RZ 3452 (# 93771) 09/09/02
Computer Science 24 pages

Research Report

Enterprise Privacy Practices vs. Privacy Promises — How to Promise What You Can Keep

Matthias Schunter and Els Van Herreweghen

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
{mts,evh}@zurich.ibm.com

Research
Almaden · Austin · Beijing · Delhi · Haifa · T.J. Watson · Tokyo · Zurich

Enterprise Privacy Practices vs. Privacy Promises — How to Promise What You Can Keep

Matthias Schunter, Els Van Herreweghen
IBM Zurich Research Laboratory, Switzerland
{mts,evh}@zurich.ibm.com

09/09/2002

Abstract

Enterprises can publish privacy promises using the W3C Platform for Privacy Preferences (P3P) and advertise their compliance with certain privacy seal programs. Their internal privacy practices should reflect and enforce the promises made. But, as privacy practices correspond to business processes, they can change frequently. It can be challenging to keep the promises up-to-date.

This article describes a methodology for enterprises to promise what they can keep. This is done by automatically transforming privacy practices into corresponding privacy promises that reflect the enterprise-internal behavior.

1 Introduction

Enterprises begin to actively manage and promote the level of privacy they offer to their customers. The goals are to obtain better publicity, to limit liabilities, and to comply with regulations. Visible signs of enterprises' privacy awareness are privacy statements and privacy seals. Customers can read such privacy promises explaining how collected data will be used. They can also examine the privacy seals (e.g., [TRU]) certifying that privacy promises exist and are accessible.

Whether or not the data inside the enterprise is used as promised depends on the enterprise's actual privacy practices as defined by the enterprise's chief privacy officer. Privacy practices reflect the business processes and should correspond to privacy promises. Today, both are synchronized manually. Since there is no sound notion of what this 'correspondence' means, they can easily get out of sync, especially if the privacy practices change frequently.

We show how consistency between practices and promises can be assured by an automatic transformation between privacy practices formalized using E-P3P [KSW02b] and privacy promises formalized using the W3C Platform for Privacy Preferences (P3P) [W3C02b]. This automated translation then ensures that privacy promises are kept up-to-date even if privacy practices change frequently. Another benefit is that enterprises can test or detect whether changes in their practices requires changes to the privacy promises made. This is important as the customer consents to a set of promises and if the actually enforced promises differ, the enterprise may be required to obtain updated consent from the customer.

1.1 Outline

In Section 2, we outline our privacy policy management model and its benefits. In Section 3, we describe E-P3P and P3P and give an example for an E-P3P policy and its corresponding P3P privacy promises. Section 4 describes the actual transformation procedure from E-P3P practices to P3P promises. In Section 6, we summarize lessons learned and identify some shortcomings and potential improvements of both languages. Section 7 concludes the article.

2 Managing Privacy Policies inside Enterprises

2.1 Different Types of Privacy Policies and their Consistency

We distinghish two types of privacy policies: Enterprise-internal *privacy practices* and published *privacy promises*.

Enterprise privacy practices define how data is collected, processed, and used. They are required to comply with legal regulations. In addition, they need to implement the privacy goals and business processes of the enterprise. Enterprise privacy practices can be formalized using (E-P3P) [KSW02b]. They can be very fine-grained and can define access rights down to individual employees. As a consequence, they may change frequently.

Privacy promises communicate certain privacy guarantees to the enterprise's customer. The most common form are textual privacy statements that explain what data is collected, how it is used, and what other enterprises may use it. Compared to enterprise privacy policies, they do not deal with enterprise-internals but offer a coarser-grained view, considering all the enterprise-internal data users and the enterprise's business agents as one data user. Privacy promises can be formalized using the Platform for Enterprise Privacy Preferences (P3P) [W3C02b].

An enterprise's privacy practices should be consistent with its privacy promises, i.e., should not allow behavior violating a promise. If, e.g., an enterprise promises not to disclose certain data to direct marketers, the practices should ensure that this will not happen. The enterprises also wants privacy promises to properly advertise good privacy practices, i.e., not to describe data usage or data disclosure that will be prevented by the privacy practices. If, e.g., an enterprise never discloses data to a direct marketer, it should not ask its customers for permission to do so.

2.2 Policy Management Model

The goal of the policy management model is to ensure consistency of published promises with frequently-changing enterprise-internal privacy practices. This is done by an automated translation of the enterprise-internal practices in E-P3P into privacy promises in P3P. The flows for managing policies are depicted in Figure 1:

1. The enterprise defines its E-P3P terminology[1]. This terminology fixes the scope of the enterprise privacy practices. In order to enable an automated translation,

[1] The enterprise may also use a pre-defined terminology or a terminology that has been standardized in a certain sector.

