A Formal Semantics for P3P

Ting Yu
Department of Computer Science
Cyber Defense Laboratory
North Carolina State University
yu@csc.ncsu.edu

Ninghui Li
Department of Computer Sciences and Center for Education and Research in Information Assurance and Security
Purdue University
ninghui@cs.purdue.edu

Annie I. Antón
Department of Computer Science
North Carolina State University
anton@csc.ncsu.edu

ABSTRACT

The Platform for Privacy Preferences (P3P), developed by the W3C, provides an XML-based language for websites to encode their data-collection and data-use practices in a machine-readable form. To fully deploy P3P in enterprise information systems and over the Web, a well-defined semantics for P3P policies is a must, which is lacking in the current P3P framework. Without a formal semantics, a P3P policy may be semantically inconsistent and may be interpreted and represented differently by different user agents; it is difficult to determine whether a P3P policy is indeed enforced by an enterprise; and privacy policies from different corporations cannot be formally compared before information exchange. In this paper, we propose a relational formal semantics for P3P policies, which precisely and intuitively models the relationships between different components of P3P statements (i.e., collected data items, purposes, recipients and retentions) during online information collection.

The proposed formal semantics is an important step towards improving P3P, making it more appropriate to be integrated with business practice and ultimately accelerating the large-scale adoption of P3P across the Internet.

1. INTRODUCTION

Privacy is increasingly a major concern that prevents Internet users from fully enjoying the convenience, variety and flexibility offered by Web sites, Web services, and other e-services. Some experienced Internet users tend to avoid websites that ask for personal information because they fear potential misuses of their private information [12]. Effective protection of individuals' privacy is in the best interest of Internet users as well as e-service providers.

The W3C's Platform for Privacy Preferences Project (P3P) [23] is one major effort to improve today's online privacy practices. P3P enables websites to encode their data-collection and data-use practices in a machine-readable XML format, known as P3P policies [11]. The W3C has also designed APPEL (A P3P Preference Exchange Language) [19], which allows users to specify their privacy preferences. Ideally, through the use of P3P and APPEL, a user's agent should be able to check a website's privacy policy against the user's privacy preferences, and automatically determine when the user's private information can be disclosed. In short, P3P and APPEL are designed to enable users to play an active role in controlling their private information.

Since proposed, P3P has received broad attention from both industry and the research community, and has been gradually adopted by companies. On the other hand, the full deployment of P3P in enterprise information systems has raised many challenging questions. For example, P3P represents an enterprise's promise to users about its privacy practice. How can we ensure that an organization and its customers have a common understanding of these promises? P3P promises must be fulfilled in the services provided by enterprises. How can a company guarantee that its P3P policy is correctly enforced in those applications? Privacy policies also concern information exchange between enterprises. How can we compare two organizations' privacy policies and ensure that no privacy violation can happen during an information exchange? To address these challenges, a well-defined semantics for P3P is a must; unfortunately, it is lacking in the current P3P framework.

This paper describes our initial efforts to improve P3P and make it more appropriate and convenient for integration with enterprise information management. We first demonstrate that a well-defined semantics for P3P will play a crucial role for privacy policy deployment. Then we propose a relational formal semantics for P3P. A P3P policy is a collection of statements, each of which describes the purpose, retention and recipient of a piece of collected data. Our formal semantics for P3P transforms each P3P policy into five relations: data purpose, data recipient, data retention, data collection and data category. Additionally, integrity constraints are introduced to maintain a P3P policy's semantic consistency.

The remainder of this paper is organized as follows. Section 2 demonstrates the need for a formal semantics in P3P for enterprise privacy protection. The relevant work is discussed in Section 4. Section 3 provides an overview of P3P and presents a formal semantics for it. Finally, a summary

ACM Workshop on Secure Web Services, October 29, 2004, Fairfax VA, USA.
ACM X-XXXXX-XXX-X.

and our plans for future work appear in Section 5.

2. THE NEED FOR A FORMAL SEMANTICS FOR P3P

P3P policies serve as a basis for users to familiarize themselves with an enterprise's privacy practices and to control the disclosure of their private information, when accessing Web sites and Web services. On the other hand, a P3P policy alone cannot contribute much to the proper protection of users' privacy. Instead, a P3P policy has to interact with a variety of enterprise information system components, such as customer information collection subsystems, information processing subsystems, as well as data dissemination and sharing subsystems. Such interactions often require that a P3P policy have a precise, unambiguous meaning; this is the fundamental reason why a formal semantics for P3P is needed. In this section, we discuss how a P3P policy interacts with other entities and system components in online privacy protection, and the need for formal P3P semantics during these interactions.

Interactions with end users: representation of P3P policies to users. End-users are the critical stakeholders in online transactions because ultimately their privacy is at stake. A user agent often needs to present a P3P policy to users, but one major criticism of P3P is that a P3P policy may be interpreted and presented differently by different user agents [10, 20]. Companies are thus reluctant to provide P3P policies on their websites, fearing that the policies may be misrepresented [10, 22]. Quoting from CitiGroup's position paper [22], "The same P3P policy could be represented to users in ways that may be counter to each other as well as to the intent of the site." "... This results in legal and media risk for companies implementing P3P that needs to be addressed and resolved if P3P is to fulfill a very important need."

