Automating the Extraction of Rights and Obligations for Regulatory Compliance

Nadzeya Kiyavitskaya[1], Nicola Zeni[1], Travis D. Breaux[2], Annie I. Antón[2], James R. Cordy[4], Luisa Mich[3], and John Mylopoulos[1]

[1] Dept. of Information Engineering and Computer Science, University of Trento, Italy
{nadzeya, nzeni, jm}@disi.unitn.it
[2] Dept. of Computer Science, North Carolina State University, U.S.A.
{tdbreaux, aianton}@ncsu.edu
[3] Dept. of Computer and Management Sciences, University of Trento, Italy
luisa.mich@unitn.it
[4] School of Computing, Queens University, Kingston, Canada
cordy@cs.queensu.ca

Abstract. Government regulations are increasingly affecting the security, privacy and governance of information systems in the United States, Europe and elsewhere. Consequently, companies and software developers are required to ensure that their software systems comply with relevant regulations, either through design or re-engineering. We previously proposed a methodology for extracting stakeholder requirements, called rights and obligations, from regulations. In this paper, we examine the challenges to developing tool support for this methodology using the Cerno framework for textual semantic annotation. We present the results from two empirical evaluations of a tool called "Gaius T" that is implemented using the Cerno framework and that extracts a conceptual model from regulatory texts. The evaluation, carried out on the U.S. HIPAA Privacy Rule and the Italian accessibility law, measures the quality of the produced models and the tool's effectiveness in reducing the human effort to derive requirements from regulations.

1 Introduction

In Canada, Europe and the United States, regulations set industry-wide rules for organizational information practices [1]. Aligning information systems requirements with regulations constitutes a problem of major importance for organizations. These regulations are written in legal language, colloquially referred to as *legalese*, which makes acquiring requirements a difficult task for software developers who lack proper training [2]. In this paper, we focus on the challenges software engineers face in analyzing regulatory rules, called rights and obligations. If engineers misinterpret these sentences, for example by overlooking an exception or condition in a regulatory rule, incorrect rights or obligations may be

conferred to some stakeholders. Thus, extracting requirements from regulations is a major challenge in need of methodological aids and tools.

The tool-supported process that we envision for extracting requirements from regulations consists of three steps: (1) text is annotated to identify 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 nonfunctional requirements. The first two steps are currently supported by Breaux and Antón's systematic, manual methodology for acquiring legal requirements from regulations [3], [2], [4]. In this process, the requirements engineer marks the text using phrase heuristics and a frame-based model [5], [3] to identify rights or obligations, associated constraints, and condition keywords including natural language conjunctions [2]. These rights and obligations may be restated into restricted natural language statements [2], after which the rules can be modeled in Description Logic using the Semantic Parameterization process [4]. This Description Logic model can be queried and analyzed for ambiguities and conflicts [4]. Our work seeks to add tool support to this process to improve productivity, quality and consistency in the first step of the output. To achieve this goal, we adopt the Cerno framework [6] for semantic annotation. The framework initially requires the construction of linguistic markers to identify various concepts, on the basis of which it provides automated assistance to engineers.

The Cerno framework has been extended to deal with some of the complexities of regulatory text. The resulting extension is a new tool called *Gaius T*.[5] The contributions of this paper are to present Gaius T with an empirical evaluation that compares performance of Gaius T with the performance of human analysts using two regulatory documents written in different languages: the U.S. Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule [7] and the Italian accessibility law (the Stanca Act) [8]. These contributions expand upon a short paper [9], in which we first outlined our preliminary research plan and first experiment with Gaius T.

The remainder of the paper is organized as follows. Section 2 discusses specific challenges that must be addressed by any tool supported process in the domain of regulations and policies, including Gaius T. In Section 3, we describe the Cerno annotation framework and introduce the new tool-supported process with Gaius T. Section 4 presents the design and evaluation through two case studies, with related work appearing in Section 5 and our conclusion in Section 6.

2 Complexity of Regulatory Texts

A number of challenges complicate the automated annotation of regulatory texts. For example, U.S. federal regulations are highly structured and written in legalese. Despite this structure, the conventions of legalese are not always used consistently, there are intended and unintended ambiguities, and individ-

[5] Named after Gaius Terentilius Harsa, a plebeian tribune who played an instrumental role in establishing the first formal code of laws through the Twelve Tablets in ancient Rome (462BC) (http://en.wikipedia.org/wiki/Terentilius)

ual requirements are described across multiple sentences and paragraphs using cross-references. We now discuss several of these challenges.

