Pre-processing Legal Text:
Policy Parsing and Isomorphic Intermediate Representation

K. Krasnow Waterman
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139, USA
kkw@mit.edu

Abstract

One of the most significant challenges in achieving digital privacy is incorporating privacy policy directly in computer systems. While rule systems have long existed, translating privacy laws, regulations, policies, and contracts into processor amenable forms is slow and difficult because the legal text is scattered, run-on, and unstructured, antithetical to the lean and logical forms of computer science. We are using and developing intermediate isomorphic forms as a Rosetta Stone-like tool to accelerate the translation process and in hopes of providing support to future domain-specific Natural Language Processing technology. This report describes our experience, thoughts about how to improve the form, and discoveries about the form and logic of the legal text that will affect the successful development of a rules tool to implement real-world complex privacy policies.

Pre-processing for Accountable Systems

We have been working on the development of policy-aware accountable systems – systems that can compute compliance with data usage policies such as privacy and data security.[1] In the course of that work, it became clear that one challenge was the ability to effectively communicate the details of law, regulation, policy, and contract from policy-makers to computer scientists. Earlier projects limited that effort to very small collections of sub-rules in order to focus on other functionality challenges.

Our current project is to model data transactions that occur in the context of Fusion Centers – entities where federal, state, and local law enforcement, intelligence, and emergency response organizations work together and share information to improve the success of their missions. Specifically, we are modeling transactions in which personal information is being transmitted and showing how a system that includes representation of complex privacy policy and the technology to reason – to make judgments about the applicability of that policy to data events – can improve compliance with such policies. This meets a critical need to address both the trust and constitutional integrity questions raised by flow of information between law enforcement and intelligence, domestic and extra-territorial activities.

In order to create even a simple model of this environment, we needed to represent significantly larger rule sets than we had done before. Based upon prior experience, we expected this to be a difficult hurdle to cross, in order to reach the intended challenges of our project. Other research has suggested, and we independently reached the same conclusion, that intermediate isomorphic representation would assist the transfer of knowledge from legal-ese to program code.[2] Like the Rosetta Stone, this would provide the intermediary necessary for each side to have a form that they more readily understand and can use to ensure the other's understanding. Isomorphism is important in this context because legal rules are so complex to the policy-maker and so convoluted to the computer scientist that combining the rules or summarizing them will make it nearly impossible to validate the accuracy of representation or to later efficiently reflect incremental changes to the underlying legal documents.

For this early phase of our current project, we have worked with two intermediate representations: the first, a representation of the Privacy Act recently created for the federal government [3] and the second, a representation of several Massachusetts laws we created to address some of the challenges identified in the first. This paper briefly describes (i) requirements for accountable systems observed while parsing and using parsed policy and (ii) benefits and deficits of the particular intermediate forms.

Development of Intermediate Isomorphic Forms

The challenge of relaying the meaning of a specific law to computer scientists has always been present but not the central focus of our work. We had made several attempts at creating an intermediate form, including converting to triples , a structured outline with unstructured text, and a highly structured table with a high level of abstraction. These each fell short because they lacked sufficient detail or were too abstract to accelerate the coding process. In later 2008 and early 2009, a project outside our group (but including one of our principals) extended this work to a new form and parsed twenty laws, regulations, and policy memos. It attempted to produce each sub-rule more completely, for example, converting

> "No agency shall disclose any record which is contained in a system of records by any means of communication to any person, or to another agency…" [4]

to the spreadsheet form in Figure 1 (next page). That form was built with a focus towards making the information readable to a policy user. It can be viewed on a monitor in two screen-widths, with the substance of the rule (the first two graphics concatenated) – the more commonly sought information – appearing to the left and the administrative details about the rule (the concatenation of the third and fourth graphic) appearing after scrolling to the right.

After attempting to create expression of the policy in code from that form, we produced another intermediate form, which converts

> "Except as otherwise provided in this section and sections one hundred and seventy-three to one hundred and seventy-five, inclusive, criminal offender record information, and where present, evaluative information, shall be disseminated, whether directly or through any intermediary, only to (a) criminal justice agencies…" [5]

to Figure 2, which is intended to be more coder-friendly, parsing the rule into subclasses that more likely align with the probable locations of the data. For example, rather than treating all of the information about an individual as about the individual, organizational attributes are segregated into their own subclass. So, too information which is likely within a data file is now separated from information about the use of the file which is more likely be contained in header data or a log file.