Part of this problem is caused by the complexity of privacy policy vocabularies and ambiguous terminology used in P3P. Another fundamental reason underlying the aforementioned technical difficulty is that the need for a semantics was apparently overlooked in the initial design of P3P, leaving too much freedom for P3P policies to be misinterpreted and misrepresented by user agents. As we show in this paper, the problem is not simply due to P3P's vocabulary ambiguities, it is also effected by how the different components (i.e., collected data items, purposes, recipients and retentions) in a P3P statement interact.

We recognize that a formal semantics by itself does not eliminate the problem of potential misrepresentation of P3P policies. Standards and guidelines for consistent user interface and vocabulary representation are also necessary. However, a formal semantics for P3P is a necessary step, without which there is little hope to completely solve P3P's misrepresentation and ambiguity problems.

Interactions with user agent and privacy preference language. The P3P framework allows users to specify their privacy preferences. These privacy preferences state which enterprise privacy practices are acceptable to the end-user so that personal information disclosures can be allowed. During online interactions, user agents (e.g, browsers) act on behalf of users and dynamically match enterprises' privacy policies with users' privacy preferences. As previously mentioned, the W3C designed APPEL (A P3P Preference Exchange Language) [19] to enable users to specify their privacy preferences.

It has been widely recognized that APPEL is complex and problematic [3, 13, 14, 24]. For example, expressing one's privacy preferences in APPEL is a highly error-prone process. A seemingly correct APPEL privacy preference often behaves in a counterintuitive manner. As pointed out by Agarawal et al. [3, 20], even the designers of APPEL made mistakes in the APPEL specification [19]. Quoting from the position paper by the Joint Research Center of the European Commission [13], "a preference exchange language is a very necessary part of P3P. However, there are various problems with the preference expression language (APPEL). Constructing the logic of matching patterns is very complex, and involves various inherent contradictions." As pointed out in [13], because P3P allows the same policy to be encoded differently, a preference specified in APPEL may accept one encoding of the policy and reject another encoding of the same policy.

We argue that the problems with APPEL come directly from its syntax-based design. It is designed to match the XML representation of a P3P policy, rather than the *underlying meaning* of a P3P policy. This design is not surprising, as no formal semantics for P3P policies had been proposed when APPEL was designed. It is fundamentally more difficult to express privacy preferences in a syntax-based preference language compared with doing so in a semantics-based language. To express one's preferences, one starts by thinking about the meanings of what policies should be accepted or rejected. Using a syntax-based language, one also needs to think about how these meanings are syntactically encoded in P3P, covering all the possible representations of the same meaning.

We believe that a preference language should query the meaning of a privacy policy instead of its syntactical representation. Of course, in order to have a semantics-based preference language, there must exist a formal semantics for P3P. In [20], we proposed such a semantics-based preference language, SemPref. SemPref is based on the semantics presented here. Compared to APPEL, SemPref has a much simple and intuitive syntax and can be easily used to rigorously specify privacy preferences. Due to space limitation, we will not discuss the details of SemPref in this paper.

Interactions with enterprise information systems. A P3P policy represents a privacy promise from the enterprise to the end users. The act of posting a P3P policy contributes nothing to the protection of end users' privacy, if the policy is not enforced. In recent years, we have seen growing interest in developing technologies for managing user information in a privacy-preserving manner. For example, IBM developed Enterprise Privacy Authorization Language (EPAL) [16, 18] to enable an enterprise to specify internal data access control policies. There is a need to ensure that the internal access control policy is consistent with the P3P policy.

Karjoth et al. [17] proposed to generate P3P policies from EPAL policies, to ensure the consistency between the two. We disagree with this approach. Privacy policies represent long-term promises made by an enterprise to its end users and are determined by concerns about business practices and law. On the other hand, access control policies represent internal data handling practices, which may change much more frequently. It is undesirable to change an enterprise's promises to its customers every time an internal access control rule changes and yet it is important to deal with these changes. Because P3P policies and EPAL policies may be authored and changed independently, these changes need to be compared to ensure consistency. To do so requires a formal semantics for P3P.

Interactions within a P3P policy. As we discuss in this paper, a policy specified in P3P may be internally inconsistent. A rigorous consistency checking needs to be based on a formal semantics for P3P.

Interactions with other enterprises. Information sharing between enterprises should also be regulated by the enterprises' privacy policies. To ensure proper privacy protection, we need to compare their privacy policies to ensure they comply with each other. For example, in P3P, the recipient of a data item may be "same", meaning that the data item may be shared with "legal entities who use the data on their own behalf under *equable* practices". To determine whether a data item can be shared without violating a P3P policy, one needs to determine whether another entity's policy is equable. Such a policy comparison also requires a formal semantics for P3P.

3. A FORMAL SEMANTICS FOR P3P

To the best of our knowledge, we are the first to have identified the lack of a formal semantics as a key issue underlying P3P [20]. So far there have been no formal semantics to ground P3P.

