Deriving Semantic Models from Privacy Policies

Travis D. Breaux and Annie I. Antón
Department of Computer Science
North Carolina State University
{tdbreaux, aianton}@eos.ncsu.edu

Abstract

Natural language policies describe interactions between and across organizations, third-parties and individuals. However, current policy languages are limited in their ability to collectively describe interactions across these parties. Goals from requirements engineering are useful for distilling natural language policy statements into structured descriptions of these interactions; however, they are limited in that they are not easy to compare with one another despite sharing common semantic features. In this paper, we propose a process called semantic parameterization that in conjunction with goal analysis supports the derivation of semantic models from privacy policy documents. We present example semantic models that enable comparing policy statements and discuss corresponding limitations identified in existing policy languages. The semantic models are described by a context-free grammar (CFG) that has been validated within the context of the most frequently expressed goals in over 100 website privacy policy documents. The CFG is supported by a qualitative and quantitative policy analysis tool.

1. Introduction

Consumers are increasingly concerned about their privacy while engaging in online transactions. At the same time, American companies in regulated industries must ensure that their online privacy policies are enforceable and compliant with privacy laws such as the Gramm-Leach-Bliley Act[1], and the Health Insurance Portability and Accountability Act[2]. Privacy policy documents are long, monolithic and difficult for the average Internet user to comprehend [1]. One approach for extracting the salient information from these privacy policy documents is to adopt a goal-driven approach in which text is parsed to extract structured natural language statements expressed as goals [1][2][3].

Policy languages offer the ability to formally express policies making it possible to automate enforcement of policy-governed interactions in an advanced policy management system. One approach to understanding the strengths and weaknesses of existing policy languages is to formally analyze the semantics of existing policy documents. In this paper we present (a) a generalizable process for developing semantic models from privacy policy goals mined from policy documents and (b) a tool that enables quantitative and qualitative model analysis including the ability to compare policy statements.

Natural language privacy policy statements can be systematically analyzed using a content analysis technique, goal-mining. *Goal-mining* refers to the extraction of goals from data sources by the application of goal-based requirements analysis methods [2]. The extracted goals are expressed in structured natural language [1]. Goals are organized according to goal class (privacy protection or vulnerability) as well as according to keyword and subject (e.g. browsing patterns, personalization, cookies, etc.). These goals are documented in a Web-based Privacy Goal Management Tool (PGMT) [1] developed at North Carolina State University. To date, the tool contains over 1,200 goal statements extracted from nearly 100 Internet privacy policy documents. The following are example goals as expressed in the PGMT:

> G_{642}: *SHARE customer information with subsidiaries to recommend services to customer*
>
> G_{867}: *USE customer email address for marketing and promotional purposes*
>
> G_{1166}: *SHARE customer information with third-parties to perform marketing services on our behalf*

Researchers have acknowledged the need for methods to analyze and refine policy specifications [4]. Bandara et al. have noted the need to derive enforceable policies from high-level goals. Their approach relies on event calculus and abductive reasoning to derive the

[1] Gramm-Leach-Bliley Act of 1999, 15 U.S.C. §§ 6801 - 6809 (2000).

[2] Health Insurance Portability and Accountability Act of 1996, 42 U.S.C.A. 1320d to d-8 (West Supp. 1998).

Proceedings of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)
0-7695-2265-3/05 $20.00 © 2005 **IEEE**

operations that together satisfy system policy goals. Our approach seeks to develop rich models that enable specification of specific rights, permissions and obligations.

Our work to date has enabled us to develop a preliminary framework for specifying and analyzing privacy policies [5], but given the informal nature of structured natural language goal statements, we seek ways to represent the rights, obligations and relationships relevant to privacy policies so that they may be compared and systematically analyzed. Ultimately, this will enable companies and government agencies to automatically monitor and audit policy enforcement.

The remainder of this paper is organized as follows. Section 2 introduces the relevant background, terminology and the semantic parameterization process. Section 3 discusses example semantic models developed using the proposed process. Section 4 discusses the results of our use of a tool to automatically compare parameterized privacy goal statements. Section 5 explains how our proposed approach is different from the support and features available in existing policy languages. Finally, Section 6 summarizes our findings and plans for future work.

2. From policy goals to semantic models

Advanced policy management and analysis, including policy authorship, organization and query relies on the fundamental ability to compare policy statements. Because natural language statements in general are intractable for our purposes, we chose to develop the capability to compare policies using policy goals that offer more concise and consistent representations of information. In this approach (see Figure 1), goals that were previously (a) mined from privacy policy documents [1] and stored in the PGMT repository are now (b) re-stated to form restricted natural language statements (RNLS) that are then (c) parameterized to derive semantic models.

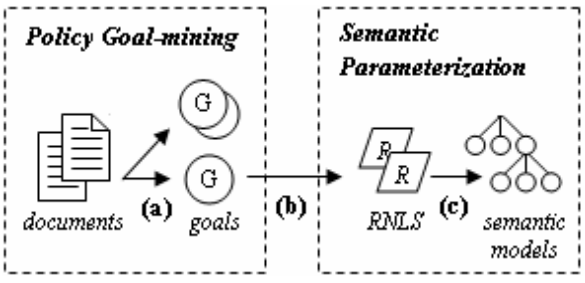

Figure 1: Semantic Parameterization Process.