Legalese written in different languages and by different legislatures introduce variability that must be addressed by automated tools. For example, the Italian language uses more accents and apostrophes than the English language, which affects how tools recognize important phrases. Similarly, Italian and English use different natural language grammars to express rights and obligations. In addition, the U.S. HIPAA Privacy Rule and the Italian Stanca Act use different document structures that affect the identification of rights and obligations. As a result, text processing tools that employ rules based on keywords, phrases and syntax cannot be naively adapted to other languages and jurisdictions without addressing these important issues.

In regulations, individual requirements can be elaborated in multiple sentences, intermixed into a single sentence or distributed across multiple paragraphs. For example, the HIPAA paragraph 164.528(a)(2)(ii) contains three sub-paragraphs (A), (B), and (C) in one sentence: "`the covered entity must:(A)...;(B)...; and (C)...`", in which each sub-paragraph describes a separate, obligated action. This hierarchical sub-paragraph structure presents several traceability challenges that our tool addresses by either identifying the subject from an encapsulating paragraph that relates to requirements stated in sub-paragraphs or by identifying which phrase fragments relate to a requirement in an encapsulating paragraph.

Cross-references to other regulations is further complicate matters. These cross-references elaborate [3], [2] and prioritize requirements [3] and may be difficult to disambiguate because cross-references can appear to be syntactically circular. For instance, HIPAA paragraph 164.528 (a)(2)(i) describes an obligation to suspend a right of an individual. This right is elaborated in a separate paragraph, denoted by the phrase "`as provided in 164.512(d)`". In paragraph 164.528(a)(2)(ii) that follows, the phrase "`pursuant to paragraph (i)`" refers back to the previous paragraph. Using Gaius T, each cross-reference in the document is annotated in such way that it can be browsed later using markup of the hierarchical document structure.

Finally, policies and regulations are *prescriptive* [10] rather than descriptive. Because stakeholders cannot afford to overlook regulatory requirements, a higher precision and recall for annotation or text-mining is required in this domain. We address this issue in the empirical evaluation described later in this paper.

3 Semantic annotation process

This section introduces the Cerno framework for semi-automatic semantic annotation and also presents the Gaius T extension, intended specifically for the annotation of regulatory text [6].

3.1 The Cerno Framework

Cerno is based on a lightweight text analysis approach that is implemented using the structural transformation system TXL [11]. The architecture and the per-

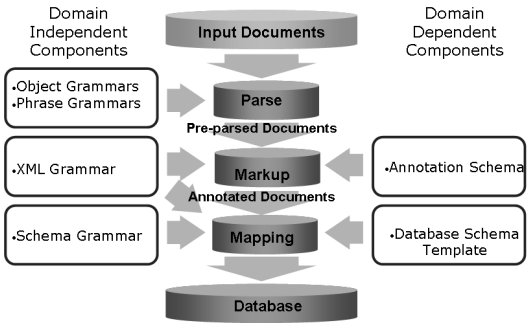

Fig. 1. Semantic annotation process in Cerno

formance of the tool are described in detail in a previous paper [6]. To annotate input documents, Cerno uses context-free grammars to generate a parse tree before applying transformation rules, which generate output in a pre-specified format.

The process for generating semantic annotations in Cerno is based on a "design recovery" process borrowed from software reverse engineering [12]. As shown in Fig. 1, this process uses a series of successive transformation steps:

Step #1 The input document is parsed in accordance with the document structural grammar and a parse tree is produced. The parse result consists of structures such as "document", "paragraph", "phrase" and "word". The grammar is described as an ambiguous context-free TXL grammar using BNF-like notation (see an example in the next subsection in Fig. 2).

Step #2 Annotations are inferred using a domain-dependent annotation schema. This schema contains a list of tags for concepts to be identified, selected from the domain semantic model, and a vocabulary of indicators related to each concept. Cerno assumes that the annotation schema is constructed beforehand either automatically using some learning methods or manually in collaboration with domain experts. Indicator lists may include literal words (see further in Fig. 5) or names of parsed entities. They also can be *positive*, pointing to the presence of the given concept, or *negative*, pointing to the absence of this concept.

