Enhancing Web Privacy Protection through Declarative Policies *

Pranam Kolari, Li Ding, Shashidhara G, Anupam Joshi, Tim Finin
University of Maryland Baltimore County
Baltimore, MD
kolari1@cs.umbc.edu

Lalana Kagal
MIT CSAIL
Boston, MA
lkagal1@cs.umbc.edu

Abstract

The Platform for Privacy Preferences (P3P) is a W3C framework for web privacy management. It provides a standard vocabulary that websites can use to describe their privacy practices. The presence of website published P3P policies enable users to configure web browsers to allow, block or warn users during access and data exchange with websites. It's a good idea that unfortunately is rarely used. We identify three primary reasons: (i) the languages available to describe user privacy preferences are not sufficiently expressive, (ii) P3P policies published by websites are not trusted by users and (iii) P3P framework does not provide a coherent view of available privacy protection mechanisms to the user. Towards addressing these issues; we present enhancements to the P3P framework. We use a more expressive policy language based on deontic concepts to describe users privacy-related policies, constraints and preferences. We introduce a new trust model for websites and describe its use in user privacy preferences. Finally, we present sample policies to demonstrate the relevance of our work and offer it as an effective starting point towards enhancing Web Privacy Protection.

1. Introduction

The importance of *Web Privacy* protection has increased with the growth of the Web. While accessing websites such as online stores and news portals, users often need to provide personal information (e.g. name, email). Moreover their browsing activities (e.g. click-stream, mouse movements, browsing trail) is usually recorded for analysis. Users web privacy is not very secure today due to a combination of increased tracking[1], personal information disclosure and information harvesting from public sources by spammers. At the extreme, the possibility of privacy related fraud remains. Distributed data mining [17] may even track a user across websites, user sessions and physical locations. Hence it is important for users to be aware of these potential problems and be able to protect their web privacy.

In order to protect web privacy, a straightforward approach is to evaluate privacy policies by reading a website's privacy statement. Such manual efforts are fairly reliable but impractical. Even if a site has a privacy policy few people can invest the time to read a lengthy privacy policy before browsing a website. This makes a case for automated tools.

Many existing web privacy enforcement mechanisms [14] like cookie busters, anonymous surfing enablers, website filters, and pop-up killers follow simple strategies. Each of them can more or less protect certain aspects of web privacy; however, they are based on user heuristics, are independent of a websites' privacy policy and fulfill user preferences only partially.

A complementary approach is W3C's P3P framework [15] for automating the privacy policy verification process. P3P requires websites to publish XML based privacy policy, allows users to specify their preference and enforces privacy protection through a user agent (usually built into web browsers). This framework is a good starting point; however, it is not widely adopted by websites. Cranor et al [7] report that only 538 of the top 5856 websites were P3P enabled (published valid P3P policies) till May 2003. In another report, P3P DashBoard [12] from Ernst & Young shows a very low increase in P3P adoption for the top 500 sites, from 16% (August 2002) to 23% (January 2004). Therefore, user agents seldom encounter websites with published P3P policy. This situation, together with P3P's limitation on the user side, has resulted in low P3P adoption from users.

In order to promote the P3P framework and make existing web privacy protection mechanisms more useable, we propose a backward-compatible enhancement to P3P framework which provides better support for users. The in-

* This work was supported in part by NSF grants IIS0242403 and IIS0325172, and DAML program under DARPA contract F30602-00-2-0591

1 see also http://www.cdt.org/privacy/guide/start/track.html

tuition is that better user side support will promote user adoption of P3P, and thus affect websites adoption of P3P. Our focus is on two aspects:

1. **Effective user preference language.** We use a highly expressive policy language Rei [16], since it not only enables users to specify their privacy preferences, but also enables integration of existing web privacy enforcement mechanisms.
2. **Extensible trust model.** We enhance the P3P's trust model by augmenting it with trust building mechanisms based on social-recommendations. A website evaluation ontology is proposed for users to exchange and assert their opinions about websites, which is used in specification of privacy preference policies.

The rest of the paper is organized as follows: section 2 provides a critical review of the current web privacy enforcement mechanisms, section 3 lists what needs to be enhanced and our choices, section 4 details Rei and its applicability for user preference specification, section 5 details a web evaluation ontology and our trust model, section 6 explains our enhancement using simple policies, section 7 briefly describes the prototype system, and sections 8 and 9 concludes our work with future directions.

2. P3P: A Critical Review

The W3C based P3P web privacy framework comprises of the following:

- **Website Privacy Policy.** Websites are required to publish their privacy policy in XML using the P3P policy vocabulary and store policy(policy reference) files in standard locations to facilitate user access.
- **User Privacy Preference Policy.** Users can specify their privacy preferences in terms of a recommended policy language (i.e. APPEL[5]) which is the counterpart of website's P3P Policy.
- **User Agent.** Before accessing a website (e.g. fetching page, exchanging cookie), a P3P user agent (which is often inbuilt into a web browser) will automatically retrieve the website's P3P policy and compare it with users' privacy policy, to verify whether the P3P policy conforms to user privacy preferences. A match decides the action to be taken, which is enforced by the web browser.

The rest of this section will review the strength, relevance and limitations of P3P's user side support.