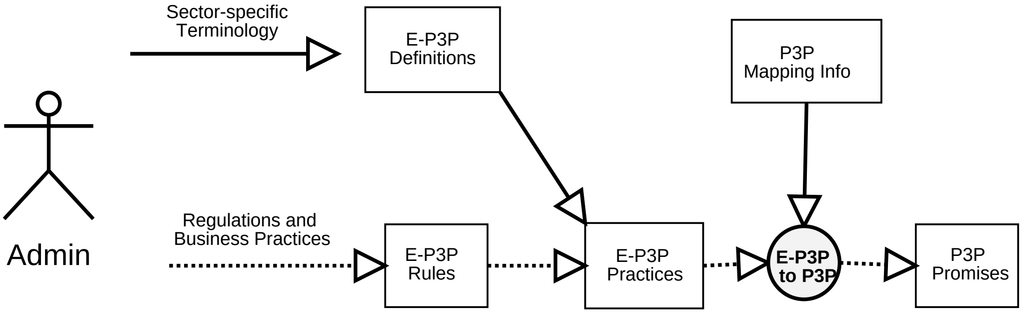

Figure 1: Flows of Enterprise Privacy Policy Management (dashed arrows denote frequent updates).

this terminology needs to be augmented with P3P specific details that cannot be derived from the E-P3P policy. This is depicted in the box "P3P Mapping Info".

2. The enterprise defines the E-P3P rules implementing the legal regulations and the business practices of the enterprise.
3. The resulting E-P3P policy is used as the default policy for using data and enforced throughout the enterprise. This can be done using traditional access control, E-P3P-aware business processes, or privacy-enabled access control systems such as [KSW02a].
4. The transformation we present in Section 4 can be used to automatically derive the corresponding P3P privacy promises from the given E-P3P policy.

3 Example: Privacy Practices and Corresponding Promises

We now illustrate our approach by describing an E-P3P policy of a web merchant and a corresponding P3P privacy statement that can be promised to the customers.

3.1 A Short Summary of E-P3P

An E-P3P policy contains definitions defining the terminology and a rule-set defining the actual permissions. E-P3P Definitions define data-categories[2] DC, purposes P, data users DU, privacy actions A, conditions $Cond$, and obligations Obl. Categories identify the types of data that need privacy-aware treatment. Purposes explain for what reason or business purpose the collected data will be used. Data users are parties accessing the data. The person whose data has been collected is a a distinguished data user called "data subject". Actions model the actual privacy-relevant operations on the data. Conditions are Boolean expressions evaluating context information such as consent. Obligations are duties imposed on the enterprise by the privacy policy, such as timely deletion of data. In E-P3P, data-categories, data-users, and purposes are ordered in hierarchies while actions, conditions, and obligations are sets.

[2] In the sequel, we denote the domain of the respective elements by sets. A set with $p3p$ subscript denotes a P3P domain. E-P3P domains are without subscript.

Data Categories:

/all/customer/financial
/all/customer/purchase
/all/customer/browsing
/all/customer/contact/postal
/all/customer/contact/homephone
/all/business-partners/financial
/all/business-partners/other
/all/anonprofiles

Data Users:

/all/internal/accounting
/all/internal/sales
/all/internal/r-and-d
/all/external/marketer
/all/external/deliverer
/all/external/telemarketer
/all/external/law-enforcer

Purposes:

/all/law-enforcement
/all/admin-r-and-d
/all/service/transaction/order
/all/service/transaction/delivery
/all/service/transaction/payment
/all/service/crm
/all/service/marketing/tele
/all/service/marketing/non-tele

Figure 2: E-P3P Data Category, Purpose, and Data User Hierarchies.

Each E-P3P rule in the rule-set is a tuple[3] $(dc,\ p,\ du,\ \pm a,\ o^*,\ c^*)$ with $dc \in DC$, $p \in P$, $du \in DU$, $a \in A$, $o \in \mathit{Obl}$, and $c \in \mathit{Cond}$ (x^* denotes a set of zero or more elements x). The rule defines that the data-user (and its descendants) can/cannot perform the action on the category (and its descendants) for the given purpose (and its descendants) under the conditions resulting in the obligations. If one rule allows an operation (i.e., +operation) while another denies it (i.e., -operation), then denial takes precedence.

3.2 A Web Merchant's E-P3P Privacy Practices

We now describe an example E-P3P policy reflecting a Web merchant's business and privacy practices. The definitions are depicted in Figure 2: The Web merchant collects data about customers and business partners. Data about customers is classified as either financial, purchase, browsing, or contact-related. Some of the customer data is used to produce anonymous profiles. Data about business partners could be financial or other. The Web merchant has three internal departments that use customer data: accounting, sales, and R-and-D. Its marketing is done by an external marketer agent. Delivery of the goods sold can be through an external delivery service. The web merchant also has contacts with an external telemarketer. It may also send data to a law enforcement entity. The merchant has two main classes of purposes: one being related to service to individual customers, the other one admin, research and development. Service to customers has sub-purposes marketing (tele- and non-tele-marketing), customer relationship management (CRM), and services related to the transaction (order, delivery and payment). From browsing and purchasing information, the enterprise also derives anonymous behavior profiles.

