Extracting Rights and Obligations from Regulations: Toward a Tool-Supported Process

Nadzeya Kiyavitskaya
Nicola Zeni
Dept. of Information and
Communication Technology
University of Trento, Italy
{nadzeya, nzeni}@dit.unitn.it

Travis D. Breaux
Computer Science Dept.
College of Engineering
North Carolina State
University, U.S.A.
tdbreaux@ncsu.edu

Annie I. Antón
Computer Science Dept.
College of Engineering
North Carolina State
University, U.S.A.
aianton@ncsu.edu

James R. Cordy
School of Computing
Queens University, Kingston,
Canada
cordy@cs.queensu.ca

Luisa Mich
Dept. of Computer and
Management Sciences
University of Trento, Italy
luisa.mich@unitn.it

John Mylopoulos
Dept. of Information and
Communication Technology
University of Trento, Italy
jm@science.unitn.it

ABSTRACT

Security, privacy and governance are increasingly the focus of government regulations in the U.S., Europe and elsewhere. This trend has created a "regulation compliance" problem, whereby companies and developers are required to ensure that their software complies with relevant regulations, either through design or reengineering. We previously proposed a methodology for extracting stakeholder requirements, called rights and obligations, from regulations. In this paper, we examine the challenges of developing tool support for this process. We apply the Cerno framework for textual semantic annotation to propose a tool for semi-automatic semantic annotation of concepts that constitute sources of requirements.

Categories and Subject Descriptors

D.2.1 [**Requirements/Specifications**]: Tools; I.7.5 [**Document and Text Processing**]: Document Capture—*Document analysis*; K.5.2 [**Legal Aspects of Computing**]: Governmental Issues—*Regulation*

General Terms

Legal Aspects, Management, Experimentation

Keywords

regulation compliance, privacy requirements, tool support

1. INTRODUCTION

Aligning software requirements and government laws, regulations and policies constitutes a problem of major importance for software systems that collect, manage, and use sensitive information [1]. In Canada, Europe and the U.S., legislation sets standards for organizations [3]. These standards are written in "legalese", which makes acquiring requirements from regulations a challenging task [7]. Legalese contains heavily qualified phrases that are laden with ambiguities, a pervasive phenomenon with natural languages in general [4]. The size of these documents and the large numbers of internal and external references to other sections of the same document or different laws that may take precedence further compounds the difficulty in analyzing them. In this paper, we focus on the challenges engineers face in analyzing prescriptive sentences. If engineers misinterpret these sentences, for example by overlooking an exception or condition in a regulatory rule, incorrect rights or obligations may be conferred on some stakeholders. Thus, extracting requirements from regulations is a major challenge in need of methodological aids and tools.

The process we envision for extracting requirements from regulations consists of three steps: (1) regulatory text is annotated to identify text fragments describing actors, rights, obligations, etc.; (2) a semantic model is constructed from these annotations; and (3) the semantic model is transformed into a set of functional and non-functional requirements. The first two steps are currently supported by Breaux and Antón's systematic, manual process for deriving semantic models from policies and regulations called Semantic Parameterization [5, 7].

Our work further supports this process with tools that seek to improve the productivity, quality and consistency of the manual process. In this paper, we address the first step of the process: the annotation of regulatory text to identify basic concepts such as rights and obligations. To achieve this goal, we have adopted the Cerno framework [12] for semantic annotation. The framework initially requires the manual construction of grammatical rules to identify basic concepts, on the basis of which it provides automated assistance to engineers. Each such rule defines a pattern that characterizes instances of a concept, such as an obligation. In this paper, we discuss the integration of these rules into a tool and the preliminary evaluation of the tool using the U.S. Health Insurance Portability and Accountability Act[1] (HIPAA).

The paper is organized as follows: in Section 2, we briefly overview Breaux et al.'s methodology for extracting rights and obligations and describe the Cerno framework. In Section 3 we present the new tool-supported process which adapts some features of the Cerno

ASE '07 Atlanta, Giorgia, USA

[1] U.S. Public Law 104-191, 110 Stat. (1996)

framework, and in Section 4 we present the design and evaluation of a case study. Related work appears in Section 5 and our conclusions in Section 6.