2.1. User Privacy Preference Language

APPEL (A Privacy Preference Exchange Language) is recommended by the W3C for expressing user privacy preferences. Users can describe their requirements for matching with P3P policies with a *RULE*, which specifies one or multiple matching requirements combined using logical operators (i.e. and, or, non-and, non-or, and-exact, or-exact). User's policy is a *RULESET*, which consists of a sequence of *RULE*s ordered by their priority in execution. An APPEL user agent evaluates rules in the order of priority until a match is found and ignores subsequent rules. Figure 1 shows a simple example of matching a P3P policy with an APPEL *RULE*, and the matched elements are connected by arrows.

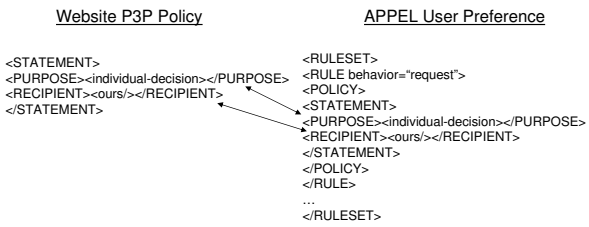

Figure 1. P3P-APPEL matching

```
<RULESET>
  <RULE behavior="request"
     condition="/POLICY[
          every $pname in STATEMENT/PURPOSE/*
          satisfies name($panme)="individual-decision"
                and
          every $rname in STATEMENT/RECIPIENT/*
          satisfies name($rname)= "ours"
                              ]"/>
  <RULE behavior="block" condition="true"/>
</RULESET>
```

Figure 2. P3P-XPref matching

The limitations with APPEL is detailed by Agrawal et al [2], namely its notion of logical connectives, rule ordering and matching criteria. Due to these reasons APPEL can specify only "what is unacceptable" but not "what is acceptable" for a user. They propose XPref, an XPath based language for user preferences, to resolve the above limitations. XPref uses logical connectives (and, or) and equivalence operators(=, !=) to specify matching requirements. A matching example for the same P3P policy as in Fig 1 using XPref is given in Fig. 2.

The limitation with XPref is its restrictive expressiveness (the same as APPEL). Specifically, it does not support obligations or dynamic policy specification and negotiation through delegation management, nor does it allow reasoning over user context in general. For instance, how would an APPEL/XPref based user preference specify obligation from websites (e-mail notification on changes to website policies), or obligation from web browsers (delete persis-

tent cookies after a current session).

These capabilities in existing policy languages are requirements of current and future web privacy enforcement frameworks, but are not supported by APPEL and XPref.

2.2. Trust Model

Maintaining and building customer trust is an important criterion for the growth of online business. A recent survey [9] by Ernst and Young suggests that 56% of online consumers believe that websites do not adhere to their privacy policies. Websites have resorted to different mechanisms to build and maintain this trust, like customer service, better handling of user data, text privacy policy, certification etc.

The P3P framework adopts a certificate based trust model. A P3P policy can establish its trust by specifying its certifier, which is a trusted authority for accountability of P3P policy such as TRUSTe.com. We argue that this model does not incorporate trust sufficiently. First, it is highly coupled to the presence of a certifier, whose adoption is low among websites. Second, in the absence of a privacy certifier the model makes a strong assumption that the presence of P3P policies is sufficient for building trust. In fact, other factors such as website popularity and prior experience of the user, which may lead to trust in a website are not sufficiently considered. These factors are neither modeled nor used by the existing framework.

2.3. User Agent

The well known P3P user agents (APPEL based) are Privacy Bird by AT&T (http://privacybird.com) and P3P Proxy by JRC (http://p3p.jrc.it). However, these P3P user agents are not the only privacy protection mechanisms for the web.

Cookie Cutters and Anonymyzing Proxies are two other popular independent privacy protection mechanisms. Cookie cutters manage cookie handling heuristic by selectively deleting cookies when a HTTP request-response is sent and received from a web server. Anonymyzing proxies tunnel all requests through a tunneling service and primarily protects users from IP address detection.

Popular web browsers have inbuilt user agents which implement a simplification of P3P. Though Mozilla does not formalize representation of user privacy requirements, Internet Explorer provides its own language [6] for user privacy specification. These browsers are limited mainly to cookie handling heuristics. Matching of user privacy specification to published P3P is restricted to compact policies (a stream of policy tokens from the P3P vocabulary) which is an over simplification.

These privacy protection mechanisms, though using simple strategies, do exhibit good performance in a certain perspective. Their existence not only reveals the difference between P3P and "what the users really wants", but also shows a promising direction of incorporating multiple privacy protection mechanisms in P3P using policy languages.

3. Enhancing P3P

Based on our analysis on the current P3P framework and existing privacy protection mechanisms, we present two key enhancements which will drive user adoption:

- **Enhancing P3P Privacy Preference Language.** An expressive policy language is preferred with at least the following attributes: i) well understood matching semantics, ii) sufficiently expressive to encode a wide range of users' preferences (e.g. obligation) over any domain specific information model, and iii) extensible to constrain the behavior of available privacy enforcement mechanisms providing better privacy protection usability.
- **Enhancing P3P Trust Model.** Beside the certificate-based trust model, user should have more choices to establish trust in websites.