The given definitions should be quite stable. The rule-set in Figure 3 reflects a simple set of permissions. The sales department can read all the customer data (positive rules) except for financial data (negative rule) for the purposes of CRM and order. The accounting department can read customers' financial data for the purpose of payment but has to delete the data after thirty days, as indicated by a delete obligation. The R&D department can read purchase and browsing data for admin and R&D purposes. The external delivery service can read customer postal contact data for delivery purposes. The external marketing company can read customer postal contact data for non-tele-marketing

[3] For brevity, we omitted the precedences in E-P3P rules. How precedence can be removed by preprocessing is sketched in Section 5.1.

(**Category**[a]	**Purpose**	**Data User**	**Action**	**Oblig.**	**Condition**)
(/all	//order	//internal/sales	+read	-	-)
(/all	//crm	//internal/sales	+read	-	-)
(//financial	//order	//internal/sales	-read	-	-)
(//financial	//crm	//internal/sales	-read	-	-)
(//financial	//payment	//internal/accounting	+read	delete(30d)	-)
(//purchase	//admin-r-and-d	//internal/r-and-d	+read	-	-)
(//browsing	//admin-r-and-d	//internal/r-and-d	+read	-	-)
(//postal	//delivery	//external/deliverer	+read	-	-)
(//contact	//marketing	//external/marketer	+read	-	opt-in)

[a] The elements are identified using XPath [W3C99]; "//[name]" denotes the unique node in our hierarchies with "name".

Figure 3: E-P3P Rules Reflecting the Web Merchant's Business Practices.

purposes if the user opted in for that purpose. The enterprise does not share any data with the telemarketing company as there is no rule allowing this.

3.3 A Short Summary of P3P

The W3C Platform for Privacy Preferences (P3P) [W3C02b] defines a protocol and an XML format for privacy statements. It allows Web sites to inform Web users of their data collection and data use practices in a machine-readable way. It enables Web users to understand what data will be collected by visited sites, how that data will be used, and what data/uses they may opt-in/opt-out to. User agents can use a Web site's P3P policies to inform users of a Web site's privacy practices (by displaying the human-readable equivalent of the P3P policy) and/or make automated decisions based on comparing a Web site's practices with the user's privacy preferences. The use of P3P policies by user agents is not specified by the P3P specification. A P3P policy file contains some policy information, a data schema, and a list of actual privacy statements. The data schema may be included in the policy file, or may be described in an external data schem file pointed to in the policy. The policy information contains information about the policy's issuer, about possible dispute resolution mechanisms, and about whether the enterprise grants the data subject access to his data. The data schema defines abstract data types called data elements in the domain DE_{p3p} that can be organized hierarchically. Data elements are used to identify data that is collected from data subjects. P3P pre-defines a base data schema that must be understood by all user agents and defines re-usable structures and a set of pre-defined data types. A policy is free to define its own data schema (possibly re-using structures defined in the base data schema) or to use only elements of the base data schema. P3P also defines a flat set of data categories DC_{p3p}={physical, demographic, socioeconomic, ...} (see [W3C02b] for the complete list). Data elements can then be labelled with one or more categories.

The list of privacy statements defines the actual permissions granted by the P3P file. Each statement contains the following elements:

- A group of data elements $de_{p3p} \in DE_{p3p}$ to which this statement applies.[4] Optionally, such a group can be declared non-identifiable, signalling that the data

[4] Note that P3P allows the same data element to occur in many statements.

Data Element	Category	Base Schema Structure
customer.financial	financial	
customer.purchase	purchase	
customer.browsing	navigation	
customer.home-info.postal	demographic, physical	postal
customer.home-info.postal.name	demographic, physical	personname
customer.home-info.postal.. . .	. . .	. . .
customer.home-info.telecom	physical	telecom
customer.home-info.telecom.telephone	physical	telephonenum
customer.home-info.telecom.. . .	. . .	. . .

Figure 4: Enterprise-specific P3P data schema

will be anonymized before being disclosed to this data user. Permissions are inherited down, i.e., if a purpose by a data user is allowed on a data element, it is also allowed on possible sub-elements. A data element can be declared optional, in which case a customer can choose whether ot not to provide it.

- A set of purposes $p_{p3p} \in P_{p3p}$ with $P_{p3p} := \{$current, admin, $\ldots\}$ (see [W3C02b] for the complete list) for which data is collected. A purpose can be declared as optional ("opt-in", "opt-out") or mandatory ("always").

- A set of data users (called recipients) $du_{p3p} \in DU_{p3p}$ with $DU_{p3p} := \{$ours, same, other, $\ldots\}$ (see [W3C02b] for the complete list) with whom the data will be shared. Also recipients can be declared optional.