Discoveries Regarding the Intermediate Isomorphic Forms

Each of the intermediate forms with which we worked attempted to abstract all data handling policy into one consistent structure. This was done by breaking the text of a rule into discrete sub-rules; breaking the text of each sub-rule into sub-classes of actors, data, actions, permissions, etc.; and, then, associating each sub-rule with sub-classes about the rule's provenance and its logical relationships with other rules. Producing these intermediate forms is complex and tedious, but sufficiently achievable to support our research effort. (It is our goal to use such manual efforts to accelerate research to produce a more successful natural language policy parser.) We attempted to use the intermediate form as the basis for our coding effort and determined that they were useful but not mature.

Validation Requirement

Using either of the intermediate forms (in Figures 1 & 2), the computer scientists still found that they wished to look at the original text. They were both definite that the intermediate form was useful and that it was insufficient as a sole source. Interestingly, it also was observed that the intermediate form was not helpful in understanding how already-written code was derived from original text; investigation is warranted to determine the reason for this.

These experiences provide support for the developing notion that some sort of validation tool is necessary and that it may be most useful if it visually aligns the three forms of representation.- (i) original text, (ii) isomorphic representation, and (iii) program code. We expect that users and their lawyers will need to trust that a program correctly expresses policy before they will adopt an accountable system. While they will be unlikely to read program code, they will wish to see that the translation to intermediate form is correct. Coders will want to validate from intermediate form to program code, while still using the original text for confirmation of understanding.

Link Relationship Challenge

Expressing the linkages between sub-rules is critical to properly informing the reasoning engine and achieving the correct results. In the portion of the relatively brief (less than two pages of text) Massachusetts law that we represented in intermediate form, nineteen of twenty four sub-rules contained an expression of an association with one or more other sub-rules; in total there were thirty-five links expressed in the law that were recursive. In the case of this particular Massachusetts law, the internal link relationships were predominantly expressions of conditions and exceptions, but several of the sub-rules had to be linked to five critical and detailed definitions contained in

Party Subject to the Rule			RULE Verb	Transaction Verb	Data Subject to the Rule		
Party subject to the Rule	Attribute of Party subject to rule	Person Context	Rule Type	Activity Type	Data Category	Special Data Category	Data Context
Government: Federal: Agency			Prohibited	Share	Person	PII: US Person	Source: System Of Records

People Involved in the Data Transaction						Circumstances in which the rule applies
Releasing Entity	Attribute of Party subject to rule	Context	Accessing Entity	Attribute of Party subject to rule	Context	Authorized Purpose detail
Government: Federal: Agency			Person; Government: Federal: Agency			

Administrative information that makes it possible to understand precedence, provenance, and linkages					
Rule precedence type	"Facilities" rule	Document	Document Citation	Document sub-Reference	Dated
1.2a: Federal Statute		The Privacy Act	The Privacy Act, 5 USC § 552a	(b)	

Administrative information that makes it possible to understand precedence, provenance, and linkages						
Record Number	Old Record Numbers	Linked to Rule Name	Linked to Records	Link Reason	Changed Perspective Flag	Perspective
180001		1) Definitions: Agency, System of Records, Record		1) AND,		

Figure 1. Intermediate Isomorphic Representation of Privacy Act Sub-rule (Iteration #4 – government produced)

Actor				Rule	Action	
Organization Category	Person Details	Organization Details	Context	Rule Verb	Action Verb	Action Context
MA, Executive Branch	Governor, Lieutenant Governor and Council, Certain officers under …		Rule applies to; Sender	required	disseminate	Directly or through intermediary

Data				Other Party				Environment
Data Category	In the Data	Trans Header/ Historical/Provenance	Context	Entity	Person Details	Organization Details	Context	
person	criminal offender record information or criminal offender evaluative information			criminal justice agencies			Recipient	