In this paper, we approach both enhancements by building on our previous work in policy language [16] and trust models [11].

We approach the first enhancement by adopting the Rei policy language. Rei has particular applicability due to its extensible structure, first order logic semantics and Semantic Web [3] background (i.e. grounding in RDF and OWL). Rei is based on deontic concepts with speech act modeling capabilities. Deontic concepts based policies allow modeling of obligations in addition to permissions and prohibitions. Obligations allow the incorporation of other independent privacy enforcement mechanisms. Speech acts allow for dynamic policy management. Rei's Semantic Web background make it easy to deal with P3P policies published in P3P-RDFS[18], which is backward-compatible with the XML P3P policy. It also allows reasoning over the user context, in general.

We approach the second enhancement by adding mechanisms that allows users to share knowledge and opinions about websites. We propose a website evaluation ontology for expressing user's knowledge and opinions about websites. With such an ontology, users may build trust in website with past experience and others' evaluation about websites. The success of Epinions (http://www.epinions.com) and Bizrate (http://www.bizrate.com) partially prove the utility of such mechanisms even when they do not come with any guarantees. We also show how such evaluation mechanisms can be incorporated into web privacy protection through user privacy preferences expressed in Rei. This

in turn decouples the strong dependence between P3P and user privacy preference inherent in current frameworks.

Note that there are, of-course, other possible candidates such as XACML[1], KAoS[23], Ponder[8]. Our enhancements(model) will work with any of them, but we choose Rei for implementation.

4. User Privacy Preference Specification

In this section we briefly describe Rei and detail its usage for user privacy preference specification. We also compare it with APPEL and Xpref to how its advantages. For further details on Rei, we refer the reader to [16].

4.1. Rei Policy Language

Rei is a declarative policy language with an RDF/XML grounding (recent versions support OWL, OWL-Lite), which includes notions of logic like variables for describing different kinds of conditions. It is modeled on deontic concepts of permissions, prohibitions, obligations and dispensations and additionally supports delegation management for dynamic policy specifications. We identify the key features of Rei which makes it particularly useful in the Web privacy domain as:

- **Policies over Domain Specific Ontologies.** Rei policies are specified over instances of ontologies that model users' privacy preferences and relevant concepts in the domain. Based on the user, this could mean not only website evaluation statements, but also user context in general. The policy language is grounded in domain independent ontologies but can also reason over specific domain dependent ontologies. The ontology-based approach provides rich semantics for specification of highly expressive policies and also provides ease of extensibility.
- **Based on Deontic Concepts.** Rei is based on deontic concepts of permissions, prohibitions, obligations and dispensations. Permissions and Prohibitions are commonly used to constrain entity behavior in terms of what is allowed and not allowed. Obligations are promises made by any of the entity in the domain. For example, in the web privacy domain it could include obligation from a website (e.g. e-mail notification on website privacy policy updates) or from a web browser (e.g. delete cookies after current session, tunnel requests through anonymizying proxies)
- **Delegation Management.** Rei provides speech act capabilities (request, revoke, delegate) for dynamic policies. Such policies are of particular interest when the user delegates rights of data sharing to website's. Since this requires changes at the web server (web site) and the way they deal with user information, we do not detail them in the context of this work, but point that this is an important capability for future frameworks.
- **Matching Semantics.** Matching semantics decide how constraints of rules are to be matched and decide prioritization between multiple rules. Rei is based on first order logic and provides the notion of logical connectives (and, or, not) for grouping constraints in policies. It also provides metapolicies for dealing with policy and rule conflicts. Such metapolicies can be used for reasoning over multiple rule matches and to prioritize over matched rules.

Associated with Rei is a rule engine that interprets and reasons over the policies, related speech acts and domain information expressed in RDF to make decisions about applicable rights, prohibitions, obligations and dispensations.

4.2. Rei for User Privacy Preference Specification

Rei has specific *domain independent ontologies* that mandate the policy syntax which can be represented in RDF-S/OWL[2]. Figure 3 shows a subset of classes in the Rei ontology that is sufficient for specifying user policies in the current enhancements. It uses the following notations: classes(starts with capitalized letter)are depicted by oval nodes; properties(starts with non-capitalized letter) are depicted by directed edges from the domain to range of the property; dashed edges associate the class *Constraint* to different possible types of specifiable rule constraints; all entities with a white background have a counterpart in APPEL; and shaded entities are additional capabilities provided by Rei. In what follows we describe Rei concepts by identifying equivalent constructs in APPEL and identifying their use in specifying user privacy policies. They are also summarized in table 1.

4.2.1. Policy. An instance of *Policy* is a counterpart to the APPEL *RULELIST* element and is used to describe a user's privacy preference.

4.2.2. Granting. An instance of *Granting* is the counterpart to the APPEL *RULE* and represents an individual rule of a user's privacy preference. These rules are specified on particular actors in the domain. We model websites and the web browser(enforcement mechanism, proxy server in our implementation) as actors(entities) of the web privacy protection system. Policies specified in Rei constrain actions applicable for these actors.