2. SEMANTIC ANNOTATION PROCESS

Our new tool-supported process is based on a previously proposed methodology for extracting stakeholder requirements from regulations [7]. In this methodology, requirements engineers mark regulatory text using phrase heuristics [6], [7] to identify rights or obligations, associated constraints, and condition keywords including conjunctions. These marked elements are codified as follows: each rule statement (a right or obligation) is followed by the originating paragraph reference for traceability, and the propositional formula that is comprised of associated constraints.

Cerno is a framework for generating semi-automatically annotations [12] and based on a lightweight text analysis approach that is implemented in a structural transformation system TXL [9]. Cerno's architecture and performance are described in [11]. To annotate input documents, Cerno uses context-free grammars, generates a parse tree, and applies transformation rules to generate output in a target format [11]. As discussed herein, by employing Cerno, we are better able to provide the required inputs to the Semantic Parameterization process.

The process for generating semantic annotations in Cerno is based on the design recovery process in software reverse engineering and uses a series of successive transformation steps [12]:

- **Step #1**. A parse tree is produced from the document structural grammar. This parse is coarse-grained and consists of structures such as "document", "paragraph", "phrase" and "word"; ignoring the linguistic structure below the phrase level. These trees are described by an ambiguous context-free TXL grammar using a BNF-like notation.
- **Step #2**. Annotations are inferred using a domain-dependent annotation schema. The schema contains a list of tags for concepts to be identified, selected from a domain-dependent semantic model and a vocabulary of indicators related to each concept. Cerno assumes that the annotation schema is constructed beforehand either automatically using learning methods or manually in collaboration with domain experts.
- **Step #3**. Annotated text fragments are selected with respect to a predefined database schema and stored in an external database. The database schema embodies the desired output format. It is manually derived from the domain-dependent semantic model and represents a set of fields of a target database. The final products of Cerno can be both marked up text, i.e., *in-line* annotation, and the populated database, i.e., *stand-off* annotation.

Similar to Cerno, the methodology of Breaux and Antón uses a number of phrase heuristics that guide the process of identification of rights or obligations [6, 7].

3. ADAPTING CERNO TO THE REGULATIONS DOMAIN

Adapting Cerno's framework to a different domain requires a domain-specific annotation schema describing the primary assumptions about the relevant entities and their inter-dependencies. The annotation schema used in this preliminary study was limited to extracting a set of "objects of concern": *right, anti-right, obligation, anti-obligation, exception* [7], and some types of *constraints*.

These terms are defined as follows:

- A *right* is an action that a stakeholder is conditionally permitted to perform.
- An *obligation* is an action that a stakeholder is conditionally required to perform.
- In contrast, *anti-rights* and *anti-obligations* state that a right or obligation does not exist.
- A *constraint phrase* is the part of a rule statement that describes a single pre-condition.

Manual analysis of the HIPAA document yielded a list of normative phrases that identify many of these objects of concern (see a fragment in Table 1) [7]. All the normative phrases were used as domain-dependent indicators in Cerno's annotation process. A few indicators are complex patterns that combine both literal phrases and general concepts. The identified normative phrases assume a preliminary recognition of the following basic constructs: (1) *cross-reference*: a citation of some legal document or a reference to a part of the same document; (2) *policy*: the name of the law, standard, act or other regulation document which establishes rights and obligations; (3) *actor*: can be an individual or an organization involved.