Although at first sight P3P appears to be a simple XML-based language, developing a formal semantics for it is quite challenging. There are often several ways to interpret a particular P3P policy and the P3P specification does not clearly state which way is correct. There are also many ways for a P3P policy to be semantically inconsistent as we discuss in this section.

3.1 An Overview of P3P's Syntax

Each P3P policy is specified by one `POLICY` element that includes the following major elements.

One `ENTITY` element: identifies the legal entity making the representation of privacy practices contained in the policy.

One `ACCESS` element: indicates whether the site allows users to access the various kind of information collected about them.

One `DISPUTES-GROUP` element: contains one or more `DISPUTES` elements that describe dispute resolution procedures to be followed when disputes arise about a service's privacy practices.

Zero or more `EXTENSION` elements: contain a website's self-defined extensions to the P3P specification.

And one or more `STATEMENT` elements: describe data collection, use and storage. A `STATEMENT` element specifies the data (e.g. user's name) and the data categories (e.g. user's demographic data) being collected by the site, as well as the purposes, recipients and retention of that data.

There are two kinds of P3P statements. The first kind contains the `NON-IDENTIFIABLE` element, which is used to indicate that either no information will be collected or information will be anonymized during collection. The second kind does not contain the `NON-IDENTIFIABLE` element; this is the commonly used one. In this paper, we will focus on the latter. A brief discussion of statements with `NON-IDENTIFIABLE` element is given in Section 3.6.

Figure 1 provides an example of a P3P statement. To conserve space in this paper, we employ the succinct representation that appears in the right-hand column of the figure. Each such statement contains the following:

One `PURPOSE` element, which describes for which purpose(s) the information will be used. It contains one or more pre-defined values such as current, admin, individual-analysis and historical. A purpose value can have an optional attribute 'required', which takes one of the following values: opt-in, opt-out, and always. The value 'opt-in' means that data may be used for this purpose only when the user affirmatively requests this use. The value 'opt-out' means that data may be used for this purpose unless the user requests that it not be used in this way. The value 'always' means that users cannot opt-in or opt-out of this use of their data. Therefore; in terms of strength of data usage, 'always' $>$ 'opt-out' $>$ 'opt-in'. In Figure 1, `PURPOSE` is admin and the attribute 'required' takes the value opt-in.

One `RECIPIENT` element, which describes with whom the collected information will be shared. It contains one or more pre-defined values such as ours, delivery and public. A recipient value can have an optional attribute 'required', which is similar to that of a `PURPOSE` element. In Figure 1, `RECIPIENT` is public.

One `RETENTION` element, which describes for how long the collected information will be kept. It contains exactly one of the following pre-defined values: no-retention, stated-purpose, legal-requirement, business-practices and indefinitely. In Figure 1, the `RETENTION` value is indefinite.

One or more `DATA-GROUP` elements, which specify what information will be collected and used. Each `DATA-GROUP` element contains one or more `DATA` elements. Each `DATA` element has two attributes. The mandatory attribute 'ref' identifies the data being collected. For example, '#user.home-info.telecom.telephone' identifies a user's home telephone number. The 'optional' attribute indicates whether or not the data collection is optional. A `DATA` element may also contain a `CATEGORIES` element, which describes the kind of information this data item is, e.g., financial, demographic and health. In Figure 1, `DATA` is postal info.

Zero or one `CONSEQUENCE` element, which contains human-readable contents that can be shown to users to explain the data usage practice's ramifications and why the usage is useful.

EXAMPLE 1. *A P3P Example Statement*

```
<STATEMENT>
 <PURPOSE><admin required="opt-in"/></PURPOSE>
 <RECIPIENT><public/></RECIPIENT>
 <RETENTION><indefinitely/></RETENTION>
 <DATA-GROUP>
  <DATA ref="#user.home-info.postal"></DATA>
 </DATA-GROUP>
</STATEMENT>
```

```
stmt(
 purpose:  {admin(opt-in)}
 recipient:  {public}
 retention:  {indefinitely}
 data:  {#user.home-info.postal}
)
```

Figure 1: An Example P3P Statement. The XML representation appears on the left side and a more succinct representation on the right side.

3.2 Towards a formal semantics of P3P

Statements comprise the core of a P3P policy, as they specify a website's data-collection and data-use practices. They are also the most complicated parts of a P3P policy. In this paper, we limit the scope of the formal semantics to statements. To develop a formal semantics for P3P statements, we must first determine the relationships among the four major components (purpose, recipient, retention and data) of a P3P statement.

In the statement in Figure 1, the three components (purpose, recipient and retention) all refer to the same data item '#user.home-info.postal'; however, for the statement to have a precise meaning, one must also determine how these components interact. We consider two interpretations. In the first interpretation, all three components are related, i.e., the purpose, the recipient and the retention are about *one* data usage. In Figure 1, the postal information will be used for the admin purpose (technical support of the website and its computer system); the information will be shared with the public and will be stored indefinitely. For this statement, this interpretation seems counterintuitive, because there is no need to share the data with the public for the admin purpose. Furthermore, it is not clear whether this data usage is required or optional, since the 'required' attribute has the 'opt-in' value for purpose but the default 'always' value for recipient. The explanation for this statement, provided by one of the P3P architects [8], is that the data item '#user.home-info.postal' will always be collected and shared with the public. Additionally, if the user chooses to opt-in, their postal information will be used for the admin purpose. In other words, whether the individual's postal information will be shared with the public does not depend upon whether or not the information is used for the admin purpose.