2 (http://www.cs.umbc.edu/~lkagal1/rei/ontologies)

	APPEL	Rei	Xpref
RootElement	RULELIST	Policy	RULELIST
RuleElement	RULE	Granting	RULE
Actions	request,block, limited	Abitrary domain action	request,block, limited
User context modeling	–	OWL/RDF	–
Constraints	P3P Specific	Domain ontology	P3P Specific
Rule Priority(1)	Serialized Ordering	RulePriority for partial ordering	Serialized Ordering
Rule Priority(2)	–	Positive over negative, vice versa	–
Delegation Management	–	Speech Acts	–

Table 1. A language feature comparison of APPEL, Rei and XPref

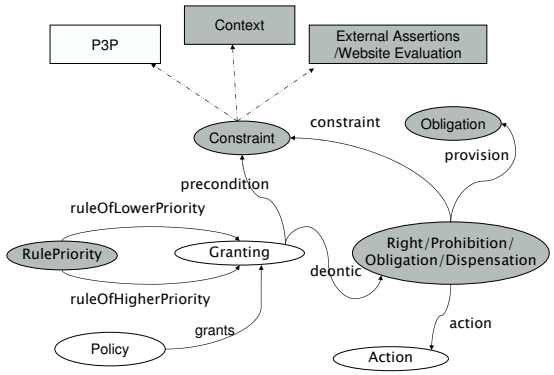

Figure 3. Rei concepts used in Web Privacy

4.2.3. Action Each policy rule is specified on certain actions using the *deontic* property. *Action* can be associated with one of *Permission, Prohibition, Obligation or Dispensation* and is defined based on domain specific ontologies. For example, *Actions* as they relate to Web privacy can be categorized into *request(allow), block, limited, request-prompt, limited-prompt* which are *Permission*'s and *block* which is a Prohibition. These *Actions* are applicable to the website and provides a direct mapping to actions in APPEL. Rei allows other actions to be specified as well, based on the capabilities of the enforcement mechanism (proxy server, web browser etc.) . We will detail *Obligation* management later in this section.

4.2.4. precondition, constraint. are properties used in Rei policies and associate *RULE* and *Granting* with instances of the class *Constraint*. *precondition* for a rule (e.g. web resource being accessed is not an activeX control) allows filtering of rules before checking for other constraints.

4.2.5. Constraint. Rei can specify a wide range of constraints through the inclusion of domain specific ontologies. A constraint is of the form "websiteX hasCertifier Trust-E" or " userX isBrowsingFrom Home" or more generally "x has an attribute y with value z". As shown in figure 3, APPEL and Xpref let a user specify constraints only on P3P policies published by websites. Rei allows constraints on other domain specific knowledge like context information (e.g. IP address of the client) from a user context ontology and privacy related statements (e.g. popularity of the site) from the website evaluation ontology. A policy engineer can specify all domain specific ontologies of interest to the user and specify policies over them.

4.2.6. Logical Connectives. Logical connectives enable the specification of complex constraints. *Constraints* can be either simple conditions or complex constraints using a combination of simple conditions. Complex constraints can be one of *AndConstraint*, *OrConstaint* and *NotConstraint*.

4.2.7. RulePriority. Two properties *ruleOfHigherPriority* and *ruleOfLowerPriority* associate the instance of *RulePriority* to instances of *Granting*. To specify priorities between more than two instances of *Granting*, multiple instances of *RulePriority* can be used and cascaded. Rei also provides *Action* preference i.e positive over negative or vice versa which can be used to specify that permissions(e.g. allow access) have priority over prohibitions(e.g. block access) or vice versa when there are conflicting actions.

4.2.8. Obligation Management. Obligation management requires particular mention, since it corresponds to an important capability of the current state of the art in policy languages and Rei in particular. Obligations can be looked at from two perspectives:

- **Obligation.** Rei can be used to specify obligations which are true at all times for actors in the domain of interest. Obligations on the web browser could include specifying that no cookies are to be stored be-

yond a specific period of time. Additionally the web browser could be obliged to tunnel all requests through a anonymizing proxy service on certain constraints being true. Obligations on websites can also be specified, but their enforcement cannot be controlled by the user and is not in the scope of our current privacy framework. However, its importance cannot be ignored given that it is an important promise made by website privacy policies. For instance an excerpt from the privacy policy of www.yahoo.com reads – "We transfer information about you if Yahoo! is acquired by or merged with another company. In this event, Yahoo! will notify you before information about you is transferred and becomes subject to a different privacy policy." Such obligations can be attached to a website when information is shared with them using Rei.

- **provision.** Rei provides the ability to specify *provision* which can be used to specify obligations on the completion of a particular *Action*. The domain of *provision* is a *Permission* and its range is *Obligation*. This is different from the *Obligation*s directly attached to actors through *Granting*, in that these obligations are dependent on certain actions being fired. An example is the obligation on the web browser (enforcement mechanism) that cookies should be deleted after the current session for a particular website. Note that in this example the actor on whom the obligation is triggered is the web browser, and the permission is given to a website. So multiple actors are involved in a particular *RULE*. Using provisions the cookie handling heuristics as seen in Internet Explorer can be modeled using Rei and enforced using a capable enforcement mechanism. [3]

An important capability of Rei policies as seen above is the ability of specifying policies over the automatic use of independent enforcement mechanisms. Such modeling provides a more coherent view of privacy protection mechanisms to the user and improves their usability.