- A retention policy indicator $ret_{p3p} \in RET_{p3p}$ with $RET_{p3p} := \{$no-retention, stated-purpose, legal-requirement, business-practices, indefinitely$\}$ indicating how long the data will be stored.

3.4 The Web Merchant's P3P Policy

We now look at the Web merchant's P3P promises, consisting of a P3P policy file together with a data schema. We make the assumption that all the customer data used by the enterprise is at some point collected. The enterprise's data schema is depicted in Figure 4; it only needs to reflect identified customer data (not the anonymous profiles or the business-partner information). The customer data set (an enterprise-defined extension of the base data schema's user data set) re-uses some of the data structures ("postal", "personname") from the base data schema and inherits their subelements (and categories). Assuming that the customer.financial data element corresponds to the E-P3P /all/customer/financial data category, customer.purchase to /all/customer/purchase etc., the statements in his P3P policy could be the ones shown in Figure 5. E-P3P purposes, data users and opt-in conditions are mapped to sets of pre-defined purposes and recipients, and opt-in delarations. All the internal departments as well as marketer are indicated with ours (ourselves and our agents). The delete obligation is translated in a retention for stated-purpose, whereas the other retention delarations (not explicitly declared in E-P3P) are assumed business-practices.

(**Data Element**	**Purpose**[a]	**Recipient**	**Retention**)
(customer.home-info	current, ind-a, ind-d	ours	business-practices)
(customer.purchase	current, ind-a, ind-d	ours	business-practices)
(customer.browsing	current, ind-a, ind-d	ours	business-practices)
(customer.financial	current	ours	stated-purpose)
(customer.browsing	admin, develop, pseudo-a	ours	business-practices)
(customer.purchase	admin, develop, pseudo-a	ours	business-practices)
(customer.home-info.postal	current	same, delivery	business-practices)
(customer.home-info.postal	contact(opt-in)	ours	business-practices)

[a] ind-a, ind-d pseudo-a stand for individual-analysis, individual-decision and pseudo-analysis.

Figure 5: P3P Statements Corresponding to the E-P3P Policy.

3.5 Some Observations

A typical P3P policy is more coarse-grained than the Web merchant's P3P policy defined above, as usually each data element (such as customer.home-info.postal) only appears in one statement. Also, it would group customer.home-info, customer.purchase and customer.browsing in one statement as their P3P statements are identical. This is the result of the fact that ours does not distinguish between different departments or enterprises' agents.

Even an a-typical P3P policy of the granularity level above, with data types closely mapping E-P3P categories, cannot be as fine-grained as its E-P3P equivalent. Whereas the E-P3P policy defines exact data users and data flows within the enterprise, the P3P policy classifies data recipients according to notions of their privacy policy (same, unrelated), business relationship with the enterprise (ours), or service (delivery). Whereas the E-P3P policy can define an exact retention time by mandating a deletion at a certain point in time, P3P policies have generic retention classes (stated-purpose, business-practices). This requires a mapping or transformation from E-P3P to P3P to transform fine-grained to coarse-grained, and concrete and absolute to abstract and relative.

4 How to Translate E-P3P into P3P

4.1 General Approach

E-P3P focuses on enterprise-specific enforcement; P3P on enterprise-independent information. The P3P policy stated by an enterprise should never publish better (stricter) privacy practices than actually enforced through the E-P3P policy. On the other hand, we would like the P3P policy to adequately reflect the E-P3P practices.

Our transformation procedure will transform an E-P3P policy into a 'best-approximation' P3P policy, using a chosen (base or enterprise-specific) data schema and P3P-specific mapping information.

The core of the transformation translates each E-P3P rule into an P3P statement. This transformation assumes that the E-P3P policy is 'fine-grained': it contains only positive authorizations ('allow') for all element-vectors where access is allowed and defaults to the ruling 'deny' if no rule is applicable. A fine-grained E-P3P policy can be derived from a generic E-P3P policy (with positive and negative authorizations and precedences) by pre-processing. How this can be done is sketched in Section 5.1. The P3P policy that is output by the transformation is fine-grained, too. This means that multiple statements may govern the use of the same P3P data element. A fine-grained P3P policy can be aggregated to a coarser-grained P3P policy where each data element is only governed by one statement. If applicable, we give hints for this aggregation. More details are described in Section 5.2.

Each chapter of this section defines a particular map for mapping one kind of E-P3P elements onto P3P elements. We collect all mapping information in a vector called $\mathit{MapInfo}$. Section 5.4 describes alternatives for storing the $\mathit{MapInfo}$ information.

4.2 Data Categories and Elements

4.2.1 General Observations

The P3P base data schema defines four data type hierarchies (`user`, `dynamic`, `business`, `third-party`) which can be augmented by additional data schemas. The P3P categories (such as `physical`, `demographic`, `financial` ...) are flat and used as labels into these data type hierarchies. P3P policy statements about the usage of data can use about fixed-category data elements or variable-category elements. A statement about a fixed-category data element `user.home-info.postal` gives information about the data type (postal contact information) and implicitly (through the base data schema) about its categories (`physical` and `demographic`). A statement about a variable-category data element such as `dynamic.miscdata` needs to be accompanied by the categories associated with it in this statement; it only communicates that this is miscellaneous data with these categories attached.