This leads us to the second interpretation, in which purpose, recipient and retention are considered orthogonal. In this interpretation, a P3P statement specifies three relations: the purposes for which a data item will be used, the recipients with whom a data item will be shared, and how long the data item will be stored. Even though these relations are specified in the same statement, they are not necessarily about a single data usage. Given this *data-centric* interpretation, the following three P3P policies will have the same meaning in the sense that all relations contain a data component:

EXAMPLE 2. *Three P3P policies that have the same meaning.*

Policy 1:

```
stmt( data: {#user.home-info.telecom,
             #user.bdate(optional)},
      purpose: {individual-analysis,
                telemarketing(opt-in)},
      recipient: {ours},
      retention: {stated-purpose} )
```

Policy 2:

```
stmt( data: {#user.home-info.telecom,
             #user.bdate(optional)},
      purpose: {individual-analysis},
      recipient: {ours},
      retention: {stated-purpose})
stmt( data: {#user.home-info.telecom,
             #user.bdate(optional)},
      purpose: {telemarketing(opt-in)},
      recipient: {ours},
      retention: {stated-purpose} )
```

Policy 3:

```
stmt( data: {#user.home-info.telecom},
      purpose: {individual-analysis,
                telemarketing(opt-in)},
      recipient: {ours},
      retention: {stated-purpose} )
stmt( data: {#user.bdate(optional)},
      purpose: {individual-analysis,
                telemarketing(opt-in)},
      recipient: {ours},
      retention: {stated-purpose} )
```

The fact that the same meaning may be encoded in several different ways makes it very difficult to correctly express privacy preferences in a syntax-based preference language such as APPEL. One representation can be accepted by a preference, but another representation could be rejected by the same preference. A preference language based on the formal semantics proposed herein will help ensure the same meaning will always be handled the same way.

We adopt this data-centric interpretation in the rest of this paper because this appears to be the intention of P3P's designers, as it is consistent with the P3P specification and the explanation we were given [8].[1] We briefly discuss semantics that take other interpretations in Section 3.7.

In the statements in Example 2, the data item '#user.bdate' is 'optional' but the purpose 'individual-analysis' is 'always'. This seems counterintuitive: if the collection of the data is optional, why is it always used for a certain purpose? According to Cranor [8], this means that the collection of '#user.bdate' is

[1] In fact, if the first interpretation is intended, it is more intuitive for the 'required' attribute to be associated with the whole statement rather than with both purpose and recipient.

optional, i.e, the user can choose not to provide the information. However, once the user provides it, it will be used for 'individual analysis'. The user cannot opt out of this purpose. This explanation is consistent with the data-centric interpretation in which the components are orthogonal. It also suggests that we need a fourth relation in the P3P semantics to specify whether data collection is required or optional.

In a P3P statement, each DATA element has a set of categories associated with it. Some categories are implicitly specified by the base P3P data schema whereas some others are specified explicitly in the statement. Thus we need another (fifth) relation to store the categories with which a data item is associated.

3.3 A data-centric semantics for P3P

We now propose a formal semantics for P3P policies. Recall that a P3P statement determines three relations, specifying the purpose, recipient, and retention associated with data items. We also need a fourth relation to specify whether the data collection is required or optional. Finally, we need a relation to store the categories (i.e. financial, health, demographic, etc.) associated with a data item. Thus, in the semantics, every P3P policy's data usage part is mapped onto five relations. The schemas for the five relations are given in Figure 2.

One can consider the semantics of a P3P policy as a database consisting of five tables (the previous subsection provides the justification and rationale for why we need these five tables). For example, the three policies in Example 2 all have the same semantics, given by the semantic database in Figure 3.

Given a set of P3P statements, it is straightforward to translate them into the data-centric semantics. A translation algorithm is given in Figure 4. Intuitively, for each data item in a statement, we pair it with each purpose, recipient and the retention in the statement, then insert the resulting pairs into corresponding relations.

3.4 Potential semantic inconsistencies in P3P policies

In general, any combinations of the values for purpose, recipient and retention are allowed in P3P. However, in a practical setting, semantic dependencies arise naturally between these values, making some of the combinations invalid. A P3P policy using invalid combinations is thus semantically inconsistent. This problem has been recognized [7, 21], and P3P's designers are beginning to address some of these conflicts [7]. Nonetheless, many places where potential conflicts may occur have not been previously identified. We now identify some additional classes of potential semantic inconsistencies in P3P.

ISSUE 1. *A P3P policy may be inconsistent because multiple retention values apply to one data item.*

P3P allows one data item to appear in multiple statements, which introduces a semantic problem. Recall that in each P3P statement, only one retention value can be specified, even though multiple purposes and recipients can be used. The rationale behind this is that retention values are mutually exclusive, i.e., two retention values conflict with each other. For instance, *no-retention* means that "Information is not retained for more than a brief period of time necessary to make use of it during the course of a single online interaction"[11]. And *indefinitely* means that "Information is retained for an indeterminate period of time"[11]. One data item cannot have both retention values. However, allowing one data item to appear in multiple statements makes it possible for multiple retention values to apply to one data item.