4.3. Queries for Enforcement

We make use of the existing query interface of Rei to identify the right deontic object in effect for actors(website and web browser). Prior to allowing access to a website(following fetching P3P policy if available), the policy engine is queried to identify the right action for the actors, and is enforced using an enforcement mechanism.

3 Rei currently allows modeling of such policies, but the engine does not support provision related queries on monitored actions.

Domain specific ontologies over which policies are specified like user context, website, appel-actions etc. are available at http://semdis.umbc.edu/privacy/.

5. Enhancements to the P3P Trust Model

Building consumer trust is important to any online vendor, especially to maintain customer base. The current framework captures trust and makes it explicit in the form of (i) text based privacy policy (ii) XML based policy through P3P and, (iii) certification of the above by an independent third party like Trust-E or BBBOnline. Ernst and Young reports[9] that 90% of online consumers believe that independent verification is a sufficient measure for trusting a website. However this model is highly tied to adoption by websites, in the absence of which users are left with little choice but to make an educated guess. This factor, combined with little or no law enforcement support for lacking enforcements from websites, results in very low consumer trust to online vendors.

We address this limitation by considering other factors that can be used to decide trust with websites, in addition to privacy certifiers. These factors are not tied to P3P, and provide an objective view of the website as it relates to privacy practices and services offered. For example, Google Pagerank [19] is based on the number of in-links to a particular website. Such information has earlier been suggested as an implicit trust certifier [20] for websites. We term these factors as website evaluation statements or website evaluations. In what follows, we detail our approach of enhancing the P3P trust model.

5.1. Gathering and sharing website evaluations

The collection of data can be in either of the following forms:
(i) **user level heuristics as decentralized sources** - can be captured through past user experiences with websites either implicitly or explicitly. Implicit collection is based on mining user preferences and browsing behavior on the client side. Explicit collection include users specifying a set of trusted websites. Simple mechanisms can capture such information, based on available data already stored at clients. Implicit trust with websites can be captured through the browser's "history" and "bookmarks" feature. Explicit trust is usually specified by users when they accept signed certificates, allow cookies from specific sites or allow execution of java based applets and active-X controls. The user might decide to share some of this information in a trusted social network[4] in re-

4 http://www.stumbleupon.com

turn for certain benefits.
(ii) **centralized sources** - act as reputation servers and capture user trust with websites. Popular websites include http://www.google.com and http://www.bizrate.com. While Google collects PageRanking information, BizRate provides website rating facilities for consumers which is used to generate overall rating for websites. Though Google provides web service interfaces for their services, BizRate does not provide this facility. To test the feasibility of scraping website content providing ratings, we have implemented a web-scraper for BizRate and exposed it as a web service. Other centralized authorities include http://www.trustwatch.com and http://www.trustguage.com which provide evaluations of websites for consumer trust as browser toolbars, showing the usefulness of this approach.

Website evaluation statements to be used by users can be obtained either from centralized sources, or from trusted social networks, based on the user's trust in these mechanisms. Social(trusted) Recommender Systems enable knowledge sharing across multiple sources in a social network. Obtaining statements from centralized sources is straightforward if their services are exposed as a web services. However using a trusted recommender system requires extensive analysis of stable mechanisms and systems. We have collected FOAF[5] data [10] to analyze the use of such systems for real world social networks and have performed experiments on knowledge outsourcing for synthesized data[11]. For the purposes of this paper and our framework, we are concerned with only using web evaluation statements and consider mechanisms to generate such information as a black-box.

5.2. Website Evaluation Ontology

The final aim of the framework is to have trusted website evaluations available, for their use in policies, to decide how a user interacts with a website. We model captured data using an ontology to leverage reasoning capabilities using such models. We introduce the rationale behind some key entities of our current model[6] in the scope of validating their use in privacy preferences.

- domainSuffix - is the suffix of a website's domain name, such as ".com", ".gov", and ".edu". For example an educational website with domainSuffix .edu, would rarely set cookies and use them in ways that might breach user privacy.
- popularity - refers to the number of users who review the website with rating (e.g Google PageRank). Intuitively, it shows the confidence about the reputation of a website.

5 www.foaf-project.org
6 http://semdis.umbc.edu/privacy/ontologies/Website.rdfs

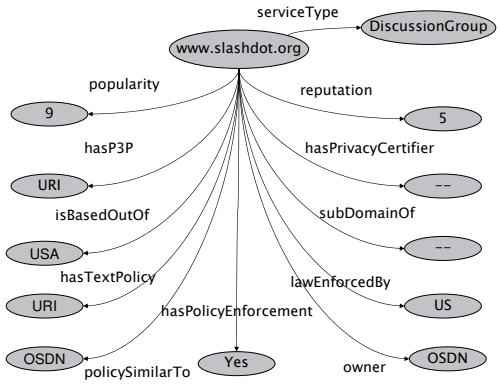

Figure 4. Typical instance of Web Evaluation Ontology

- lawAccountability - lawAccountability gives information about the privacy laws which are in effect for a particular website. This could be because of a particular website being hosted at a particular country or state. A user might be confident of privacy laws in U.S and not very certain about the one's in Asia.
- domainOfService - represents website classification based on the type of service provided like informational site, advertising site, search site etc. The kind of service a website offers can give valuable information about data that has to be protected and released. For example, a zip-code is sufficient information for a search engine to provide localized services, whereas street address is of unnecessary granularity.