P3P user agents' preferences and interpretations may be more targeted at data types or at categories, or both. In addition, the fact that multiple data elements are grouped into one policy statement specifies common collection and usage practices for these data.

E-P3P has a hierarchical data (category) model with enterprise-specific data categories. The definition of the data deployment model (mapping of individual pieces of data to categories) is outside the scope of E-P3P.

For a mapping from E-P3P to P3P, we need to express E-P3P categories in terms of P3P categories and/or data elements. (A detailed projection of E-P3P rules, including obligations and conditions, to P3P, also has to take E-P3P deployment data into account). This entails:

1. A mapping between E-P3P categories and P3P data schema elements

2. For E-P3P categories which do not map to P3P the base data schema elements, the definition of an enterprise-specific P3P data schema

3. An assignment of P3P categories to variable-category data elements in the base data schema as well as any enterprise-specific data schema.

Some assumptions and decisions may simplify the mapping. E.g., one could omit most of 1 and 2 by only mapping E-P3P categories to (one or more) P3P categories, as represented in Figure 6, where each leaf in the E-P3P data hierarchy is labelled with one or more P3P categories; categories accumulate upward in nodes: each node collects all the categories of its children (not shown in Figure 6). Using such a mapping, an E-P3P rule

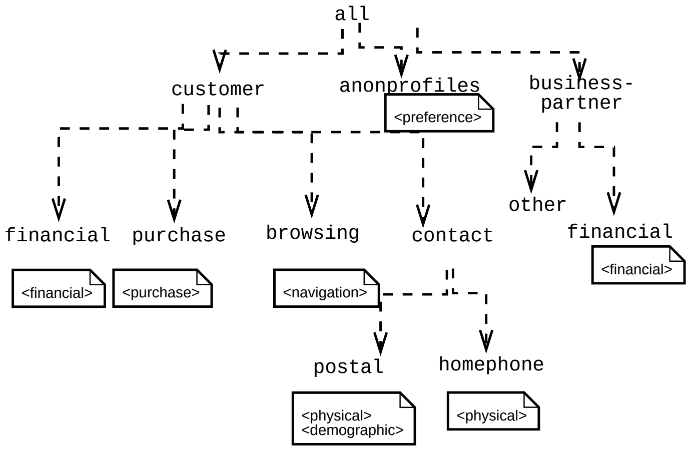

Figure 6: Mapping of E-P3P Data Categories to P3P Data Categories Only

allowing marketer (one of our agents) to read customer-contact-postal data for non-telemarketing:
(//customer-contact-postal, //non-tele-marketing, //marketer, +read, -, -)
would be translated into a P3P statement about a miscellaneous data element with categories demographic and physical, purpose contact and recipient ours:
(dynamic.miscdata(physical, demographic), contact, ours).

The advantage of this mapping is that it bypasses any data modelling in P3P, and the resulting P3P policy can be interpreted well by P3P user agents specialized in interpreting category information. However, it does not allow user agents to make interpretations and decisions based on data types. The next section presents a more general solution.

4.2.2 General solution

P3P user agents can use both data element information as category information to interpret P3P statements and make decisions. Thus, a general solution should exploit the full potential of P3P and of user agents enforcing the full range of privacy preferences one can express with APPEL [W3C02a]. It should enable user agents to act on data types as well as categories; it should allow re-use of pre-defined categories as well as the other category; it should use the P3P base data schema and its category assignments but also allow for the definition of a new P3P data schema (with appropriate P3P category associations). The most general data mapping labels each E-P3P leaf category representing *P3P-relevant* data (identifiable customer data) with zero or more P3P data types (data elements). These data elements can be taken from the base data schema, or from an enterprise-specific data schema, where data elements are appropriately labeled with P3P categories. This way, we associate with each E-P3P data category the corresponding P3P data elements as well as P3P categories, giving user agents the choice whether to use only data type information, only category information or both. The P3P enterprise-specific data schema is depicted in Figure 4. The actual mapping information set is depicted in Figure 7 and formalized as a mapping and a subset of E-P3P categories that identify “non-identifiable” categories of E-P3P:

$$DataMap \quad = \quad \{CategoryMap, NonIdentMap\}$$

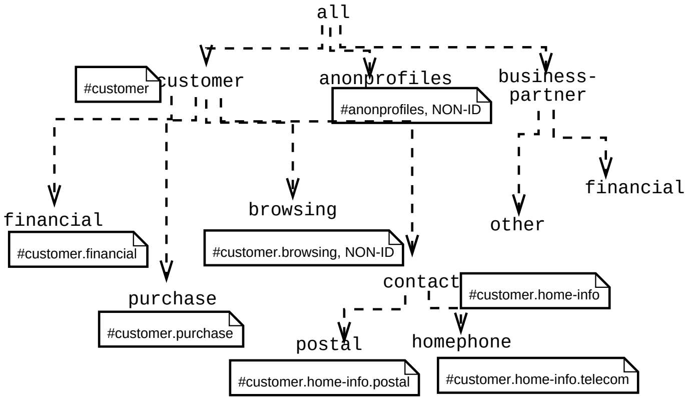

Figure 7: Mapping of Data Categories: Each E-P3P Category is Labeled with the Collected P3P Data Elements and an optional NonID tag.