Step #3 Annotated text fragments are selected with respect to a predefined database schema template and stored in an external database. The database schema template embodies the desired output format. It is manually derived from the domain semantic model and represents fields of a target database.

Similar to Cerno, the methodology of Breaux and Antón uses a number of phrase heuristics that guide the process of identifying rights or obligations [2]. We encode these heuristics into Cerno's domain-dependent components and enrich the framework with other domain- and task-specific knowledge. In this way, we can facilitate the generation of a requirements model. Moreover, we seek to formalize specific characteristics of legal documents and test the generality

Table 1. Normative phrases in HIPAA

Concept type	Indicators
Right	<actor>...</actor> may ; <actor>...</actor> can ; <actor>...</actor> could ; <policy>...</policy> permits ; <actor>...</actor> has a right to ; <actor>...</actor> should be able to
Cross-Reference Constraint	set by <cross-reference> ;...

of our framework. The extensions to the Cerno framerwork for legal documents are further referred to as *Gaius T*.

3.2 Gaius T for HIPAA

To evaluate Gaius T, we first annotate a fragment of the HIPAA Privacy Rule in order to identify instances of rights, obligations, and associated constraints, and then we evaluate the quality of the annotations obtained. The "objects of concern" that we annotate consist of: *right, obligation, exception*, and some types of *constraints* [2], in which 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; a *constraint* is the part of a right or obligation that describes a single pre- or post-condition, and *exceptions* remove elements from consideration in a domain.

The manual analysis of the Privacy Rule yielded a list of normative phrases that identify many of these objects of concern [2], see examples in Table 1. All the normative phrases were employed as positive indicators in the domain-dependent indicators of Cerno's Markup step. Some of the indicators are complex patterns which combine both literal phrases and general concepts, thus assuming a preliminary recognition of several basic constructs: *cross-references* can be internal references that refer the reader of a regulation to another paragraph within the same regulation or external references, a citation of another regulation, act or law; *policy* is the name of the law, standard, act or other regulation document which establishes rights and obligations; and *actor* is an individual or an organization involved. To recognize these objects, we extended the parse step of Cerno with the corresponding object grammars.

Internal cross-references are consistently formatted throughout the Privacy Rule which results in consistent identification by the tool using a set of patterns shown in Fig. 2. However, due to the variety of reference styles used by different laws, it is necessary to extend these patterns when analyzing a new law, as we observed during our analysis of the Italian accessibility law.

To recognize instances of the actor and policy concepts, we exploit the fact that the Privacy Rule uses standard terms, called a *term-of-art*, consistently throughout the entire document. These terms are ritually defined in a separate definitions section, such as HIPAA section 160.103 titled "Definitions of HIPAA". For example, it contains terms such as "policy", "business associate", "act", and "covered entity". Example indicators that are used to identify basic entities and that were derived from the definitions section are shown in Fig. 3.

```
define citation
   '§ [opt space] [number] [repeat enumeration] | 'paragraph [space] [repeat enumeration]
  | 'paragraph [opt space] [decimalnumber] [repeat enumeration]
  |[decimalnumber][repeat enumeration]
end define
define enumeration
  '( [id] ') | '( [number] ')
end define
```

Fig. 2. The grammar for cross-reference object

```
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; <...>;
```

Fig. 3. Indicators for basic entities

```
<Right>A <Actor>covered entity</Actor> may deny an <Actor>individual</Actor>'s
request for amendment,</Right> if it determines that the <Information>protected health
information</Information> or record that is the subject of the request:
<Index>(i)</Index> Was not created by the <Actor>covered entity</Actor>,
<Exception>unless the <Actor>individual</Actor> provides a reasonable basis to believe
that the originator of <Information>protected health information</Information> is no
longer available to <Policy>act</Policy> on the requested amendment </Exception> ...
```

Fig. 4. A fragment of the result generated by Gaius T for HIPAA Sec.164.526

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*.