Although goals are difficult to compare, they are well suited for the semantic parameterization process because their structure often satisfies the requirements for RNLS. Furthermore, the parameterization process yields semantic models that are easily comparable using automated techniques.

We begin with an introduction to policy goals by presenting the important features that make goals amenable to parameterization through RNLS re-statement. Afterwards, we present the semantic parameterization process including the formal definition of semantic models. Finally, we discuss the context-free grammar derived from the modeling notation used to perform automated analysis.

2.1. Restating goals into RNLS

The PGMT goal repository contains over 1,200 privacy goals and vulnerabilities extracted from 100 website privacy policy documents. The goals are expressed in structured natural language that describes an event and allows nested activities that identify the *actors*, *actions* and *objects*. For example, an actor in the repository is always either a customer or provider of products or services. In addition, each goal begins with a specific keyword, a verb that describes the primary action performed by the actor. Consider PGMT goal G_{161}:

> *G_{161}: COLLECT information from non-affiliates.*

From this goal, we identify the action "collect" (a verb) and the object "information" (a noun). The actor is identified as the "provider" in the PGMT. Depending on the action, other parts of speech will consistently follow the action, object pair. For example, in this goal a noun follows the preposition "from". The verb "collect" suggests the pairing of this preposition with a noun however it is not required by the verb; that is "from non-affiliates" could have been omitted in the goal statement but this would have generalized the statement's meaning. Additional information conditioned by the type of action is common in goals.

RNLS like goals have exactly one primary actor, action and at least one object. Unlike goals that may describe nested activities, each RNLS only describes one activity with external references to other activities. Consider goal G_{707}:

> *G_{707}: RECOMMEND customer select access codes that are easy for customer to remember but hard for others to guess (e.g. combination of numbers/letters)*

Re-stating G_{707} as RNLS(s) requires decomposing the goal into discrete but related activities. The activities described by an RNLS each have one actor and action, and must exhaustively describe the essential information in the original goal. In the decomposition, we use the modal "may" to distinguish rights, "will" to distinguish obligations and "can" to distinguish general

Proceedings of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)
0-7695-2265-3/05 $20.00 © 2005 **IEEE**

abilities of actors. The following RNLSs correspond to goal G_{707}:

> ***RNLS #1:*** *The customer will select access codes.*
>
> ***RNLS #2:*** *The provider will recommend (RNLS #1) to the customer.*
>
> ***RNLS #3:*** *The customer can easily remember the access codes.*
>
> ***RNLS #4:*** *Others can hardly guess the access codes.*

The above decomposition demonstrates the re-statement of goals into RNLS(s) with respect to two common cases: transitive verbs and objects described by other activities. In RNLS #2, note the parenthetical reference to the activity described in RNLS #1; this reference is characteristic of transitive verbs like "recommend" that describe another activity. RNLS #3 and #4 both describe the "access codes" in RNLS #1 using two activities; these activities are "easy to remember" for the customer and "hard to guess" for others. Semantic models maintain these important relationships.

2.2. Building semantic models

Semantic parameterization allows analysts to express restricted natural language statements (RNLS) as comparable semantic models. We begin by providing an overview of the modeling terminology using an example RNLS before introducing the formal modeling notation. The example concludes with a complete semantic model.

Semantic models are built from formal *components* that each describes exactly one activity through a set of unique *parameters*. The parameters are second-order semantic relationships that assume values from natural language or other components. As depicted in Figure 2, every component includes at least the following parameters: an *actor*, *action*, and *object*. The actor in an activity has a general capability to perform the action with respect to the object.

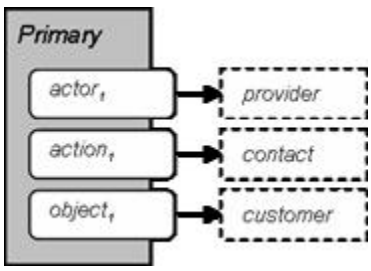

Figure 2: The minimal model component.

Every semantic model has only one *primary component* that describes the primary activity. However, semantic models may have *auxiliary components* assigned as values to a parameter for building relationships between activities (see Figure 3). The significance of these distinctions will become clearer as we proceed through the following example of a model instance.

Semantic models are instantiated by assigning words from a RNLS with a single part-of-speech to the values of specific component parameters. Consider RNLS #5:

> ***RNLS #5:*** *The provider may collect information.*

In RNLS #5 the values for the primary actor, action and object are "provider," "collect" and "information," respectively. In order to maintain an atomic, meaning-preserving correspondence, parameter values never combine two or more parts-of-speech. For example, these assignments correspond to a noun, verb and noun in the RNLS, respectively. Some parts-of-speech are subsumed by specific parameters. Modals such as "will" or "may" are subsumed by a parameter discussed later in Section 4.2. The parameterization process systematically accounts for other parts-of-speech including adjectives, articles, determiners, possessive qualifiers, and conjunctions among others.

