace engineering from Stanford University. Contact him at dorfman@computer.org.