The Gaius T regulatory analysis process for the Privacy Rule is organized into three main phases: (1) Recognition of structural elements of the document: section boundaries, section attributes which are number and title, sentence boundaries (see [13]); (2) Identification of basic objects: actor, policy, event, date, information and cross-reference; (3) Deconstruction of a rule statement to identify its components and constraints. Fig. 4 illustrates an excerpt of annotated text from HIPAA section 164.526(a)(2) resulting from the tool's application of Gaius T. Each embedded XML annotation is a candidate "object of concern." For instance, the "Index" annotation denotes the sub-paragraph index "(i)" and the Actor annotation denotes the "covered entity"; the latter appears twice in this excerpt.

3.3 Gaius T for Italian regulations

The Stanca Act [8] describes accessibility requirements governing all web sites of the Italian Public Administration to ensure accessibility for the disabled. The Act includes technical requirements and general restrictions that web sites must respect. The annotation schema for the accessibility law contains *right, anti-right, obligation, anti-obligation, exception*, and some types of *constraints*, where *anti-rights* and *anti-obligations* state that a right or obligation is not conferred by a specific law, respectively [2].

```
Obligation: dov[ere], è fatto obbligo, farla osservare, promuov[ere], comport[are],
costituiscono motivo di preferenza, defin[ire];
AntiObligation: non dov[ere], non sia, non si applica, non si possono stipulare, non
esprim[ere];
Right: po[sso|uoi|uò|ssiamo|tete|ssono|ssa];
AntiRight: non po[sso|uoi|uò|ssiamo|tete|ssono|ssa];
```

Fig. 5. A sample of the syntactic indicators used to identify categories in Stanca

For identification of actor instances in the Italian law, we adopted two solutions: (1) some instances were mined manually from the definition section "Definizioni"; (2) in order to acquire instances of actors not mentioned in the definitions, we exploited the results provided by a Part of Speech Tagger (POS) [14], i.e., all proper nouns we marked as actors. For resource instances, we followed only the first solution reusing the terms stated in the definition section.

In order to identify action verbs, we adopted the following heuristic: annotate all verbs in present tense, passive tense and impersonal tense. The verbs in the listed forms also refer to obligations, in accordance with the instructions for writing Italian legal documents [15]. Thus, the corresponding heuristic rule was adapted for identifying obligations.

For rights, obligations and their antitheses, it is more difficult to identify these statements in Italian than in English. For example, English modal verbs (must, may, etc.) are consistently used to state prescriptions, such as "`the users must present their request`," while Italian regulations use present active ("`gli utenti presentano la domanda`"), present passive ("`la domanda è presentata`") and impersonal tenses ("`la domanda si presenta`") of verbs to describe an obligation. The choice of the style highly depends on the individual lawmaker. Each of these styles is equally recommended by the law writing guidelines [15]. Therefore, in identification of rights and obligations, our strategy included: (1) translation of normative phrases identified for the HIPAA; (2) annotation of those sentences that contain verbs in the tenses that intrinsically express obligations as instances of obligation. A subset of the syntactic indicators for the Italian law is shown in Fig. 5 and a fragment of the annotated document in Fig. 6.

4 Empirical Evaluation

The proposed process for extracting rights and obligations was validated in a comparative evaluation that compared the number of automated annotations inferred by Gaius T with the number of manually derived annotations. For the HIPAA Privacy Rule, we also evaluated the productivity effect of using the tool. The comparative evaluation was difficult to realize because in many cases manual and automated annotations are not comparable because the granularity of these annotations differed.

4.1 The HIPAA document

After extending the framework as discussed in Section 3.2, we applied it to two sections of the HIPAA Privacy Rule [7]: 160 ("General Administrative Require-

```
Art. 10 (Regolamento di attuazione)
<Obligation>
1. <Constraint>Entro novanta giorni dalla data di entrata in vigore della presente
<Policy>legge</Policy></Constraint>, con <Policy>regolamento</Policy>emanato ai
sensi dell'articolo 17, comma 1, della <Policy>legge</Policy>23 agosto 1988, n. 400,
sono definiti:
a) i criteri e i principi operativi e organizzativi generali per l'accessibilità;
b) i <Resource>contenuti</Resource>di cui all'articolo 6, comma 2;
c) i controlli esercitabili sugli operatori privati che hanno reso nota l'accessibilità dei
propri siti e delle proprie <Resource>applicazioni</Resource>informatiche;
d) i controlli esercitabili sui <Actor>soggetti</Actor>di cui all'articolo 3, comma 1.
2. Il <Policy>regolamento</Policy>di cui al comma 1 è adottato previa consultazione
con le associazioni delle <Actor>persone disabili</Actor>maggiormente rappresentative,
con le associazioni di sviluppatori competenti in materia di accessibilità e di
produttori di <Resource>hardware</Resource>e <Resource>software</Resource>e
previa acquisizione del parere delle competenti Commissioni parlamentari,
<Constraint>che <Action>devono</Action>pronunciarsi entro quarantacinque giorni dalla
richiesta</Constraint>, e d'intesa con la Conferenza unificata di cui all'articolo 8 del
<Policy>decreto</Policy>legislativo 28 agosto 1997, n. 281.</Obligation>
```