$$\begin{aligned}&\text{with } \mathit{CategoryMap} \subseteq \mathit{DC} \times \mathit{DE}_{p3p}\\ &\text{and } \mathit{NonIdentMap} \subseteq \mathit{DC}\end{aligned}$$

In the example, this mapping maps rules about /all/customer/contact/postal to a P3P statement about customer.home-info.postal.

Data element labels in $\mathit{CategoryMap}$ accumulate upward as each node category collects its children's' data element labels (not shown in Figure 7). A non-identifiable label in $\mathit{NonIdentMap}$ does not propagate upward to a parent node unless all the children of the parent node are non-identifiable. The set of E-P3P categories in DE_{p3p} which are part of $\mathit{CategoryMap}$ contains at least the leaf elements of DE_{p3p} which correspond to P3P-relevant P3P data elements.

The policy administrator creating $\mathit{CategoryMap}$ may decide to also include non-leaf elements of DC in $\mathit{CategoryMap}$. E.g., in Figure 7, /all/customer and /all/customer/contact are also labeled with P3P data elements. This later facilitates automatic aggregation of the resulting fine-grained P3P policy: Assume that the data schema also contains customer.home-info.online, but the enterprise currently does not collect this information. If the E-P3P rules about //postal and //homephone were identical, the translation would lead to identical P3P statements about customer.home-info.telecom and customer.home-info.postal. These statements could not, however, be automatically aggregated into a statement about customer.home-info as this could lead a user agent to interpret that the enterprise also collects e-mail). The node labeling indicates that such an aggregation is allowed (either because the administrator knows this situation cannot occur, or because he wants to allow it).

Note that the mapping can be many-to-many: we cannot exclude that the enterprise's data storage system stores the same P3P data element as part of multiple E-P3P data categories. Specifically, a data element could be stored in a non-identifiable way as a part of a non-identifiable E-P3P data category, and in an identifiable way as part of an identifiable E-P3P data category. This results in multiple P3P statements about the same data. When aggregating such seemingly conflicting statements, one needs to make a worst-case approximation by retaining the stronger statements granting the maximum permission to the enterprise while discarding weaker statements. The same approach will be applied for the mapping of E-P3P data users and purposes to their P3P equivalents.

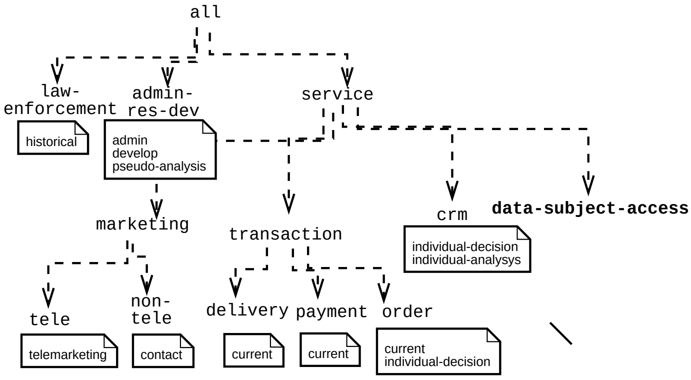

Figure 8: Purpose Mapping

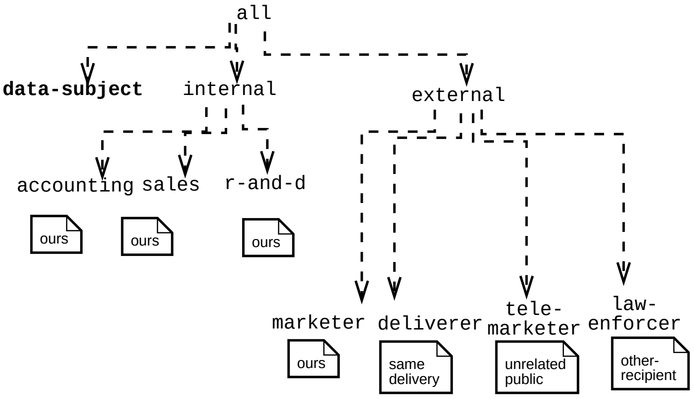

Figure 9: Data User Mapping

4.3 Data Purposes and Data Users

Data purpose and data user mapping are similar. Each mapping labels P3P-relevant leaf elements of the E-P3P hierarchies (data users or purposes acting on P3P-relevant data) to one or more elements from the corresponding P3P set (purposes P_{p3p} or recipients DU_{p3p}):

$$\begin{aligned} \mathit{UserMap} &\subseteq DU \times DU_{p3p} \\ \mathit{PurposeMap} &\subseteq P \times P_{p3p} \end{aligned}$$

As P3P only mandates purpose and recipient elements for statements about identifiable data elements, the labeling is optional for E-P3P purposes or data users acting only on `non-identifiable` data. Also here, labels accumulate upward into parent nodes.