Semantic models are formally defined using a modeling notation with only two asymmetric relations. Model parameters are defined using the associative relation α over a component and a parameter. Values are assigned to a parameter using the instance relation δ over a parameter and a value. The solid directed arrows in Figure 2 represent instance relations from the parameters to the values. Components, parameters, and values are represented in the notation using unique predicates with subscripts to distinguish parameters between different components.

Continuing with RNLS #5, we derive the following associative relations $\alpha(activity_1, actor_1)$, $\alpha(activity_1, action_1)$, and $\alpha(activity_1, object_1)$, as well as the instance relations $\delta(actor_1, provider)$, $\delta(action_1, collect)$, and $\delta(object_1, information)$. Note, in this example the $activity_1$ predicate is a handle for the component. Using only the associative and instance relations, the parameterization process is complete if and only if every word in an RNLS is assigned to or subsumed by one parameter. Completeness of the process guarantees that each semantic model maintains a natural language correspondence that enables reconstructing natural language statements from the instantiated model.

In addition to words from an RNLS, a parameter value may also be another component. In this case, the auxiliary component is an extension to the semantic

model and describes a separate but related activity through the nested component. We extend RNLS #5 by including the purpose "to market services" stated in RNLS #6:

> ***RNLS #6:*** *The provider will collect information to market services.*

We complete the second parameterization by defining an auxiliary component using the formal parameter $\alpha(activity_1, purpose_1)$ and the assigned value $\delta(purpose_1, activity_2)$. In the case of the purpose parameter, the preposition "to" will always be subsumed by this parameter; and in general, prepositions are normally subsumed by parameters. The new component for "to market services", includes the associative relations $\alpha(activity_2, action_2)$, $\alpha(activity_2, object_2)$ and instance relations $\delta(action_2, market)$, and $\delta(object_2, services)$. Unless the purpose explicitly states a different actor, the actor for the auxiliary component is assumed to be the same actor as the primary component. Therefore, the relations $\alpha(activity_2, actor_2)$ and $\delta(actor_2, provider)$ are also implied by the purpose parameter. Figure 3 shows the purpose parameter with an auxiliary component as the assigned value.

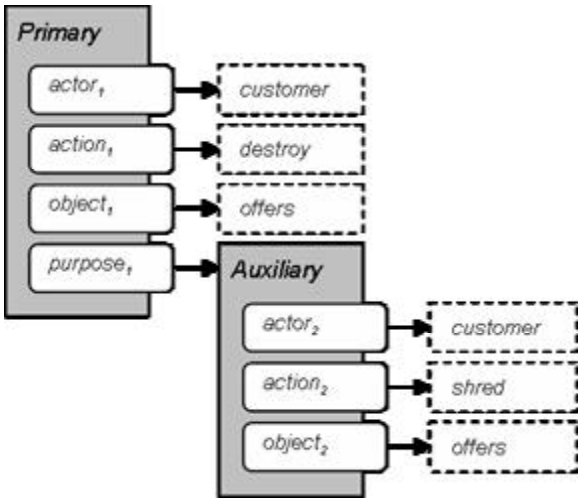

Figure 3: Semantic model with a purpose.

Using these models, we are able to compare RNLSs by holding select parameter values constant and querying the values of remaining parameters across a set of instantiated models. Consider RNLS #6 and RNLS #7, below:

> ***RNLS #7:*** *The provider may contact the customer to market services.*

We can build a query to ask the question, "What activity can the provider perform to market services?" The query will constrain the parameters $\alpha(activity_1, actor_1)$, $\alpha(activity_2, action_2)$, and $\alpha(activity_2, object_2)$ using the values $\delta(actor_1, provider)$, $\delta(action_2, market)$, and $\delta(object_2, services)$, respectively. The query parameters $\alpha(activity_1, action_1)$, and $\alpha(activity_1, object_1)$ will then acquire the values ⟨*collect*, *information*⟩ and ⟨*contact*, *customer*⟩ from both parameterizations, respectively. These result sets are indeed the answers to our query.

3. Example semantic models

For this investigation, the semantic parameterization process was applied to the 100 most frequent goals in the PGMT. These goals were restated to form proper RNLS(s). In order to handle these re-statements, two passes were made through the 100 goals in the goal subset. In the first pass, the semantic models were derived from the goals only when an obvious combination of parameters in the model notation was identified for a complete parameterization. In the second pass, the goals that were not previously parameterized were re-stated using observations from the first pass to produce a complete parameterization. In general, identifying the atomic activities and making explicit the implied actors and objects is all that is required to restate goals into proper RNLS and build a complete semantic model. The two-pass procedure made it possible to consistently parameterize the entire goal set. The entire two-pass procedure required less than eight person-hours and included the process of developing this methodology.

Applying the semantic parameterization process to the policy goal subset produced valuable insight into the semantic relationships within privacy policies. We summarize our most interesting observations in four semantic models. These models may be generalized as two distinct cases. The first case occurs when a parameter of a primary component is assigned a value of an auxiliary component. Recall, this type of assignment was first introduced in Section 2.2 and shown in Figure 3. In the second case, we show how an auxiliary component can further distinguish a value in a primary component. In these three models, the only formal relations mentioned are those that characterize the point of emphasis. Other relations included in the complete parameterization are not discussed for the sake of brevity.