Fig. 6. A fragment of the annotated accessibility law

Table 2. Comparative evaluation results for section 164.520 of HIPAA

	Rights	Obligations	Constraints	Cross-references
Gaius T	12	15	5	31
Human	9	17	54	37

ments") and 164 ("Security and Privacy"). Gaius T parsed 33,788 words and required 2.79 seconds on a personal computer based upon an Intel Pentium 4, 3 GHz processor, RAM 2 Gb, running Suse Linux. This results include over 1800 basic entities and 140 rights and obligations.

Due to the lack of a gold standard (i.e., a reference annotated document to compare with), the annotation quality was evaluated manually by comparing results acquire from section 164.520 "Notice of privacy practices for protected health information". We chose this section because we can compare the Gaius T results to the manual results reported by Breaux et al. [2]. The manual analysis by an expert analyst of the reported fragments, containing a total of 5,978 words or 17.8% of the Privacy Rule, took an average of 2.5 hours per section. The preliminary analysis of the resulting annotations for section 164.520 is summarized in Table 2. The number of rights, obligations, constraints and cross-references is reported for the manual process [2] and for Gaius T.

There are several notable distinctions that we can discuss at this stage of the analysis. Section 164.520 contains stakeholder rights whose description begins in one paragraph and continues into a sub-paragraph. The latter-half of these rights, and likewise for obligations, is called a *continuation*. 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. and were not included in the number of stakeholder rights and obligations report by Breaux et al. [2]. Gaius T identified four stakeholder rights in these two paragraphs. The total number of constraints was limited to those due to internal cross-references.

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 manage cross-references and may provide useful input for the Semantic Parameterization process. Gaius T largely reduces human effort and time spent for analysis by facilitating recognition of relevant text fragments.

In addition to the expert evaluation, we conducted an experiment inexperienced users using Gaius T. The goal of this study was to test the usefulness of the tool for non-experts in the regulatory text who may have to analyze such documents to generate requirements specifications for a new software system. The problem is that requirements engineers are not always supported by lawyers when designing new software. For this purpose, we selected section 164.520 of Privacy Rule for annotation by a different group of people, who are not working with rules and regulations directly. The experiment involved four junior researchers from the software engineering area, two of whom were not from the group working on the tool. We motivated the participants by paying a wage per hour of their work. All participants were non-native English speakers, received the same training in semantic annotation for one hour, but none of them had earlier participated in legal document analysis. A detailed explanation of the annotation process and examples of the concepts to be identified were available. Moreover, the participants were provided with a user-friendly interface to facilitate insertion and modification of tags in the input documents.

In this experiment, the participants were given two different parts of section 164.520 to annotate, one of which was original text and the other augmented with annotations generated by Gaius T. These parts were selected in such a way as to have an approximately equal number of statements and comprised 1,205 words and 1,057 words respectively. The annotators were asked to incrementally identify rule statements and their components in each of the two parts: first, inserting markups on the original page for the unannotated part, and second, modifying Gaius T's annotations in the part that was previously automatically annotated. We measured the time spent for annotation of both parts by each analyst and counted the number of different entities identified.

The results for this experiment are collected in Table 3 and include the number of entities collected by human annotators working with and without tool support. Observing this table, we notice that when annotators were assisted by Gaius T: (a) the total number of entities identified was about 10 percent larger than when starting from the original document; however, t-test results do not allow us to claim that this improvement is statistically significant; (b) annotators were faster by about 12.3 per cent. The part of analysis that the annotators found the most complicated and time-consuming was relating constraints contained in a rule statement to their corresponding subjects.