Rule Details					
Rule Name	Citation	Effective Date	Termination Date	Hierarchy Number	Hierarchy Name
Full: Dissemination of record information; certification; … Short: Dissemination of record information	Mass. Gen. Laws. Ann., Ch.6 § 172			1.2a	

Sub-rule Details						
Subrule Num (Hidden)	Citation	Effective Date	Termination Date	Relationship to other sub-rules	Other Subrule Name	Other Sub-rule Hidden Number
dig00001	Para. 1, Sent. 1, (a)			1) May be excepted by 2) Must be conditioned by 3) Must be conditioned by 4) Must be conditioned by	1) MGLA., Ch.6 § 173 - 175 2) MGLA., Ch.6 § 172, Para. 1, Sent. 2, Cl. 1 3) MGLA., Ch.6 § 172, Para. 2 4) MGLA., Ch.6 § 172, Para. 5, Sent. 1, Cl. 2 OR MGLA, Ch.6 § 172, Para. 5, Sent. 2, Cl. 2	1) [insert when assigned] 2) dig00004 3) dig00008 4) dig00011 OR dig00013

Figure 2 – Intermediate Isomorphic Representation of Massachusetts Criminal Offender Records Law (Iteration #5)

four other laws.

We believe that accountable systems must mirror those relationships as failing to do so makes both validation and updates extremely difficult. In order to properly express the logic intended by the policy-maker, each sub-rule must contain an expression of its relationships with any other sub-rule. This includes describing exceptions, conditions, joins, disjoints, definitions, and sources of needed values; it also includes expressions of whether the associations themselves are fixed or conditional. However, even working from the intermediate form, links were not transferred to program code, but not consciously observed. A visualization tool may be helpful. It is relatively easy to imagine a graphical interface (a bubble chart or spider graph with slider bar) that makes it possible to see the relationships between sub-rules within and outside a rule.

Condition Subsequent Requirement

Temporal challenges abound in data usage policy. The policies themselves have both effective dates and expiration dates that will need to be closely observed, particularly when the accountability function is historical audit. Many rules contemplate events which occurred *before* the current event (e.g., the original collection of the data) or require knowledge of time elapsed since an event (e.g., a training class attended by the actor or the birth of the subject of the data). In our intermediate representation – and our coding, to be discussed in a later report – a significant challenge arose regarding how to represent requirements for what had to occur *after* the data usage grant, what lawyers call "conditions subsequent."

For example, consider a rule that says a party may share information, so long as the recipient destroys it within 180 days. While the destruction clause may be described as additional *context* information for the action, it probably should not be placed in the same location in the intermediate form. Because it describes a requirement which cannot be met at the time the reasoner will determine compliance and because it is too far beyond the bounds of what a user could be expected to offer as a condition in seeking a compliance decision, it appears to require its own place in the abstracted form, a place from which it can be drawn to provide a conditional compliance response. This simple modification of the intermediate form will highlight how often the problem occurs and what level of priority should be assigned to the problem.

Additional System Design Requirements Discovered While Parsing Privacy Law

Dynamic Identification of Agency Policy

While reviewing the parsed version of the federal Privacy Act, it was discovered that more than a third (48 of 134) of its sub-rules require each federal agency to issue its own rule on a particular topic or to cause a particular effect. This means that for an accountable system to run completely, it would need to be able to find and call each of those required agency rules as part of the process.

In the Privacy Act, which applies to all US federal agencies, the most common example is that an agency can disclose (share or grant access to) information if that disclosure is compliant with the agency's published rules for the specific data system, known as Routine Uses:

> "(b) **Conditions of Disclosure.—** No agency shall disclose any record which is contained in a system of records . . . …unless disclosure of the record would be— . . .
> "(3) for a routine use as …described under subsection (e)(4)(D) of this section". . .
>
> "(e) **Agency Requirements.—** Each agency that maintains a system of records shall—
> (4) . . .publish in the Federal Register upon establishment or revision a notice of the existence and character of the system of records, which notice shall include— . . .
> (D) each routine use of the records contained in the system, including the categories of users and the purpose of such use". [6]

The practical impact is that a system determining whether a data transaction is compliant with the Privacy Act will need to make that determination under all the requirements explicitly stated in the Act; in addition, it will need to reason over the additional usage rules *for the particular system*. In order to do so, while processing it will have to identify the existence and location of those usage rules from a previously published Routine Use notice.