Table 1: Normative phrases in HIPAA

No	Concept type and its indicators
	Right
1	<actor>... < /actor> may
2	<actor>... < /actor> can
3	<actor>... < /actor> could
4	<policy>... < /policy> permits
5	<actor>... < /actor> has a right to
6	<actor>... < /actor> should be able to
	Anti-Obligation
1	<policy>... < /policy> does not restrict
2	<policy>... < /policy> does not require
3	<actor>... < /actor> must not
4	<actor>... < /actor> is not required

To identify these objects, we extended the parse step of Cerno's framework with new object grammars. We consider two types of cross-references that appear in the HIPAA: internal references that refer the reader of a regulation to another paragraph within the regulation; and external references that refer to another regulation, act or law. Internal cross-references are consistently identified by Cerno using a small set of patterns. To recognize instances of the *actor* and *policy* concepts, we exploit the consistent use of terms and definitions in the HIPAA document. Many regulations and policies, including the HIPAA, strictly define these terms and provide hyponyms (e.g., related kinds). A fragment of HIPAA Section 160.103 "Definitions of HIPAA" is shown in Fig. 1.

In the sections that we analyzed, we found other terms that we could generalize into a common, abstract type, including *event*, *date*, and *information*. Thus, on the basis of the definition section, we derived a list of hyponyms for the basic concepts: *actor* and *policy* as well as *event*, *date* and *information*. Finally, the new Cerno-based regulation analysis process is organized into three main phases: (1) recognition of structural elements of the document: section boundaries which are numbered and titled, sentence boundaries; (2) identification of basic objects: actor, policy, event, date, information and cross-reference; (3) deconstruction of a rule statement to identify phrases and constraints.

```
Actor: ANSI, business associate(s), covered
entit(y|ies), HCFA, HHS, <...>;
Policy: health information, designated record
set(s), individually identifiable health information,
protected health information, psychotherapy notes;
```

Figure 1: A part of indicators for basic entities according to the information in the definitions section

4. EMPIRICAL EVALUATION

We first discuss the challenges to automated annotation posed by regulatory texts, before describing our experimental design and evaluation. U.S. Federal regulations, including the HIPAA, are highly structured and written in a specialized language called "legalese". Despite this apparent structure, the legalese is not always used consistently, contains ambiguities, and frequently elaborates requirements at different levels of detail.

This structure also presents several traceability challenges. Rights and obligations do not always appear in separate statements; they may be intermixed, distributed or refined across different statements. In our pilot tool, we address this problem by introducing a fine-grained identification of phrase fragments that relate to a right or obligation. Thus, we assume that one sentence may have fragments related to different rights or obligations. The extracted models are later checked for redundancies that are present in the original document and identified by our process.

Another challenge is identifying the subject for sentence fragments that appear sub-paragraphs. In linearly written regulatory texts, the sentence fragments will semantically relate to the last proposition that appears in the text. To address this challenge, we use a heuristic that links each listed item of the same level with the last (annotated) phrase of the level above.

Finally, cross-references to other regulations pose a significant challenge; these cross-references elaborate and prioritize requirements [7] and may be difficult to disambiguate because cross-references can be circular. At this preliminary stage, we simply annotated each cross-reference in the document such that it can be manually resolved later using the markup of the hierarchical document structure.

After extending Cerno as discussed in Section 3, we applied it to the full text of the HIPAA Privacy Rule [10] consisting of two parts, numbered 160 ("General Administrative Requirements") and 164 ("Security and Privacy"). The automatic annotation of the HIPAA Privacy Rule, containing a total of 33,788 words, by the Cerno framework takes only 3.07 seconds on a personal computer Intel Pentium 4, 2.60 GHz processor, RAM 512 MB, running Windows XP operating system. As a result, about 1900 basic entities and 140 rights and obligations were identified.

Due to the lack of a gold standard (i.e., a reference annotated document to compare with), the annotation quality must be manually evaluated and was limited to Sections 164.520, 164.522, 164.524, and 164.526. We chose these sections because we can compare the Cerno results to the manual results reported by Breaux et al. [7]. Those results covered a total of 5,978 words or 17.8% of HIPAA and were obtained in about 2.5 hours per section.

The preliminary analysis of the resulting annotations for 164.520 is summarized in Table 2. The number of rights (R), obligations (O), constraints (C) and cross-references (CR) is reported for the manual process [7] and for Cerno.