The evaluation results obtained thus far look promising, but larger studies must be conducted to prove the observed improvement is statistically significance. Most important, unlike human annotations, automatic annotations are more consistent and much faster, and thus show promise as the technology im-

Table 3. Number of extracted items for two fragments

	Fragment 1				Fragment 2			
	Without tool		With tool		Without tool		With tool	
	A1	A3	A2	A4	A2	A4	A1	A3
Obligations	10	2	13	13	9	12	10	13
Rights	3	9	0	2	6	4	2	1
Anti-Obligations	1	0	2	1	0	0	0	0
Anti-Rights	1	2	1	1	0	0	3	2
Constraints	36	23	18	16	36	32	41	19
Actors	45	14	56	19	22	11	17	50
Actions	25	14	27	18	28	22	24	44
Resources	32	34	29	14	22	14	31	27
Targets	1	5	4	0	9	10	11	5
Totals	154	103	150	84	132	105	139	161
Time in min	58	28	63	21	61	45	42	36

proves. Nevertheless, as a result of our experimental study, we observed a number of current limitations of Gaius T that should be addressed in future development of the tool:

- Additional types of constraints should be considered. The reason for missing some of constraints is that normative phrases for them are not explicitly provided by the manual methodology. Therefore the future development of the tool should involve revision of the annotation schema and indicators.
- Another problematic aspect in analyzing regulatory texts is that the concepts expressing constraints require correctly identifying the subject or object to which these constraints apply This task is difficult for human analysts, especially if related fragments are scattered over a long statement. However, Gaius T can facilitate their work by identifying a constraint phrase and subject candidates and then suggest to a human to connect the given constraint to the identified object that is most relevant.
- Identification of the subjects of conjunctions or disjunctions ("and", "or") must be completed for the Semantic Parameterization process. This task is problematic even for full-fledged linguistic analysis tools. In our case, we propose to extend the tool to highlight such cases and prompt a human analyst to resolve them manually.

4.2 The Italian Accessibility Law

After extending Gaius T with features intended to support the analysis of Italian law, we applied it to the full text of the Stanca Act, containing a total of 6,185 words. The automatic annotation required only 61 milliseconds on a personal computer Intel Pentium 4, 3 GHz processor, RAM 2 Gb, running Suse Linux. As a result, a total of 683 basic entities and 36 rights and obligations were identified.

Table 4 presents the results of this evaluation, consisting of the number of instances of the concepts of interest that the tool identified compared to a single human annotator. The tool outperformed the human annotator in identifying instances of the concepts actor, policy, action, and resource. As for complex

Table 4. Quantitative evaluation summary for the accessibility law

	Actors	Actions	Resources	Policies	Obligations	Anti-obligations	Rights	Anti-rights	Constraints
Gaius T	241	77	279	86	26	2	7	1	12
Human	170	55	58	3	24	2	9	0	32

concepts, the tool identified nearly all instances of rights and obligations, however the performance was essentially lower for the constraint concept.

There were difficulties in analyzing the Italian text for both the human annotator and the tool that emerged in this study. For example, the subject is frequently omitted, as in passive forms of verbs, or hidden by using impersonal expressions, thus making it difficult to correctly classify phrases in the regulatory fragment and find the bearer of a right or obligation. Surprisingly, the official English translation of the accessibility law in most cases explicitly states this information. Consider the use of verb phrases (in bold) to state the obligation in Italian and English versions of the same fragment, below:

Italian statement: *"Nelle procedure svolte dai soggetti di cui all'articolo 3, comma 1, per l'acquisto di beni e per la fornitura di servizi informatici, i requisiti di accessibilità stabiliti con il decreto di cui all'articolo 11* **costituiscono motivo di preferenza** *a parità di ogni altra condizione nella valutazione dell'offerta tecnica, tenuto conto della destinazione del bene o del servizio.*

English translation: *"The subjects mentioned in article 3, when carrying out procedures to buy goods and to deliver services,* **are obliged***, in the event that they are adjudicating bidders which all have submitted similar offers, to give preference to the bidder which offers the best compliance with the accessibility requirements provided for by the decree mentioned in article 11."*

Overall, the annotation results suggest that the Gaius T process for regulation analysis is applicable to documents that are written in different languages. The effort required to adapt the framework for the new application was relatively small with respect to the implementation. This experiment also revealed several language differences that we were able to quantify using Gaius T. In our future work we plan to conduct a more extensive analysis that may remove other language effects independently from legislator effects.