A different example is the Massachusetts law which sets the policy for releasing criminal offender records.[7] That law requires a Criminal History Systems Board to create a list of approved criminal justice agency recipients and the scope of permissible release. A fully effective policy aware system would need to make all of the release decisions under the state statute and also find the board-created policy about the specific agency and apply it.

The author is unaware of any existing technical method to cross-reference agency policy to these requirements and, as a practical matter, unaware of any current manual practices that do so. An accountable system should have the ability to express that such policies are expected, to search for them in an organization's rules library, and to either run them at the appropriate point in the rule sequence or to report that they are missing. We believe this might result in a higher degree of compliance than is currently achieved, both because it will consider *all* the required rules before reaching a conclusion and because it will identify rules which are required but missing.

Granular Representation of Definitions

Lawyers are trained to understand that a law means what it says on its face, only to the extent that it is not modified or clarified by other information: another statute, case law, etc. Often, a word in a statute is not used for its common definition, but a special definition used just in that rule or a group of related rules. For example, in the Privacy Act, the definition of "maintain" also includes "collect" and disseminate", which fall outside of our traditional "keep in an existing state" definition.[8] For an accountable system to correctly determine compliance, it must use languages capable of representing this level of specificity in both the policies and about the data.

Also, the differences between definitions of terms will be the deciding factor in some situations. For example, consider two definitions of "criminal justice agency." Massachusetts [9] says:

> "an agency at any level of government which performs as its principal function activity relating to (a) the apprehension, prosecution, defense adjudication, incarceration, or rehabilitation of criminal offenders, or (b) the collection, storage, dissemination, or usage of criminal offender record information."

Compare this with Maryland [10], which defines "criminal justice agency" as:

> "(i) courts: and (ii) a governmental agency or any subunit therefore that: 1. performs the administration of criminal justice pursuant to a statute or executive order, and 2. allocates a substantial part of its annual budget to the administration of criminal justice, and (2) includes federal and state inspectors general offices."

Massachusetts would find an inspector general – typically an executive branch civil oversight role - not a "criminal justice agency" because it generally does not deal with criminal offenders, but Maryland would find it to be a "criminal justice agency" because the Maryland definition has explicitly this type of organization. If a Massachusetts law permits sharing with a "criminal justice agency", that means sharing with an agency that meets the Massachusetts definition of the term. An effective accountable system must look past the fact that a Maryland entity bears the same "criminal justice agency" label and use the granular components of the definitions to make the correct compliance decision.

Implied Meaning (NLP Limitation)

Our basic philosophy is to represent the rules as written. Modeling the legal world, each rule is parsed as it is written, with the expectation that a rules library would also contain the interpretive overlay rules created by case law, in-house advice, or other policy, and that the reasoner would be able to process them with the correct conflict resolution logic. This is intended to make it possible to link each sub-rule exactly to its source, thus making it possible to quickly identify where and how to change any representation when its legal world original is changed (e.g., a new case decision, a change in opinion or policy). During the policy parsing process, it became clear that there may need to be an exception to this construct where the choice of language in the original rule obfuscates the ability to correctly implement it. For example, if a statute were to say "anyone may access [data x]" it does not normally mean that the legislature intended for the public to hack into the system and pull the data at will. It means that a responsible government entity will make the data accessible. In the cases, such as this example, in which straight parsing would achieve the inherently wrong result, we will parse a restated version to include the proper entity and specifically mark that we have done so. Such an approach should make it possible to study how often the problem occurs and how any natural language parser would need to be enhanced to address it.

Intermingled Objective and Subjective Values

Parsing legal text into discrete logical elements also made apparent another issue which will need to be addressed by anyone attempting to do automated reasoning about data usage. The rules regularly require knowledge of both objective and subjective information about the actors, data, and environment.