ISSUE 2. *A statement may have conflicting purposes and retention values.*

Consider a statement in a P3P policy that collects users' postal information for the purpose *historical* with retention *no-retention.* Clearly, if the postal information is going to be "... archived or stored for the purpose of preserving social history ...", as described by the *historical* purpose, it will conflict with *no-retention*, which requires that the collected information "... MUST NOT be logged, archived or otherwise stored"[11].

ISSUE 3. *A statement may have conflicting purposes and recipients.*

Consider a statement that includes all the purpose values (e.g., history, admin, telemarketing, individual-analysis, etc.) but only the recipient value *delivery* (delivery services). This does not make sense as one would expect that at least *ours* should be included in the recipients.

ISSUE 4. *A statement may have conflicting purposes and data items.*

Certain purposes imply the collection and usage of some data items. This has been recognized by the P3P designers and reflected in the guidelines for designing P3P user agents [7]. For example, suppose a statement contains purpose contact but does not collect any information from the categories physical and online. Then the statement is inconsistent because, in order to contact a user, "the initiator of the contact would possess a data element identifying the individual .... This would presuppose elements contained by one of the above categories"[7].

We suggest that all semantic inconsistency instances be identified and specified in the P3P specification. Completion of this work requires a detailed analysis of the vocabulary, which is beyond the scope of the current paper, ideally by the individuals who design and use these vocabularies.

3.5 Integrity constraints in the semantics

We handle the aforementioned semantic problems by employing integrity constraints in the semantics. If a P3P policy is translated into a semantics database that violates these constraints, then this policy is invalid. Such a set of integrity constraints can benefit both end users and websites. First, policies that are semantically inconsistent can be automatically rejected by user agents. Thus, the design of preference languages is simplified because we only need to handle semantically consistent policies. Second, a website may also use integrity constraints to detect semantic inconsistency in their policies and fix them in time, to avoid confusing Internet users.

Relation name	Field name	Domain of the field	Key for the relation
d-purpose	*data* *purpose* *required*	URI references to data items The P3P-defined purpose values {opt-in, opt-out, always}	(*data*, *purpose*)
d-recipient	*data* *recipient* *required*	URI references to data items The P3P-defined recipient values {opt-in, opt-out, always}	(*data*, *recipient*)
d-retention	*data* *retention*	URI references to data items The P3P-defined retention values	(*data*)
d-collection	*data* *optional*	URI references to data items {required, optional}	(*data*)
d-collection	*data* *category*	URI references to data items The P3P-defined category values	(*data*, *category*)

Figure 2: The schema for the five relations in the data-centric semantics for P3P.

d-purpose	*data*
	`#user.home-info.telecom`
	`#user.home-info.telecom`
	`#user.bdate`
	`#user.bdate`
		purpose
		`individual-analysis`
		`telemarketing`
		`individual-analysis`
		`telemarketing`
			required
			`required`
			`opt-in`
			`required`
			`opt-in`

d-recipient	*data*
	`#user.home-info.telecom`
	`#user.bdate`
		recipient
		`ours`
		`ours`
			required
			`required`
			`required`

d-retention:	*data*
	`#user.home-info.telecom`
	`#user.bdate`
		retention
		`stated-purpose`
		`stated-purpose`	

d-collection:	*data*
	`#user.home-info.telecom`
	`#user.bdate`
		optional
		`required`
		`optional`	

d-category [2]	*data*
	`#user.home-info.telecom`
	`#user.bdate`
		category
		`'Physical Contact Information'`
		`'Demographic and Socioeconomic Data'`	

Figure 3: The semantic database for the three P3P policies in Example 2, which have the same meaning.

In the formal semantics for P3P, we specify the following classes of integrity constraints:

Data-Centric Constraints. The keys in the relations imply four functional dependency constraints. For example, in the d-purpose relation, a pair (*data item, purpose value*) can only have one *required* value. This is a reasonable constraint because the three required choices are mutually exclusive. It does not make sense for the purpose of a data item to be, for example, both opt-in and required. The same constraint is also applied to the d-recipient relation and d-retention relation. Similarly, a website cannot specify that the collection of a data item is both optional and required. Therefore, the d-collection relation requires that no more than one optional value be associated with a data item. These constraints are implied by the definition of the semantics and do not need to be explicitly specified.

Data Hierarchy Constraints. Data items in P3P are organized into hierarchies. For example, '#user.home-info' would include '#user.home-info.postal', '#user.home-info.telecom' and '#user.home-info.online'. This introduces the potential for a semantics conflict. For example, if the collection of '#user.home-info' is required, which means that the collection of all data items under '#user.home-info' is required, then it does not make sense for the collection of '#user.home-info.online' to be optional. However, it may be reasonable for the collection of '#user.home-info' to be optional, but the collection of '#user.home-info.online' to be required, which would mean that the collection of '#user.home-info.postal' and '#user.home-info.telecom' are optional.