An instance of the website evaluation ontology for the website http://www.slashdot.org is shown in Figure 4.

5.3. Incorporating website evaluation statements into Privacy decisions

To specify policies over website evaluation statements, we leverage the fact the Rei can be used to specify policies over any domain specific ontology. Website evaluation statements are used as *Constraints* for *Actions*, to decide if access to a website is to be allowed or not, and if so what obligations are in effect. A simple example, would be that a user specifies that access to all websites with domain suffix ".edu" or for that matter for websites with domainOfService of "banking" is to be allowed unconditionally. A point of significance here is that such policies imply that user preferences are no more dependent on P3P policies published by websites, thus decoupling the dependence of user side protection with websites.

```
<policy:Policy rdf:about="&wwwpolicy;comprehensive“ policy:desc="Sample  policy">
    <policy:grants rdf:resource="&wwwpolicy;grantingPermission" />
            ..
</policy:Policy>
<!– Granting Objects -->
<policy:Granting rdf:about="&wwwpolicy;grantingPermission">
    <policy:desc>Current policy allows access to a website</policy:desc>
    <policy:to rdf:resource="&wwwpolicy;var1"/>
    <policy:deontic rdf:resource="&wwwpolicy;right1"/>
</policy:Granting>
…
<!– Deontic Objects -->
<deontic:Permission rdf:about="&wwwpolicy;right1">
   <deontic:actor rdf:resource="&wwwpolicy;var1"/>
   <deontic:action rdf:resource="&wwwpolicy;request"/>
   <deontic:constraint rdf:resource="&wwwpolicy;complexconstraint" />
    …
</deontic:Permission>
```

Figure 5. A Simple Policy Template

6. Sample policies enabled by enhancements

The examples that follow depict various user policy requirements and their specification using Rei to highlight some key features. Our implementation enforces the deontic concepts of permission and prohibition. We leave the enforcement of more powerful features to web browser implementations based on our framework.

We represent only sections of the user policies which are relevant in the depictions that follow and use namespaces without declaring them, for conciseness. Note also that it is assumed that tools will enable specifying such policies. RDF/XML representation of the complete policies and other complex policy examples are available online.[7]

All policies are written over two actors - the website being accessed and the web-browser.

- **Simple policy.** Figure 5 represents a generic policy template. The aim of this policy example is to show a partial snapshot of how the various policy constructs provided by Rei can be used to specify policies. *Policy* element consists of multiple rules. Metapolices can also be specified with the *Policy* element, which will be used in subsequent examples. Each Rule is on a particular *Deontic* Object, as linked through the *Granting* element. *Constraints* similar to the specified *constraint1* in this example is enumerated in subsequent examples. More information on the syntax and semantics of Rei is available online.[8]
- **Obligation management.** Obligation specification is one of the key language features which provides a comprehensive view of enforcement mechanisms. Figure 6 shows how obligations can be used to specify conditional access. The deontic action - *Permission* gets resolved to a website during policy evaluation. The obligation on part of the web browser is fired for this website. This obligation requires that the web browser use an anonymizing proxy to tunnel HTTP requests to the websites. Note the value of such mechanism to protect privacy of large enterprises against web usage mining based competitor analysis. Other kinds of obligations can be specified using the *provision* attribute of actions and linked to *Actions*. Constraints are used to specify when such rules fire. Some examples of specifying constraints follow.

```
<policy:Policy rdf:about="&wwwpolicy;obligationexample"
            <policy:grants rdf:resource="&wwwpolicy;grantingRight" />
            <policy:grants rdf:resource="&wwwpolicy;grantingObligation" />
            …
</policy:Policy>
<policy:Granting rdf:about="&wwwpolicy;grantingRight">
            <policy:to rdf:resource="&wwwpolicy;var1"/>
            <policy:deontic rdf:resource="&wwwpolicy;right1"/>
            …
</policy:Granting>
<policy:Granting rdf:about="&wwwpolicy;grantingObligation">
            <policy:to rdf:resource="&wwwpolicy;webbrowser"/>
            <policy:deontic rdf:resource="&wwwpolicy;obligation1"/>
            ..
</policy:Granting>
<deontic:Permission rdf:about="&wwwpolicy;right1">
      <deontic:actor rdf:resource="&wwwpolicy;var1"/>
      <deontic:action rdf:resource="&wwwpolicy;request"/>
       …
</deontic:Permission>

<deontic:Obligation rdf:about="&wwwpolicy;obligation1">
      <deontic:actor rdf:resource="&wwwpolicy;webbrowser"/>
      <deontic:action rdf:resource="&wwwpolicy;tunnelRequest"/>
</deontic:Obligation>
…
```