3.1. Instruments as actions

The instrument of an activity is an additional means by which that activity is achieved. In a semantic model, the instrument is a parameter that may take on the value of words from a RNLS or another component. In the former situation, the words are always nouns describing an entity that maintains an implied, characteristic ability to perform some facilitating activity. For example, an instrument may be a "telephone" with the ability "to

communicate with other people." Here, the ability "to communicate" is both implied and characteristic of the telephone. We consider the latter situation here, where a component explicitly describes the facilitating activity. Consider goal G_{267}, an obligation where the main actor is the customer:

`G267: DESTROY offers by shredding them.`

Using the parameterization process, we decompose goal G_{267} into RNLS #8 and #9. Introducing the new parameter $\alpha(activity_1, instrument)$, we assign it the value of the component derived from the second restricted statement $\delta(instrument, activity_2)$ to produce the model in Figure 4.

RNLS #8: *The customer will destroy offers.*

RNLS #9: *The customer will shred offers.*

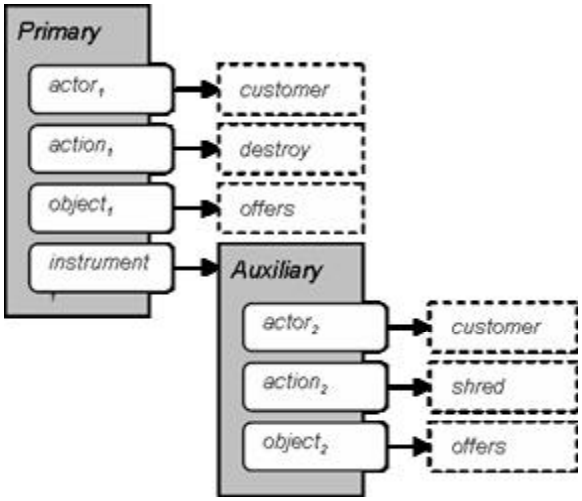

Figure 4: Semantic model with an instrument.

The above semantic model was instantiated by three different goals in the entire subset. In addition to the above example, the other two instances included the statements "by memorizing personal identification numbers" and "by browsing the institution's website."

3.2. Objects as actions

Transitive verbs that describe how an actor may affect another activity can be captured by a unique semantic model. In some situations, these models may describe how actors delegate permissions and obligations to other parties. In other situations, these models may describe notifications and warnings that actors provide to other parties. In the semantic model of Figure 5, we examine the situation where an actor restricts the activities of another party. Consider G_{500}, an obligation where the main actor is the provider:

`G500: RESTRICT non-affiliate sharing of customer information.`

We use the parameterization process to decompose G_{500} into RNLS #10 and #11. Recognizing the transitive verb "restrict" in RNLS #11 we derive the parameters $\alpha(activity_1, object_1)$ and assign it the value $\delta(object_1, activity_2)$ derived from RNLS #10.

RNLS #10: *The non-affiliate may share customer information.*

RNLS #11: *The provider will restrict (RNLS#10).*

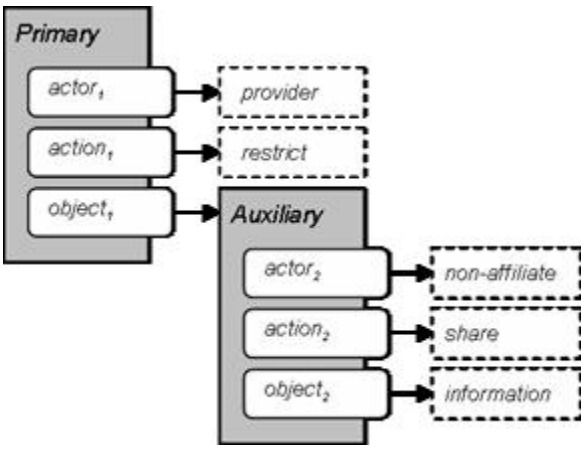

Figure 5: Object value is a component.

In general, the model in Figure 5 covers situations where a transitive verb directly affects another activity. In addition to "restrict", other transitive verbs identified in the goal subset during this process include "allow," "deny," "notify," "limit," and "recommend."

3.3. Objects as actors of other activities.

Activities may refer to objects that are actors in other activities. The purpose of including references to other activities serves to limit the scope of the primary activity to those objects that *have* or *have not* performed some other activities. Consider goal G_{581}, an obligation where the actor is the provider.

`G581: COMPLY with federal laws governing information`

We use the parameterization process to decompose G_{581} into RNLS #12 and #13. The relations $\alpha(activity_1, object_1)$ and $\delta(object_1, laws)$ from RNLS #12 are aligned with the relations $\alpha(activity_2, object_2)$ and $\delta(object_2, laws)$ from RNLS #13. The resulting model is illustrated in Figure 6.