Based on the above observation, we define the following constraint. For any two tuples d-purpose(d_1, p_1, r_1) and d-purpose(d_2, p_2, r_2), if d_1 is more specific than d_2 and $p_1 = p_2$, then $r_1 \geq r_2$. ('always' $>$ 'opt-out' $>$ 'opt-in'). Similar constraints apply to the d-recipient relation and the d-collection relation.

The data hierarchy constraint for the d-retention relation is as follows. For any two tuples d-retention(d_1, r_1) and d-retention(d_2, r_2), if d_1 is more specific than d_2, then $r_1 \geq r_2$ ('indefinitely' $>$ 'business-practices', 'legal-requirement', 'stated-purpose' $>$ 'no-retention'; the middle three values are

```
translate-P3P-to-semantics (P3P-policy)
     foreach STATEMENT element S of P3P-policy
        foreach DATA element d in S.DATA-GROUP
           // update relations d-collection and d-category;
           if (d.optional does not exist) ∨ (d.optional=NO)
           then add (d.ref, REQUIRED) into d-collection;
           else add (d.ref, OPTIONAL) into d-collection;
           foreach category c in d.CATEGORIES
              add (d.ref, c) into d-category;
           // update relation d-purpose
           foreach purpose p in S.PURPOSE
              if p.required does not exist
              then add (d.ref, p, ALWAYS) into d-purpose;
              else add (d.ref, p, p.required) into d-purpose;
          // update relation d-recipient
          foreach recipient r in S.RECIPIENT
             if r.required does not exist
             then add (d.ref, r, ALWAYS) into d-recipient;
             else add (d.ref, r, r.required) into d-recipient;
          //update relation d-retention
          add (d.ref, S.RETENTION) into d-retention;
```

Figure 4: The procedure to translate a P3P policy into the data-centric semantics.

incomparable).

Semantic Vocabulary Constraints. As discussed in Section 3.4, many contradictions may arise if we carefully examine the semantics of P3P's pre-defined values. These can be specified as integrity constraints. For example, we may use the following integrity constraints to address some of the semantic inconsistencies among purposes, recipients and retentions.

CONSTRAINT 1. *If a data item is collected for the purpose* historical, *then its retention cannot be* no-retention.
$\forall p \in$ d-purpose $\neg\exists r \in$ d-retention, $p.data = r.data \wedge p.purpose = historical \wedge r.retention = no-retention$

CONSTRAINT 2. *If a data item is shared with a public fora, then its retention should be* indefinitely. *(From [11].)*
$\forall s \in$ d-recipient $r \in$ d-retention, $(s.data = r.data \wedge s.recipient = public) \rightarrow r.retention = indefinitely$

Identifying all the constraints to handle all such inconsistencies is beyond the scope of this paper, but these examples demonstrate the need for individuals who design these vocabularies to conduct a detailed analysis.

3.6 Dealing with P3P statements having the NON-IDENTIFIABLE element

A STATEMENT element in a P3P policy may optionally contain the NON-IDENTIFIABLE element, which "signifies that either no data is collected (including Web logs), or that the organization collecting the data will anonymize the data referenced in the enclosing STATEMENT" [11]. We call such statements non-identifiable statements.

From the above description, we see that the NON-IDENTIFIABLE element is used for two unrelated purposes in P3P. We argue that using the NON-IDENTIFIABLE element to signify that no data is collected is inappropriate. Intuitively, if a statement with the NON-IDENTIFIABLE element contains the DATA-GROUP element, it means that the data collected in this statement is anonymized. If such a statement does not have the DATA-GROUP element, it means that no data is collected. However, this statement is meaningless when the policy contains other statements that collect and use data. In general, the fact that a policy does not collect any data should not be specified at the level of a STATEMENT element; instead, it should be specified at the level of a POLICY statement. We suggest using a separate sub-element (or an attribute) for the POLICY element to denote that a policy collects no data.

Another issue that arises from having the NON-IDENTIFIABLE element is that a data item may appear both in normal statements and in non-identifiable statements. In this situation, it is not clear whether the data item is anonymized upon collection. According to Cranor [8], it may be possible that: "a company keeps two different unlinkable databases and the data is anonymized in one but not the other."

It is possible to extend our relation-based semantics for P3P to deal with the NON-IDENTIFIABLE element. The most straightforward way is to annotate an anonymized data item so that it is different from a normal data item, e.g., one may use '#user.home-info' to denote a normal data item and '#@.user.home-info' to denote an anonymized version of the same data.

3.7 Discussion

Because the P3P specification is often unclear on how to interpret P3P policies, we have to make some judgment calls in defining a formal semantics for P3P. We make these decisions based on the information we obtained from P3P architects and the rationales for these decisions are documented in this paper. The proposed semantics offers a strong step in the formal study of online privacy policies; however, it is not intended to be the only interpretation of P3P to be accepted by everybody.