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. This review does not claim to be an exhaustive survey and we focus only on several works that are most related to our method with respect to the problem considered and our approach used.

The SACD system [16] relates well to our approach. The tool, implemented in Prolog, uses a combination of syntactic parsing and keyword-based rules, that

rely on the regularity of prescriptive documents, to generate a knowledge base from the logical structure of regulatory text. Once the processing is completed, SACD requires attention of the human specialist in revising the results provided. Similar to Gaius T, SACD recognizes several layers in prescriptive texts: the structural layer, called *macrostructure*; the logical layer, called *microstructure*; and the *domain* layer describing domain-specific information.

Cleland-Huang et al. [17] suggested an algorithm for detection and classification of 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 and then used to identify NFRs in requirements specifications. Along similar lines, the EA-Miner [18] tool supports separation of aspectual and non-aspectual concerns and their relationships by applying natural language processing techniques to requirements documents. The identification criteria in EA-Miner is based on a domain specific lexicon that was built observing related words. Similarly to these methods, we use normative phrases to identify the presence of regulatory requirements. However, our tool further recognizes the paragraph structure of regulatory text, which is necessary to acquire complete requirements from across continuations. The challenge of continuations cannot be addressed by indicator terms alone. Antón proposed the Goal-Based Requirements Acquisition Methodology (GBRAM) to manually extract goals from natural language documents, including financial and healthcare privacy policies [19]. Additional analysis of these extracted goals led to new semantics for modeling goals [20], which distinguish rights and obligations, and new heuristics for extracting these artifacts from text [2]. These heuristics have been combined into a frame-based method for manually acquiring legal requirements and priorities from regulations [3]. As discussed in this paper, our tool incorporates several of these heuristics to identify rights and obligations.

Wilson et al. [21] performed a detailed analysis of NASA requirements documents to identify recommendations for writing clearer specifications. As a part of this work, the authors discovered that good requirements specifications use imperative verbs (shall, must, etc.) to explicitly state requirements, constraints or capabilities. They also introduced the notion of *continuances*, i.e., additional phrases that refine upon previously stated requirements. We observed similar findings in language regularities in prescriptive documents that were incorporated into our set of heuristics to detect requirements. We also operate with the notion of continuances, which we call continuations, across sub-paragraphs.

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.

In [2], Breaux and Antón proposed a methodology for extracting stakeholder requirements from regulations. This paper presents a tool intended to provide

automatic support for analyzing policy documents. The new tool-supported process - named Gaius T - exploits the findings of our earlier work on requirements analysis, and exploits 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. Our envisioned process fits into a broader context, in which a requirements engineer or other analyst must integrate requirements from multiple regulations that affect a single product, service or system. We reserve this broader integration challenge for future work and our current focus remains on the immediate challenge of correctly identifying requirements from regulations.

To verify to what extent the semantic annotation tool can be applied to the domain of regulatory texts, we devised two empirical studies, involving annotation of a fragment of the U.S. HIPAA regulations and the Italian accessibility law, and compared the performance of the tool with manual identification of instances of rights, obligations, and associated constraints. The results of this study are encouraging, and have also revealed a number of useful extensions for the tool and the tool-supported process. The phrase heuristics used are now extended for documents in English and Italian. We believe that our tool supported process can be re-used in regulations developed for different areas of human activity due to its modularity.

We are interested in developing reasoning facilities on the annotations using constraints of the domain meta-model, for instance, cardinality constraints. Apart from the regulation compliance problem, another potential application of this work may be in providing support to lawmakers in writing regulations in terms of improved consistency and reduced ambiguity for use by engineers. We believe that semi-automated tools such as the one proposed in this paper can be effectively used to improve the overall quality of rules and regulations at many levels.

Acknowledgments

This work has been funded, in part, by the EU Commission through the SERENITY project, the Natural Sciences and Engineering Research Council of Canada, Provincia Autonoma di Trento through the STAMPS project and the U.S. National Science Foundation ITR #032-5269.