Figures 8 and 9 show the representations of both mappings for our example. In Figure 9, we have labeled all the internal departments `ours` as well as the marketing service, which acts as the merchant's agent. The external data users have similar, unknown, or other privacy practices.

We added a distinguished purpose `/all/service/data-subject-access` and data-user `/all/data-subject` to the E-P3P hierarchies which allows us to formulate E-P3P rules expressing data subjects' access rights (P3P ACCESS element) as discussed in Section 4.6.

Two other special purposes, `pseudo-decision` and `pseudo-analysis` deserve special attention. While `non-identifable` seems to be a feature of the data during or after collection, the P3P purposes `pseudo-decision` and `pseudo-analysis` may act on identifiable data but with the purpose of making pseudonymous decisions or building pseudonymous profiles.

In Figure 8 we labeled `/all/admin-res-dev` with `admin`, `develop` and `pseudo-decision`. In our definition, this indicates that all the processing for this E-P3P purpose is pseudonymous. The labeling of the same E-P3P purpose with both `pseudo-decision` and `individual-decision` or `individual-analysis`, however, we define to be semantically invalid.

4.4 Optional Data, Opt-In and Opt-Out

In both E-P3P and P3P, notions of choice and options are mixed with notion of user consent for specific usages of data. The approach taken here is that a user consents to a policy including (or modified with) specific opt-in or opt-out choices specified by the policy. A condition such as: "if consented to by the user" then means: "if the user consented to the policy and made this specific (opt-in or opt-out) choice."

In P3P, opt-in and opt-out choices are attached to purposes and/or to data users within the same statement; and also each data element within a statement can be optional or mandatory for the set of purposes and recipients in that statement.

As data use (or, in P3P, 'sharing') is always associated with a recipient and a purpose, the difference in semantics between an optional (opt-in or opt-out) purpose or recipient disappears when considering tuples with atomic elements (one data element, one purpose and one recipient). In addition, the collection of data, the use of which is optional and not consented to by the user, should always be optional, regardless of whether it is declared as such: whether or not data collection is optional should be consistent with (choices about) its use.

In E-P3P, data subject consent (or, more specifically, opt-in or no opt-out of certain uses of data) is tied to a specific rule, and thus to a combination of data category, data user, data purpose and action). The need for the presence of an opt-in choice or the absence of an opt-out choice is represented by a condition verified at run-time based on context containers provided by the application.

In order to map E-P3P's opt-in and opt-out conditions to P3P choices, we first define which E-P3P condition identifiers are interpreted as opt-in and opt-out conditions. Let OptMap define the identifiers of the E-P3P conditions testing the presence of opt-in and the absence of opt-out:

$$\mathit{OptMap} = \{\mathit{opt\_in\_cond}, \mathit{opt\_out\_cond}\} \in String \times String$$

When transforming a fine-grained E-P3P policy to fine-grained P3P, a rule with an $\mathit{opt\_in\_cond}$ or $\mathit{opt\_out\_cond}$ is transformed into a P3P statement with `opt-in` or `opt-out` for the stated purpose and an `optional`='yes' for the data: the data collection is optional (for this purpose and recipient). When aggregating statements about the same data into one statement, we can only assign `optional`='yes' if it is 'yes' for all occurrences.

4.5 Data Retention and Deletion

P3P uses a set of abstract values expressing how long data is retained: RET = {`no-retention`, `stated-purpose`, `legal-requirement`, `indefinitely`, `business-practices`}; several of these may apply to the same data. For all the retention values other than `no-retention` (which is basically "current session") and `indefinite`, the site's human-readable policy must give more information.

An E-P3P policy implementing any retention policy should enforce that the E-P3P authorization engine mandates deletion of data corresponding to the targeted retention policy. As a consequence, we mandate a 'delete' obligation to any 'store' rule about data that has a finite retention. The deletion may be conditional on consent obtained by the data subject. The transformation then uses these obligations to derive the appropriate P3P retention label to be assigned to each data element that is collected. If data can be used for several purposes, some of which are optional, and these purposes have different retention times or policies, the actual deletion of the data should occur at the maximum retention time for the purposes to which the user consented (or which were required). As consent may not be known at collection and store time, this implies that the 'store' rule execution creates delete obligations for each of the data use purposes, and that each of them delete obligation, at scheduled execution time, only actually deletes data if no other pending delete obligations for the same data for consented use exist.