Looking at the rules through this prism, it would be apparent to someone with knowledge of their existing data repositories, which of the objective information is currently available in metadata or other form expose-able to a reasoner and, thus, what metadata creation requirements should be included in the next system update. Values are often readily available for variables like “organization name,” “organization type,” and “job role” or “data category”. Not often readily available, but achievable would be to carry more data provenance, more detailed information about the date, method, and subject volition associated with the original collection as well as the repositories and owners it has passed through.

Significantly more difficult, but still objective, is the need to expose the name of an individual contained in a file and match it to the name of the requestor of information or the requirement to determine if the requestor has been granted access by a court decision of a particular type.

However, there is a broad array of requirements which are unlikely to be objectively inferred, such as whether data is necessary for actual performance of actions or duties sustaining the public interest; whether data is directly relevant to the decision to release a prisoner; whether a description is reasonable, whether a law enforcement officer has probable cause or a reasonable suspicion, etc. While parsing policy with this in mind, it is possible to identify those pieces of information which should be left to user assertion for the foreseeable future. While such subjective information is not infer-able by a system, the mere act of collecting it will provide more complete transparency about how and why data usage occurs.

Conclusion

The first phase of this project has been very fruitful. Though just beginning the exploration of the robustness and extensibility of the policy language and reasoner to a broader and more complex set of rules, we already have learned quite a lot about the challenges of getting law-based rules to the policy programmer. While parsing policy, we identified five challenges. Four are requirements for accountable systems, including requirements for granular representation of definitions, expression of link relationships between sub-rules, clear distinction between objective and subjective input, and the ability to dynamically identify and reason over a rule as part of the processing of a different rule. Additionally, we expanded our knowledge in a way that may improve natural language policy parsing. While attempting to use an intermediate representation form, itself a by-product of earlier research challenges, we identified three new technologies which should be explored as ways to improve and expedite the translation process. These are: a visualization tool showing all three versions of the rule; a graphical interface displaying sub-rule link relationships; and a mechanism to handle condition subsequent requirements.

Acknowledgments

Thanks to my MIT DIG and TAMI colleagues Hal Abelson, Tim Berners-Lee, Li Ding, Jim Hendler, Lalana Kagal, Deborah McGuinness, Joe Pato, Oshani Seneviratne, Fuming Shih, Gerry Sussman, and Samuel Wang for raising the questions and challenges which led to this paper. Thanks to Martin Smith, Department of Homeland Security, for funding and Patricia Kennedy Hammar, for co-leading, the team that produced the form in Figure 1. Special thanks to Michael J. Rosensweig, whose long days of coding and occasional questions led to many of these insights. This work was made possible by the confluence of multiple projects. The idea was germinated in the prior work sponsored by NSF Cybertrust Award #0524481 and IARPA Award #FA8750-07-2-0031, and proceeds under the IARPA (DHS Fusion Center Award).

References

[1] Weitzner, Abelson, Berners-Lee, Hanson, Hendler, Kagal, McGuinness, Sussman, Waterman, Transparent Accountable Data Mining: New Strategies for Privacy Protection,; MIT CSAIL Technical Report MIT-CSAIL-TR-2006-007 [DSpace handle] (27 January 2006) and Kagal, Hanson, & Weitzner, Integrated Policy Explanations via Dependency Tracking, IEEE Policy 2008.
[2] Bench-Capon, TJM & Coenen, FP, *Isomorphism and Legal Knowledge Based Systems*, Artificial Intelligence and Law, Vol 1, No 1, pp65-86 (1992).
[3] Waterman, K, Hammar, P, et al., parsing of 20 laws, regulations, and policies for the US Department of Homeland Security, Mar 2009
[4] 5 USC § 552a(b)(7).
[5] Mass. Gen. Laws. Ann., Ch.6 § 172
[6] 5 USC § 552a(b)(3) & (e)(4)(D).
[7] Mass. Gen. Laws. Ann., Ch.6 § 172
[8] Merriam-Webster Online, http://www.merriam-webster.com/dictionary/maintain (2009)
[9] Mass. Gen. Laws. Ann., Ch.66A § 1
[10] Md. Regs. Code 12.15.01.03