References

1. Berghel, H.: The two sides of 'ROI': Return-on-investment vs. risk-of-incarceration. Communications of ACM **48**(4) (2005) 15–20
2. Breaux, T.D., Vail, M.W., Antón, A.I.: Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In: Proc. of RE'06, Washington, DC, USA, IEEE Computer Society (2006) 46–55
3. Breaux, T.D., Antón, A.I.: Analyzing regulatory rules for privacy and security requirements. IEEE Transactions on Software Engineering **34**(1) (2008) 5–20

4. Breaux, T.D., Antón, A.I., Doyle, J.: Semantic parameterization: A process for modeling domain descriptions. ACM Transactions on Software Engineering Methodology **18**(2) (2009)
5. Breaux, T.D., Anton, A.I.: A systematic method for acquiring regulatory requirements: A frame-based approach. In: Proc. of RHAS-6, Pittsburgh, PA, USA, Software Engineering Institute (SEI) (September 2007)
6. Kiyavitskaya, N., Zeni, N., Mich, L., Cordy, J.R., Mylopoulos, J.: Text mining through semi automatic semantic annotation. In: Proc. of PAKM'06. Volume 4333 of LNCS., Springer-Verlag (2006) 143–154
7. U.S.A. Government: Standards for privacy of individually identifiable health information, 45 CFR part 160, Part 164 subpart E. In Federal Register **68**(34) (Feb. 20, 2003) 83348381
8. Italian Parliament: Stanca Act, Law no. 4, January 9, 2004: Provisions to support the access to information technologies for the disabled. Gazzetta Ufficiale **13** (17 January 2004)
9. Kiyavitskaya, N., Zeni, N., Breaux, T.D., Antón, A.I., Cordy, J.R., Mich, L., Mylopoulos, J.: Extracting rights and obligations from regulations: Toward a tool-supported process. In: Proc. of ASE'07. (2007) 429–432
10. Moulin, B., Rousseau, D.: Knowledge acquisition from prescriptive texts. In: Proc. 3rd Int. Conf. on Industrial and engineering applications of artificial intelligence and expert systems, New York, NY, USA, ACM Press (1990) 1112–1121
11. Cordy, J.R.: The TXL source transformation language. Science of Computer Programming **61**(3) (2006) 190–210
12. Dean, T.R., Cordy, J.R., Schneider, K.A., Malton, A.J.: Using design recovery techniques to transform legacy systems. In: Proc. of ICSM'01. (November 2001) 622–631
13. Zeni, N., Kiyavitskaya, N., Mich, L., Mylopoulos, J., Cordy, J.R.: A lightweight approach to semantic annotation of research papers. In: Proc. of NLDB'07, Springer Verlag (2007) 61–72
14. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proc. of Int. Conf. on New Methods in Language Processing, Manchester, UK (1994)
15. Presidenza del Consiglio dei Ministri: Guida alla redazione dei testi normativi. Gazzetta Ufficiale **101**(2) (2001) 105
16. Moulin, B., Rousseau, D.: Automated knowledge acquisition from regulatory texts. IEEE Expert **7**(5) (1992) 27–35
17. Cleland-Huang, J., Settimi, R., Zou, X., Solc, P.: The detection and classification of non-functional requirements with application to early aspects. In: Proc. of RE'06, Washington, DC, USA, IEEE Computer Society (2006) 36–45
18. Sampaio, A., Chitchyan, R., Rashid, A., Rayson, P.: EA-Miner: a tool for automating aspect-oriented requirements identification. In: Proc. of ASE'05, New York, NY, USA, ACM Press (2005) 352–355
19. Antón, A.I., Earp, J.B., He, Q., Stufflebeam, W., Bolchini, D., Jensen, C.: Financial privacy policies and the need for standardization. IEEE Security and Privacy **2**(2) (2004) 36–45
20. Breaux, T.D., Antón, A.I.: Analyzing goal semantics for rights, permissions, and obligations. In: Proc. of RE'05. (2005) 177–186
21. Wilson, W.M., Rosenberg, L.H., Hyatt, L.E.: Automated analysis of requirement specifications. In: Proc. of ICSE'97, New York, NY, USA, ACM Press (May 1997) 161–171