RNLS #12: *The provider will comply with laws.*

RNLS #13: *The federal laws can govern information.*

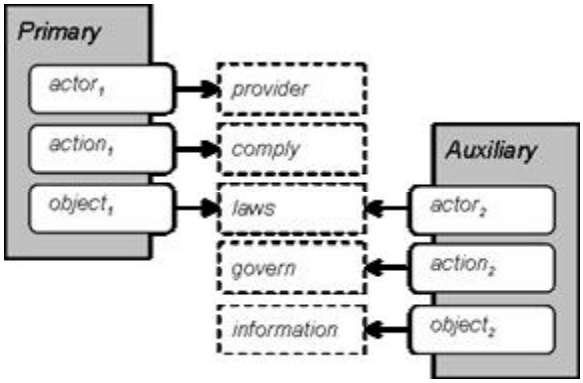

Figure 6: Object value shared by object, actor.

All of the goals that instantiated the above semantic model involved state and federal laws as both objects and actors. However, other goals with this relationship are foreseeable in policy documents.

3.4. Range of Possible Models

The above examples provide a glimpse of the range of possible models. In the first three models (see Figures 3, 4 and 5), each parameter that connects the component carries a different meaning to that component: (1) the purpose or intentionality of the primary action; (2) the instrument or mechanism to achieve the primary action; and (3) the transference of the primary action directly to the object that is another activity. In the last model (see Figure 6), the other component further distinguishes the object of the primary component.

In Table 1, we present the occurrences ***(Occur.)*** of different model parameters across the goal subset in addition to how many had auxiliary components as values ***(Comp.)*** The table includes two parameters not mentioned previously, the source and target parameters. The *source* parameter specifies the origin of information objects, whereas the *target* specifies the destination. The parameter values of the purpose are always an auxiliary component while this is never the case for parameter values of the target. Of the 100 parameterized goals, only 31 models were composed exclusively from actor, object, and action parameters. This suggests that the models are generally composed from a variety of parameters; thus, there is no single model configuration that can describe all parameterizations.

Table 1: Model Parameter Occurrences

Model Parameter	*Occur.*	*Comp.*
Action	113	0
Actor	113	0
Instrument	20	3
Object	113	10
Purpose	15	15
Source	23	5
Target	39	0

4. Analysis Results

The tool we developed to support analysis across instantiated semantic models includes a static interpreter for parsing a context-free grammar based on the modeling notation. The tool supports semantic queries over parameterized goals represented in the CFG. These queries have been used to perform quantitative and qualitative analysis of the goals.

4.1. Context-free grammar

In order to ensure correctness of the semantic models throughout the parameterization process, the models are formally expressed in a context-free grammar (CFG), included in Appendix A. The CFG extends the modeling notation to internally account for conjunctions, disjunctions and negations, and provides capabilities for automated analysis through queries. A static interpreter was developed to validate the CFG and automate queries.

In the CFG, an RNLS with logical conjunctions and disjunctions that are attributed to a single parameter value require special treatment. For example, the objects of an activity in an RNLS might be "employees or contractors." In this case, the restricted statement is divided into two statements, one whose object is "employees" and another whose object is "contractors". Such disjunctions have been encountered with actions, objects and purposes, and each is handled in the same fashion.

To limit the burden placed on the user, the CFG includes special operators to describe conjunctions and disjunctions while defining special interpretations that are handled by the static interpreter. Conjunctions are handled by interpreting the instance relation δ as a set relation with a new conjunction operator. In contrast, disjunctions describe different interpretations of a semantic model for each value. For example, the interpretation of disjunctions v_1, v_2 for a parameter p in Figure 7 includes cloning the model instance I for each value v_1, v_2, … , v_n in a disjunction and assigning each distinct value v_i to the same corresponding parameter p in one of the cloned models I_i. For n separate disjunctions there are 2^n total clones of the semantic model. The cloning process is automated by the static interpreter.

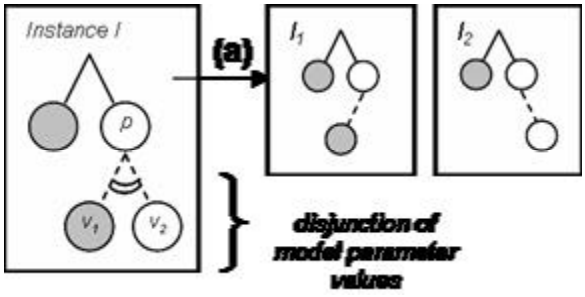

Figure 7: Interpreting a disjunction of values.

Queries are expressed in the CFG as semantic models with the addition of special query variables (written *?name* for the *name* of the variable) that replace parameters and values. When a query matches a model instance, the corresponding parameter or value is stored by variable name. The variable names are used in the tool to access the results from successful queries. The presence of a query variable in a model expression distinguishes a regular model instance from a query in the CFG.

4.2. Quantitative analysis

The quantitative results from our analysis show the distribution of rights and obligations for the actors: customer, provider, and third-party. In our analysis, rights are equivalent to permissions. In the parameterization, we distinguished rights and obligations by the modal associated with each RNLS. If a RNLS that derived the primary component used the modal "may" we identified the action as a right of the actor. If the RNLS used the modal "will" then the action was identified as an obligation of the actor.