Rather, we view the proposed semantics as a starting point for a standard semantics for P3P. We now explore alternative designs and related issues

Alternative Semantics for P3P. As mentioned in Section 3.2, other alternative semantics certainly exist. One attractive alternative is to have a purpose-centric semantics [1], where a data item along with a purpose determines other elements (i.e., recipients and retention) in a P3P statement. Similar to the data-centric semantics, a purpose-centric semantics may be modelled as two relations dp-recipient$(data, purpose, recipient)$ and dp-retention$(data, purpose, retention)$, where $data$ and $purpose$ form a primary key for both relations. Integrity constraints can be defined accordingly.[3] The rationale of this purpose-centric semantics is obvious. In practice, certain data are sometimes used for multiple purposes. Depending on the specific purposes, the data may be shared by different parties and may be kept for different periods of time. In this sense, the purpose-centric semantics represents a finer-grained interpretation of P3P whereas the data-centric semantics is relatively coarse-grained.

Both interpretations have their pros and cons, which reflect a tradeoff between expressiveness and ease of management. It is not apparent that one is always preferred than the other. In general, a coarse-grained semantics enables users to act more conservatively when disclosing information to a website, without worrying about complex relationships between the major components of P3P statements. In many situations, users may only want to know, for example, whether their information may be shared with third parties, without caring much for what purpose their information is shared. On the other hand, given a purpose-centric semantics of a P3P policy, we can always derive its corresponding data-centric semantics. Therefore, even though the purpose-centric semantics is adopted, through a presentation middleware, users can still enjoy the simplicity of the data-centric semantics.

The purpose-centric semantics reveals more details of the internal operations of an enterprise, which may allow users to make better informed decisions regarding private information disclosure. Meanwhile, revealing operation details may not be desirable to enterprises, since it imposes more legal responsibility on organizations, discouraging the adoption of P3P in the enterprise side.

Vocabulary Issues. Besides providing a set of pre-defined values, P3P also allows websites to define their own data schemas, so that privacy policies can be better tailored to fit specific applications. However, Internet users and websites often interact from different domains. They may not have any pre-existing knowledge about each other. We currently lack a mechanism that allows two parties to *dynamically* agree on a common vocabulary for the data schema definitions. Without such a mechanism, websites' self-defined data schemas will not be understood by user agents. In order to protect their privacy, users may have to reject any policies involving non-standard data schemas, or design very complex rules, hoping to cover all possible self-defined data schemas from a website.

[3]In fact, as shown in Section 3.2, some tricky issues may arise when one tries to handle the *required* attribute of purposes and recipients. Thus, more integrity constraints may need to be introduced. A detailed discussion about this topic is outside the scope of this paper.

4. RELATED WORK

A detailed description of P3P and APPEL can be found at [9, 11, 19]. Several implementations of P3P and APPEL have been developed, including Privacy Bird from AT&T Labs-Research [5], which can be integrated into users' Web browsers, and a Java-based implementation from JRC [6, 15]. Agrawal et al. [2] designed a server-centric architecture for P3P, where user privacy preferences are matched with websites' P3P policies at the server side. In this architecture, privacy policies are stored in a relational database, and users' APPEL preferences are translated into SQL queries. Though database techniques are used for privacy preference matching, this approach is still syntax based. Relational databases are only used as a means for storing the XML representation of policies, and preference matching is still done by matching the representation of policies. No formal semantics is defined for P3P in [2].

Many researchers have noted the limitations of P3P and APPEL [3, 13, 14, 21, 24]. Hogben [13, 14] identified the limitations of P3P in terms of cookie management, user interfaces and vocabularies. The ambiguity and awkwardness of APPEL was also pointed out in [13, 14]. Schunter et al. [21] also showed the ambiguity of P3P and argued that the current P3P specification lacks a clear guideline for policy design and interpretation. Suggested solutions included augmented consent models, more specific element definitions and a simplified syntax. Our work extends previous work by showing that the lack of a clear formal semantics is the fundamental reason for a variety of problems in P3P and APPEL.

Agrawal et al. [3] showed the limitations of APPEL with a series of plausible examples. They then proposed an XPath-based privacy preference language, XPref. XPref only uses a small subset of the XPath specification. Therefore, it can be efficiently evaluated. XPref has many advantages over APPEL in terms of clarity, ease of use and expressiveness. On the other hand, XPref is still a syntax-based preference language and, thus, cannot overcome APPEL's problems completely. The work reported in this paper is in part inspired by the analysis of APPEL in [3].

As evidenced by the recent JetBlue Airways case [4], it is becoming increasingly important for enterprises to effectively *enforce* their privacy policies in addition to simply specifying them. In [1], Agrawal et al. proposed a set of principles for designing databases that enforce a company's privacy policy. Karjoth et al. [16, 18] proposed a privacy-centric access control language (E-P3P and its successor EPAL), and designed an architecture for privacy policy enforcement in the entire life cycle of customers' information, based on the principle of separation of duty. Because of the dynamic nature of enterprise authorization, Karjoth et al. [17] also investigated the translation from enterprise authorization policies to P3P policies. Such a translation will help enterprises keep their privacy promises consistent with their privacy practices. Although a

formal model is designed for EPAL in [18], it is focused on the information flow of an enterprise's internal operation. The semantics proposed in this paper concentrate on private information collection during online transactions. Therefore, some key concepts in EPAL such as action hierarchies and user hierarchies are not applicable in our model.