There are several notable distinctions that we can discuss at this stage of the analysis. Section 164.520 contains statements for stakeholder rights that begin in one paragraph and continue into a subparagraph. The latter-half of these statements is called a *continuation*, in general. Due to continuations, there are two false-positives in the number of rights and obligations reported. Furthermore, paragraphs 164.520(b)(1) and (b)(2) describe so-called "content requirements" that detail the content of privacy notices. These requirements were not included in the number of stakeholder rights and obligations reported by Breaux et al. [7]. Cerno identified four stakeholder rights in these two paragraphs. The total number of constraints was limited to those due to internal cross-references.

Table 2: Number of rights, obligations, constraints and cross-references found in HIPAA

Section	R	O	C	CR
164.520 (Manual)	9	17	54	37
164.520 (Cerno)	12	15	5	31

The tool correctly identified nearly all instances of the concepts actor, policy, event, information and date. It also correctly recognized section and subsection boundaries, titles and annotated paragraph indices. These annotations may be reused to disambiguate and manage cross-references and may provide useful input for the Semantic Parameterization Process. The Cerno-based tool adapted to the domain of regulatory texts largely reduces human effort and time spent for analysis by facilitating recognition of relevant text fragments.

Nevertheless, as a result of our experimental study, we observed a number of current limitations of Cerno that should be addressed in future work: (1) Additional types of constraints should be considered. (2) For the concepts expressing constraints, the correct subject or object to which they apply must be identified. At present, Cerno can facilitate this by identifying constraint phrases and likely subject candidates for manual analysis later. (3) Identification of the subjects of conjunctions or disjunctions is problematic even for full-fledged linguistic analysis tools. We propose to extend the tool to highlight such cases and prompt a human analyst to manually resolve them. Each of these issues is planned to be revised and appropriately elaborated in the next build of the Cerno tool.

The empirical validation of our tool sought to test the ability of non-experts to analyze regulations and generate requirements specifications for a new software system. We selected section 164.520 of HIPAA for annotation by a different group of people, who are not working with rules and regulations directly. We provided the participants two parts of section 164.520 (containing a total of 2269 words) to annotate: one was the original text and the other included annotations generated by Cerno. These parts were selected to have an near equal number of statements. The participants were asked to identify rule statements and phrases in each of the two parts, inserting markup in the original page for the unannotated part, and modifying Cerno's annotations in the remaining part.

From the quantitative data that we collected, we observed: (1) when participants were assisted by the automatic support, the total number of entities identified was about 10% more than when starting from the original document; (2) when assisted by the automatic support, participants were faster by about 12.3%. All participants expressed satisfaction with the tool-provided annotations, finding them useful to read and interpret the document. The low number of false results likely contributed to this observation. However, certain improvements must be realized in the future to render the annotations more helpful to the final users.

5. RELATED WORK

The idea of using contextual patterns or keywords to identify relevant information in prescriptive documents is not new. A number of methodologies based on similar techniques have been developed. However, tools to realize and synthesize these methods under a single framework are lacking.

In [8], the authors suggested an algorithm for detecting and classifying non-functional requirements (NFRs). In a pilot experiment, the indicator terms were mined from catalogs of operationalization methods for security and performance softgoal interdependency graphs. These indicators were then used to identify NFRs in fifteen requirements specifications. The results have shown a satisfactory recall and precision for the security and performance keywords.

To facilitate reasoning with regulations, Antoniu et al. [2] introduced the regulations analysis method based on defeasible logic rules. The method manually acquires facts from regulations using defeasible theory.

Several approaches to requirements analysis are relevant. LIDA [13] is an iterative model development method based on linguistic techniques. It uses a part-of-speech tagging to derive instances of the UML abstractions. In [15], the authors analyzed NASA requirements documents and made several recommendations on writing clear requirements specifications. The authors discovered that good requirements specifications use imperative verbs, such as "shall", "must", "must not". In aspect-oriented requirements engineering, the EA-Miner [14] tool supports separation of aspectual and non-aspectual concerns by applying natural language processing techniques to requirements documents.