Nine queries were executed that identify the rights and obligations of customers, providers, and third-parties. Third-parties include subsidiaries, affiliates and non-affiliates. The query responses were tallied to determine the total number of rights versus obligations per actor. Because the 100 goals are not equally distributed throughout the entire 1,200 goals in the PGMT, the totals for vulnerabilities *V* and protections *P* per actor were weighted by the total number of occurrences *O* within the 100 policies. The frequency was calculated by the formula $(V + P) \times O / 100$. Results of the queries are shown in Table 2 as the distribution of vulnerabilities (***V***), protections (***P***), and the frequency (***F***) across the PGMT repository.

Privacy protection goals express ways in which sensitive information is protected [2]. *Privacy vulnerabilities* reflect ways in which sensitive information may be susceptible to privacy invasions [2]. Whereas this expression of privacy protection goals and vulnerabilities enables more semantic analysis than is possible given an original privacy policy document, the analysis process is still human intensive and tedious. Our analysis tool is thus a significant improvement over our prior means of conducting this kind of analysis.

Table 2: Distribution of Rights and Obligations

Stakeholder	*V*	*P*	*F*
Consumer, Obligations	2	12	3.86%
Consumer, Rights	0	5	0.55%
Provider, Obligations	11	39	65.23%
Provider, Rights	39	0	29.79%
3rd-party, Obligations	0	5	0.57%

Several observations are worth noting from this analysis. The total obligations of customers that are vulnerabilities (2) include "accept the terms of use to access services." Otherwise, customer obligations are generally protections. All of the goals that are both obligations of providers and vulnerabilities (11) are in fact disclaimers of responsibility or negated protections. All of the rights of providers (39) were vulnerabilities. Rights of third-parties, however, were not found in the goal subset. These results reflect the types of activities that are most frequently emphasized in the 100 privacy policy documents in the PGMT.

4.3. Qualitative analysis

The results from our qualitative analysis demonstrate the comparability of goals using the semantic models. In order to show this capability, we present the results from an example query that asks the question *"With whom and what type of information is shared?"* The answer includes the goal ID for each matching parameterized goal. In this query, we restrict the action to "share" and the object of the primary activity to types of information and let the goal ID, actor, and target parameters range over any possible value. In this example, the "target" is the recipient of the action "share." Each row in Table 3 represents a result from the query. The repetition of the goal IDs among responses is characteristic of model cloning resulting from disjunctions in the original goal statement.

Table 3: Results from Qualitative Analysis

ID	*Object*	*Target*
155	transaction information	subsidiary
155	experience information	subsidiary
822	PII	affiliate
822	PII	service-provider
954	information	third-party
954	statistics	third-party

156	transaction information	affiliate
156	experience information	affiliate
170	PII	subsidiary

The query results demonstrate the comparability of instantiated models and corresponding RNLS. The ability to formulate such queries is prerequisite to tasks such as automated conflict identification and policy categorization. We see qualitative analysis based on queries playing a more significant role in the future of authoring and auditing machine enforceable policies.

5. Related Work

This section provides an overview of the two more relevant areas of related work: (a) policy languages with expressive capability parallel to our observations from the semantic models, (b) related approaches that transform requirements artifacts such as goals into conceptual models or semantic graphs.

5.1. Policy Languages

Three existing policy languages (P3P, EPAL and Ponder) support policy specification as we now discuss.

The Platform for Privacy Preferences Project (P3P) was developed to specify privacy practices that govern information transfer between a user agent and a website [6]. P3P can express purpose using either a pre-defined element or "other-purpose" tag that contains an unstructured natural language statement. The pre-defined element includes such purposes as: fulfilling the current transaction, website and system administration, and telemarketing. Unlike the purpose in our semantic models, purposes in P3P are not always structurally comparable [7]. This limitation introduces a source of ambiguity in P3P, because the *other-purpose* element may re-state purposes without any indication of similarity. Moreover, we know that organizations have been slow to adopt P3P because different user agents can interpret the same policies in different ways [8]; thus, policy comparisons are not reliable.

The Enterprise Privacy Authorization Language (EPAL), introduced by IBM, is used to define policies describing transactions between two organizations sharing a common vocabulary [9]. EPAL supports user-defined purposes that can be organized into hierarchies. Purpose hierarchies enable the comparison of purposes through common ancestors. These hierarchies are more expressive than P3P purposes; however, purposes in EPAL are still only predicate-based descriptions. In contrast, our semantic models demonstrate how purposes can be further decomposed into at least three additional vectors: actor, action and object, to support richer semantic comparisons. For example, the purposes "to market by telephone" and "to market by postal mail" share the same action "market" yet differentiate themselves through unique instruments, "telephone" and "postal mail," respectively. Preserving these semantic relationships in our models enables unforeseeable comparisons between purposes that are otherwise lost in predicate-based hierarchies that combine multiple semantic relations in a single predicate.

The Ponder language, introduced by Lupu et al., was developed for the specification of network management and security policies for distributed systems [10]. Ponder supports expressions for permissions, obligations and delegation of rights. Ponder lacks explicit support for purposes and instruments, although, they may be relegated to a condition in a permission or obligation. It expresses conditions as either Boolean predicates or evaluations. Because this is a non-standard usage that is unspecified in Ponder, there are no formal guidelines for this type of expression. In our semantic models, we explicitly and consistently support purposes and instruments.