Figure 6. Obligation on Web-browser

- **Use of metapolicies.** Metapolicies are used to explicitly specify rule ordering and priorities over deontic objects. Figure 7 shows the use of *RulePriority* as well as *defaultModality* to specify priorities on rules and over deontic objects. Modality decides which deontic object has higher priority (prohibition in this example). Further the modality meta-policy has higher precedence over RulePriority. The semantics of such rules imply that if all prohibitions fail, different kinds of rights(differing actions) based on *RulePriority* are checked to find the best match.
- **Trust Model.** Figure 8 depicts a particular constraint based on an instance of the website evaluation ontology. This is also useful in decoupling the strong dependence on websites publishing P3P. If websites publish P3P their validity can be additionally inferred using website evaluation statements. In the absence of published P3P only such statements can be used to make privacy decisions. The example also shows the use of complex constraints using *OrConstraint*.
- **User Context.** Figure 9 shows how constraints on pol-

7 http://semdis.umbc.edu/privacy

8 http://www.cs.umbc.edu/ lkagal1/rei

```
<policy:Policy rdf:about="&wwwpolicy;ruleprioritыexample">
   <policy:defaultModality rdf:resource="&metapolicy;NegativeModalityPrecedence/>
   <policy:metaDefault rdf:resource="&metapolicy;CheckModalityPrecFirst" />
   <policy:grants rdf:resource="&wwwpolicy;grantingRight1" />
   <policy:grants rdf:resource="&wwwpolicy;grantingRight2" />
   <policy:grants rdf:resource="&wwwpolicy;grantingProhibition" />
   <metapolicy:rulePriority rdf:resource="&wwwpolicy;rulepriority1"/>
    …
</policy:Policy>
   …
<metapolicy:RulePriority rdf:about="&wwwpolicy;rulepriority1">
   <metapolicy:ruleOfGreaterPriority rdf:resource="&wwwpolicy;grantingRight1" />
   <metapolicy:ruleOfGreaterPriority rdf:resource="&wwwpolicy;grantingRight2" />
</metapolicy:RulePriority>
```

Figure 7. Use of Metapolicies

```
<constraint:SimpleConstraint rdf:about ="&wwwpolicy;domainOfServiceconstraint"
                constraint:subject="&wwwpolicy;var1"
                constraint:predicate="&weo;domainOfService"
                constraint:object="&weo;travel" />
…
<constraint:SimpleConstraint rdf:about ="&wwwpolicy;trustedDomainEDUconstraint"
                constraint:subject="&wwwpolicy;var1"
                constraint:predicate="&weo;domainSuffix"
                constraint:object="&weo;edu" />
..
<constraint:Or rdf:about ="&wwwpolicy;trustedDomainSuffixServiceconstraint">
      <constraint:first rdf:resource= "&wwwpolicy;trustedDomainGOVconstraint" />
      <constraint:second rdf:resource= "&wwwpolicy;domainOfServiceConstraint" />
</constraint:Or>
..
```

Figure 8. Constraints on Website Evaluation

icy actions can be specified using the user context. The user context is modeled using a specific ontology and instances of this ontology are used for specifying preferences. The constraint is used to specify the fact that the user is browsing websites when away from home i.e. when traveling. Most users would ideally prefer to protect against IP detection and use while traveling. This example shows how Rei can be used to specify policies against any domain specific ontology of interest to the user.

7. System Design and Implementation

Figure 10 overviews the framework with four key components, namely, the client using the web privacy protection framework, the website being accessed, the website recommender network and other middleware (intelligent privacy proxy, Rei Engine, XSLT transformer and the Privacy Expert) which evaluate and enforce user privacy preferences.

```
<constraint:SimpleConstraint rdf:about="&wwwpolicy;awayconstraint">
      <constraint:subject rdf:resource='&wwwpolicy;userContext'/>
      <constraint:predicate rdf:resource='&context;browsingFrom'/>
      <constraint:object>&context;away</constraint:object>
</constraint:SimpleConstraint>
```

Figure 9. Constraints on User Context

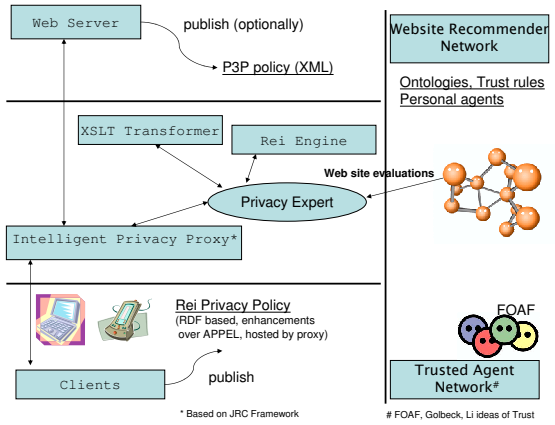

Figure 10. The enhanced P3P Web privacy framework

Note the enhancements to the system from traditional P3P framework.