5. CONCLUSION

A formal semantics for P3P is essential for enabling consistent presentation of P3P policies, designing a semantics-based preference language that avoids the pitfalls of syntax-based preference languages such as APPEL, verifying that P3P policies are adequately enforced by access control policies, checking the internal consistency of P3P policies, and comparing P3P policies of different enterprises to determine whether data flow violate privacy policies or not.

In this paper, we develop a data-centric relational semantics, in which a P3P policy is modelled as a relational database. This semantics is both simple and intuitive. In the process of creating the semantics, we have identified various ambiguities and semantic problems in P3P.

This paper represents our first attempt at a formal study of online privacy policies. As discussed, many challenging issues remain to be addressed. We plan to extend our semantics-centered approach to areas including textual privacy policy modeling, privacy negotiation, privacy policy enforcement and privacy practice auditing. We plan to further validate our semantics by expressing actual textual privacy policies using our semantics, and to investigate the relationships between natural language policies, P3P policies, policies expressed in our semantics, and EPAL policies. We are currently building a prototype system for privacy policy design and analysis, which will serve as a testbed for future research activities.

6. REFERENCES

[1] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Hippocratic databases. In *Proceedings of the 24th International Conference on Very Large Databases*. ACM Press, August 2002.

[2] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Implementing P3P using database technology. In *Proceedings of the 19th International Conference on Data Engineering*, March 2003.

[3] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. An XPath-based preference language for P3P. In *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)*, pages 629–639. ACM Press, May 2003.

[4] Annie I. Antón, Qingfeng He, and David Baumer. The Complexity Underlying JetBlue's Privacy Policy Violations. *IEEE Security and Privacy*, 2004.

[5] AT&T Privacy Bird. http://privacybird.com.

[6] JRC P3P Resource Centre. http://p3p.jrc.it.

[7] Lorrie Cranor. P3P user agent guidlines, May 2003. P3P User Agent Task Force Report 23.

[8] Lorrie Faith Cranor. Personal communication.

[9] Lorrie Faith Cranor. *Web Privacy with P3P*. O'Reilly, 2002.

[10] Lorrie Faith Cranor and Joel R. Reidenberg. Can user agents acurately represent privacy notices?, August 2002. Discussion draft 1.0.

[11] Massimo Marchiori et al. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, April 2002. W3C Recommendation.

[12] UCLA Center for Communication Policy. The UCLA Internet report: Year three. Available at http://ccp.ucla.edu/pages/internet-report.asp.

[13] Giles Hogben. A technical analysis of problems with P3P v1.0 and possible solutions, November 2002. Position paper for W3C Workshop on the Future of P3P. Available at http://www.w3.org/2002/p3p-ws/pp/jrc.html.

[14] Giles Hogben. Suggestions for long term changes to P3P, June 2003. Position paper for W3C Workshop on the Long Term Future of P3P. Available at http://www.w3.org/2003/p3p-ws/pp/jrc.pdf.

[15] Giles Hogben, Tom Jackson, and Marc Wilikens. A fully compliant research implementation of the P3P standard for privacy protection: Experiences and recommendations. In *Proceedings of the 7th European Symposium on Research in Computer Security (ESORICS 2002)*, volume 2502 of *LNCS*, pages 104–125. Springer, October 2002.

[16] Gunter Karjoth and Matthias Schunter. A privacy policy model for enterprises. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW-15 2002)*, pages 271–281. IEEE Computer Society Press, June 2002.

[17] Gunter Karjoth, Matthias Schunter, and Els Van Herreweghe. Translating privacy practices into privacy promises – how to promise what you can keep. In *Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003*, pages 135–146. IEEE Computer Society Press, June 2003.

[18] Gunter Karjoth, Matthias Schunter, and Michael Waidner. Platform for enterprise privacy practices: Privacy-enabled management of customer data. In *Proceedings of the Second International Workshop on Privacy Enhancing Technologies (PET 2002)*, number 2482 in LNCS, pages 69–84. Springer, 2003.

[19] Marc Langheinrich. A P3P Preference Exchange Language 1.0 (APPEL1.0). W3C Working Draft, April 2002.

[20] Ninghui Li, Ting Yu, and Annie I. Antón. A semantics-based approach to privacy languages. Technical Report TR 2003-28, CERIAS, November 2003.

[21] Matthias Schunter, Els Van Herreweghen, and Michael Waidner. Expressive privacy promises — how to improve the platform for privacy preferences (P3P). Position paper for W3C Workshop on the Future of P3P. Available at http://www.w3.org/2002/p3p-ws/pp/ibm-zuerich.pdf.

[22] Daniel M. Schutzer. Citigroup P3P position paper.

Position paper for W3C Workshop on the Future of P3P. Available at http://www.w3.org/2002/p3p-ws/pp/ibm-zuerich.pdf.

[23] W3C. Platform for privacy preferences (P3P) project. http://www.w3.org/P3P/.

[24] Rigo Wenning. Minutes of the P3P 2.0 workshop, July 2003. Available at http://www.w3.org/2003/p3p-ws/minutes.html.