5.2. Knowledge-based approaches

Three knowledge-based approaches that transform requirements artifacts into conceptual or semantic graphs are relevant to the approach proposed herein.

Conceptual graphs (CGs) have been used to formally represent concepts including actors and processes [11]. In CGs, nodes represent concepts that are connected by semantic relationships denoted by arcs. In general, nodes and arcs in CGs are unrestricted in their expressive capabilities; a single node may represent one entity or a complex transaction composed of several entities. The power of abstraction in CGs makes interpretation subjective and requires either strict modeling guidelines or a restricted ontology to ensure separate, conceptually-related graphs are comparable.

CGs provide no guidelines to restrict the labeling of nodes and arcs to exclusive information types. For example, two nodes in a CG labeled "withPurpose" and "hasPurpose" may be synonymous to the reader; however the relationship between "has" and "with" in this case is lost in the node labeling strategy. Alternatively, the relationships "with" and "has" could have been specified using arcs. Unlike CGs, our semantic models enforce specific guidelines that ensure parameter values are limited to single parts-of-speech that represent atomic concepts. Relationships like "with" and "has" are consistently subsumed by the same parameters, ensuring relevant information remains comparable. Our models use separate relations to differentiate between abstract associations and instance data. This separation enables comparison by compartmentalizing the variable data from the conceptual relations; an ambiguity in standard CGs.

Delugach et al. present an algorithm for converting requirements specifications encoded in Entity-Relationships (ER), data flow, or state transition diagrams into CGs with temporal extensions [12]. Our parameterization process begins with natural language goals, not structured specifications, and seeks to derive comparable semantic models while avoiding the unbound abstraction problems associated with CGs mentioned earlier.

Koch et al. describe a framework that combines semantic graphs with goal-oriented policies [13]. The goal-oriented policies are derived from requirements specifications and defined using templates with attributes including subject, action, target object, and modality that determine authorization or obligation policies. The templates are populated using natural language requirements that describe discrete activities. Our semantic models are more expressive than the goal templates given their ability to represent purpose (see Figure 3), instrument (see Figure 4), as well as actors and objects distinguished by separate activities (see Figure 6). Unlike Koch et al., our approach has also been validated using an extensive repository of privacy goals.

Michael et al. describe a process intended to transform natural language policy statements into logical representations [14] in which natural language policies describe policy requirements without the structural advantage of specifications or goals. Their approach employs an automated pipeline built from lexical and semantic analyzers including part-of-speech tagging, morphological reductions, and rule-based phrase and clause transforms. Because this approach accepts unstructured natural language, the transformation is as effective as the (possibly incomplete) rules generated from previously encountered statements. Unfortunately, Michael et al. do not demonstrate how their logical representation enables policy analysis. Our approach avoids the complexity associated with parsing the full scope of natural language by using restricted natural language statements that describe at most one activity. In addition, we demonstrate how our semantic models can be used in quantitative and qualitative policy analysis.

6. Discussion and future work

This paper proposes a generalizable process for developing semantic models from privacy policy statements (or goals) and discusses a tool we developed to support quantitative and qualitative analysis of policy statements. The semantic models developed using our parameterization process can be employed in policy management and analysis. Management and analysis tasks of interest include automatic generation of hierarchical purpose taxonomies, automatic re-construction of RNLS(s) from semantic models and conflict identification. Purpose taxonomies, such as those supported by EPAL, categorize purposes by combining multiple semantic relationships. In general, taxonomies organize policies into categories, possibly reducing the number of factors that distinguish policies. Identifying which specific combinations of semantic relationships are desirable in taxonomies will enable automatic taxonomy generation. An approach to reconstructing the RNLS from complete semantic models is currently under investigation. The approach has been successful using templates; however, there are too many templates to describe each variation of parameters in all possible models. We plan to continue this investigation to generalize the natural language correspondences and automate the re-construction of RNLS(s). Conflict identification is required for policy alignment tasks between parties [1]. Policy conflicts can be identified without characterizing the source of the conflict. Understanding the source of conflicts through specific semantic relations will at a minimum narrow the scope of redress and may even propose methods for partially automating conflict resolution.

We foresee semantic models playing a role in policy management and analysis tasks, but we must still address the limitations of our approach. Using the semantic parameterization process, we were able to completely parameterize 87 of the 100 privacy goals. The remaining 13 goals were not completely parameterized due to limitations in the context-free grammar including the lack of support for representing values over a continuous range and temporal relations.

Range values occurred in one goal with "children under 13 years of age." The context-free grammar presently supports discrete ranges, such as a complex conjunction and disjunction of elements. We are currently adding support for representing continuous ranges via the introduction of new operators.