- **JRC Privacy Proxy.** The JRC Proxy was one of the first implementations of a P3P user agent. Users register with the proxy by publishing their user preferences in APPEL. The proxy fetches the P3P XML policy of a website and matches it against user specified APPEL policy using an APPEL evaluator. This enables the proxy to enforce the user's preferences on all http requests. We make minor changes to the proxy for our proof of concept implementation. In our system the privacy proxy acts as an enforcement mechanism (allow or block access) and delegates the policy evaluation functionality to the Privacy Expert (see section 7). Registered users in our enhanced system are required to publish their user preferences using the Rei policy vocabulary.
- **Website Recommender Network.** We incorporate a new trust model by using website evalution statements provided by the Website Recommender Network. This recommender network is a social network of trusted agents that uses ontologies and rules of trust for knowledge sharing. Trust between agents on this network can be derived by using FOAF or other approaches[13, 21, 11].
- **Privacy Expert.** The Privacy Expert (PE) takes the following inputs, namely, P3P policy, Rei user preference, the website to be accessed and other domain specific instances(ontology based). It uses the XSLT transformer [24] to convert P3P from XML to RDF. It can also query the website recommender network for website evaluation statements and to any context server for user context. The privacy decision is fi-

nally made by the Rei policy engine and enforced by the JRC proxy. Our proof of concept implementation uses a proxy server as an interface(enforcement mechanism) between the privacy expert and the browser. For real world usable systems we recommend the incorporation of the PE and the enforcement mechanism directly into web browsers(the model used by ATT Privacy Bird).

A notable point in the entire framework is that no changes are required from web servers, making our scheme backward compatible, as it were. We recognize that describing privacy policies expressed in Rei is not something that an average user will be able to do. However, there are ongoing efforts by researchers to either learn user preferences from observing their behavior (web mining on the client side), or at least provide graphical interface s and templates for policy specification.

The entire framework including sample files are available for download at http://semdis.umbc.edu/privacy.

8. Conclusion and Future Work

In this paper we have presented enhancements to the P3P framework through the use of a more expressive user preference language and an improved trust model. We believe that these enhancements will be effective in making the web privacy protection mechanisms more useable leading to their widespread adoption. The key contribution of our work is in showing how expressive user preference languages can be effectively used in realizing the enhanced framework. As a follow on to this work, we are exploring the use of Rei's delegation management capabilities for future web privacy protection directions, especially policy negotiation [4]. We will also address the disconnect between user preference languages and enterprise wide privacy enforcement mechanisms like EPAL[22].

References

[1] Extensible access control markup language (xacml) version 2.0.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. An xpath-based preference language for p3p. In *Proceedings of the twelfth international conference on World Wide Web*, pages 629–639. ACM Press, 2003.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 279(5):34–43, May 2001.

[4] P. Bonatti and P. Samarati. Regulating service access and informational release on the web. In *Proceedings of the 7th Conference on Computer and Communications Security*, 2000.

[5] L. Cranor, M. Langheinrich, and M. Marchiori. A p3p preference exchange language 1.0.

[6] L. F. Cranor. *Web Privacy with P3P*. O'Reilly & Associates, 2002.

[7] L. F. Cranor, S. Byers, and D. Kormann. An analysis of p3p deployment on commercial, government, and children's web sites as of may 2003. Technical report, AT &T Labs-Research, 2003.

[8] N. Damianou, N. Dulay, E. Lupu, and M. Sloman. The ponder specification language. In *Workshop on Policies for Distributed Systems and Networks(Policy 2001)*, 2001.

[9] J. R. DeVault, D. Roque, jay Rosenblum, and K. Valente. Privacy promises are not enough. Technical report, Ernst&Young, 2001.

[10] L. Ding, T. Finin, and A. Joshi. Analyzing social networks on the semantic web. *IEEE Intelligent Systems*, 8(6), 2004.

[11] L. Ding, L. Zhou, and T. Finin. Trust based knowledge outsourcing for semantic web agents. In *Proceedings of IEEE/WIC International Conference on Web Intelligence*, 2003.

[12] Ernst&Young. P3p dashboard report: Top 500 p3p dashboard, 2004.

[13] J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic web. In *Proceedings of Cooperative Intelligent Agents*, 2003.

[14] http://www.epic.org/privacy/tools.html. Epic online guide to practical privacy tools.

[15] http://www.w3.org/P3P/. The w3c p3p framework.

[16] L. Kagal, T. Finin, and A. Joshi. A policy based approach to security for the semantic web. In *Proceedings of 2nd International Semantic Web Conference (ISWC2003)*, September 2003.

[17] H. Kargupta and P. Chan, editors. *Advances in Distributed and Parallel Knowledge Discovery*. MIT/AAAI Press, 2000.

[18] B. McBride, R. Wenning, and L. Cranor. An rdf schema for p3p.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bring order to the web. Technical report, Stanford, 1998.

[20] J. W. Palmer, J. P. Bailey, and S. Faraj. The role of intermediaries in the development of trust on the www: The use and prominence of trusted third parties and privacy statements. In *Journal of Computer-Mediated Communication, JCMC 5 (3) March 2000*, 2001.

[21] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *Proceedings of the Second International Semantic Web Conference*, 2003.

[22] M. Schunter and C. Powers. The enterprise privacy authorization language (epal 1.1).

[23] A. Uszok, J. Bradshaw, and R. J. et al. Kaos policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement. In *IEEE 4th International Workshop on Policies for Distributed Systems and Networks(Policy 2003)*, 2003.

[24] R. Wenning. (w3c privacy activity lead) personal communication: Xslt for p3p rdf, march 10, 2004.