6. CONCLUSIONS

Regulations and policies constitute rich sources of requirements for software systems that must comply with these normative documents. In order to facilitate alignment of software system requirements and regulations, systematic methods and tools automating regulations analysis must be developed. This paper presents a tool intended to provide automatic support for analyzing policy documents. The new tool-supported process exploits the findings of our earlier work on requirements analysis, and uses the Cerno framework to yield annotations marking instances of concepts found in regulation texts. These instances include rights and obligations that must be incorporated into software requirements to comply with the law.

In summary, the proposed tool has demonstrated promising results with limited effort required to adapt it to a specific regulation document. Although, the phrase heuristics used are limited to the HIPAA document and may need revision when analyzing other regulations and policies, we believe that our tool supported process can be re-used in a different domain due to its modularity. Further extensions and experimental evaluation are planned and being realized.

7. REFERENCES

[1] A. I. Antón, J. B. Earp, and R. A. Carter. Precluding incongruous behavior by aligning software requirements with security and privacy policies. *Information & Software Technology*, 45(14):967–977, 2003.

[2] G. Antoniou, D. Billington, and M. J. Maher. On the analysis of regulations using defeasible rules. In *Proc. of HICSS'99*, volume 6, page 6033, Washington, DC, USA, 1999. IEEE Computer Society.

[3] H. Berghel. The two sides of ROI: return on investment vs. risk of incarceration. *Commun. ACM*, 48(4):15–20, 2005.

[4] D. Berry, E. Kamsties, and M. M. Krieger. *From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity*. Technical Report, School of Computer Science, University of Waterloo, Waterloo, ON, Canada, 2003.

[5] T. D. Breaux and A. I. Antón. Analyzing goal semantics for rights, permissions, and obligations. In *Proc. of RE'05*, pages 177–186, 2005.

[6] T. D. Breaux and A. I. Antón. Mining rule semantics to understand legislative compliance. In *Proc. of WPES'05*, pages 51–54, New York, NY, USA, 2005. ACM Press.

[7] T. D. Breaux, M. W. Vail, and A. I. Antón. Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In *Proc. of RE'06*, pages 46–55, Washington, DC, USA, 2006. IEEE Computer Society.

[8] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc. The detection and classification of non-functional requirements with application to early aspects. In *Proc. of RE'06*, pages 36–45, Washington, DC, USA, 2006. IEEE Computer Society.

[9] J. R. Cordy. The TXL source transformation language. *Science of Computer Programming*, 61(3):190–210, 2006.

[10] U. S. Goverment. Standards for privacy of individually identifiable health information, 45 CFR part 160, Part 164 subpart E. *In Federal Register*, 68(34):83348381, Feb. 20, 2003.

[11] N. Kiyavitskaya, N. Zeni, L. Mich, J. R. Cordy, and J. Mylopoulos. Applying software analysis technology to lightweight semantic markup of document text. In *Proc. of ICAPR 2005*, pages 590–600, 2005.

[12] N. Kiyavitskaya, N. Zeni, L. Mich, J. R. Cordy, and J. Mylopoulos. Text mining through semi automatic semantic annotation. In *Proc. of PAKM'06*, volume 4333 of *LNCS*, pages 143–154. Springer-Verlag, 2006.

[13] S. P. Overmyer, B. Lavoie, and O. Rambow. Conceptual modeling through linguistic analysis using lida. In *Proc. of ICSE'01*, pages 401–410, Washington, DC, USA, 2001. IEEE Computer Society.

[14] A. Sampaio, R. Chitchyan, A. Rashid, and P. Rayson. EA-Miner: a tool for automating aspect-oriented requirements identification. In *Proc. of ASE'05*, pages 352–355, New York, NY, USA, 2005. ACM Press.

[15] W. M. Wilson, L. H. Rosenberg, and L. E. Hyatt. Automated analysis of requirement specifications. In *Proc. of ICSE'97*, pages 161–171, New York, NY, USA, May 1997. ACM Press.