Temporal relations occurred in some goals that were not completely parameterized. In each of the model instances with shared objects, the action value of the additional component was a past-tense verb unlike the action value of the primary component. For example, "information provided by the customer" uses the past-tense verb "provided." In addition, other components were related to primary components using temporal conditions. These conditions most often coincided with the conjunction "unless" and the preposition "upon" (preconditions). For example, a customer right may be withheld "unless the customer initiates the transaction" or a provider obligation must be fulfilled "upon customer notification." Each of these examples relates the primary activity conditionally with the completion of a separate activity. Temporal relations were also identified from the adverbs "annually," "monthly,"

"periodically," and "repeatedly." Adverbs can easily be treated as attributes to actions in much the same way as adjectives are handled for actors and objects.

Finally, we recognize that the RNLS restatement process, whether applied to goals or directly to policy statements, may change the meaning from what was intended in the original policy documents. For this reason, we foresee the RNLS(s) and semantic models playing a direct role in the authorship process; when policy authors need to specify policy semantics.

Acknowledgements

The authors thank Qingfeng He and Will Stufflebeam for their helpful comments.

References

[1] Antón, A. I., Earp, J. B., Bolchini, D., He, Q., Jensen, C., and Stufflebeam, W. "The Lack of Clarity in Financial Privacy Policies and the Need for Standardization," *IEEE Security & Privacy*, 2(2), pp. 36-45, 2004.

[2] Antón, A.I., Earp, J. B., "A Requirements Taxonomy for Reducing Website Privacy Vulnerabilities." *Journal of Requirements Engineering,* 9(3), pp.169 – 185, 2004.

[3] Antón, A. I., He, Q., and Baumer, D. *IEEE Security & Privacy*, July-August, v. 2, no. 4, pp. 2-9, 2004.

[4] Bandara, A. K., Lupu, E. C., Moffett, J., Russo, A., "A Goal-based Approach to Policy Refinement." *5th IEEE Workshop on Policies for Distributed Systems and Networks (POLICY'04)*, London, June 2004, pp. 229 – 239.

[5] Antón, A.I. E. Bertino, N. Li, and T. Yu. "A Roadmap for Comprehensive Online Privacy Policy Management," Purdue University CERIAS Technical Report #TR 2004-47, 2004.

[6] Cranor, L., Langheinrich, M., Marchiori, M., Pressler-Marshall, M., Reagle, J., "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification", Recommendation of the Word-Wide-Web Consortium, April 2002.

[7] Stufflebeam, W., Antón, A. I., He, Q., and Jain, N. "Specifying Privacy Policies in P3P and EPAL: Lessons learned." *ACM Workshop on Privacy in Electronic Society (WPES'04)*, Washington, D.C., pp. 35-36. October 2004.

[8] Li, N., Yu, T., and Antón, A. I. "A Semantics-Based Approach to Privacy Languages", abstract in *IEEE Symposium on Security and Privacy*, 2003.

[9] Karjoth, G., Schunter, M., Waidner, M. "Platform for Enterprise Privacy Practices: Privacy-Enabled Management of Customer Data" *2nd Int'l Workshop on Privacy Enhancing Technologies*, San Francisco, California, Lecture Notes in Computer Science 2482, pp. 69 – 84, April 2002.

[10] Lupu, E., Sloman, M., Dulay, N., Damianou, N. "Ponder: Realizing Enterprise Viewpoint Concepts" *4th Int'l Conf. on Enterprise Distributed Object Computing*. Japan, pp. 66 – 75, Sept. 2000.

[11] Mineau, G. W., "From Actors to Processes: The Representation of Dynamic Knowledge Using Conceptual Graphs." *6th Int'l Conf. on Conceptual Structures: Theory, Tools, and Applications (ICCS-98)*. Montpellier, France, pp. 65 – 79, August 1998.

[12] Delugach, H. S., "Specifying Multiple-Viewed Software Requirements with Conceptual Graphs." *Journal of Systems and Software*, vol. 19, pp. 207 – 224, 1992.

[13] Koch, T., Krell, C., Kraemer, B., "Policy Definition Language for Automated Management of Distributed Systems." *2nd IEEE Int'l Workshop on Systems Management (SMW'96)*, Toronto, Canada, p. 55, June 1996.

[14] Michael, J. B., Ong, V. L., Rowe, N. C., "Natural Language Processing Support for Developing Policy-governed Software Systems." *39th Int'l Conf. on Technology of Object-Oriented Languages and Systems (TOOLS-39)*, Santa Barbara, California, p. 263, August 2001.

[15] Parr, T. J. "ANTLR: a predicated-LL(k) parser generator." *Journal of Software Practice and Experience,* 25(7), pp. 789 – 810, July 1995,

Appendix A

The following CFG represented using BNF notation is modeled on the original syntax for the ANTLR parser generator toolset developed by Terence Parr [15]. The ANTLR grammar syntax combines aspects of regular expressions with standard productions for improved legibility.

```
<start> ::= (<term>)+
<term> ::= (IDENT | VAR) <block>?
<block> ::= LBRKT (<stmt>)+ RBRKT
<ref> ::= NEGATE? <abs>
<abs> ::= <term> (ABS <abs>)?
<set> ::= <item> ((OR | AND) <set>)?
<item> ::= LPAREN <set> RPAREN | <ref>
<stmt> ::= <ref> (EQ (<value> | <item>))?
<value> ::= NUMBER | STRING
```