Take the example of `/all/customer/contact/postal` data which can be used by deliverer (for `current` purpose) and by marketer (for `non-tele-marketing` purpose). Assume retention periods for these purposes are two, respectively twelve months. Storing `/all/customer/contact/postal` puts two delete obligations on the obligation stack; the first one (executed after two months) will delete the data only if the user has not opted-in for non-tele-marketing, in which case also the second delete obligation is taken from the obligation stack. In P3P, the published retention in the fine-grained P3P policy will be `stated-purpose` for both statements.

We now give a more detailed description of the mapping and transformation achieving retention and deletion consistency between the E-P3P and P3P policies. We assume that E-P3P policy writers create the appropriate store rules and obligations; however, the procedure can easily be adapted to derive the correct store rules starting from desired P3P retention values (see also Section 4.9).

We do not consider a mapping to `business-practices`: a company can either state its business practices in the form of purposes (and thus can claim `stated-purpose` retention) or keeps the data for purposes not consented to by the data subject, in which case we consider the retention to be equivalent to `indefinitely`. The mapping is defined as follows:

- We define a $RetLawMap \in P$ indicating which E-P3P purpose is associated with law enforcement. This allows to treat retention for law enforcement as retention for any other purpose.
- For each tuple (data category, purpose) occurring in an authorized rule with a P3P-relevant action (see Section 4.7), define the retention time and a human-readable explanation of the use:
  $RetTimeMap \subseteq T \times P \times \{String\} \times \{Time\}$

When transforming an E-P3P rule to P3P statement, we now proceed as follows:

- If the data in the rule is not stored by any store rule, retention no-retention.
- Else if, among the possible multiple delete obligations in the data's store rule, there is an obligation to delete the data after the purpose-specified time in $RetTimeMap$, retention is stated-purpose (or law enforcement if the purpose is $RetLawMap$) with explanation of the use as in $RetTimeMap$.
- Else, retention is indefinitely.

A P3P data-aggregation procedure can then derive the retention value for a data element occurring with different retention values.

- If a data element has a retention of indefinitely in any of the statements, then the retention value of the grouped statement is indefinitely.
- Else, if the data element has a retention of law-enforcement or stated-purpose in any of the statements, these are copied into the retention for the aggregated statement.
- Else, retention in the aggregated statement is none.

Note that for a transformation to be meaningful, we have to assume that the E-P3P policy is correctly enforced throughout the enterprise. In addition, we require that the E-P3P policy follows certain conventions. E.g., an enterprise's retention policy can only be meaningfully translated if it is implemented through 'delete' obligations in 'store' rules; any other retention mechanism would go undetected by the translation and will result in indefinitely retentions in P3P. Also, if an enterprise allows storage of data without authorization through the E-P3P engine, the P3P version will not reflect the actual retention policy.

4.6 Data Subject Access

The "ACCESS" element in a P3P policy describes the data subject's access (read or update) rights to identified data collected about him. P3P does not specify a mechanism for it, although it seems implied that data subjects access their data by contacting a representative of the enterprise. Indeed, a real enforcement by giving concrete E-P3P access rights to data subjects is not desirable. However, we can model the notion of access in E-P3P by defining a purpose (e.g., data-subject-access) and a data user or role (e.g., data-subject) which can be used by the authorized enterprise representative to access data on behalf of data subjects (after appropriate authentication of the data subject).

The values in the set $ACCESS$ = {nonident, all, contact-and-other, ident-contact, other-ident, none} indicate that: the web site does not collect identified data; access is given to all identified data; access is given to (some[5]) identified online and physical contact information as well as to certain other identified data; access is given to (some) identified online and physical contact information; no access to identified data

[5] Any disclosure (other than all) is not meant to imply that access to all data is possible, but that some of the data may be accessible and that the user should communicate further with the service provider to determine what capabilities they have.

is given. In order to derive a P3P access statement from an E-P3P policy, mapping information has to specify:

- which E-P3P data user $AccessSubject \in DU$ and purpose $AccessPurpose \in P$ correspond to data subject access. E.g., $AccessSubject$=`/all/data-subject` and $AccessPurpose$=`/all/data-subject-access`.
- values for zero or more of the subsets $AccessMapAll \subseteq DC$, $AccessMapContact \subseteq AccessMapAll$, $AccessMapContactAndOther \subseteq AccessMapAll$, $AccessMapOtherIdent \subseteq AccessMapAll$, $AccessMapIdentContact \subseteq AccessMapContact$ indicating which sets of data correspond to P3P `all`, `contact`, `contact-and-other`, `other-ident`, `ident-contact`. E.g., in our example, $AccessMapAll$ = `/all/customer` and $AccessMapContact$ = `/all/customer/contact`;
- which action(s) $AccessMapActions \subseteq A$ correspond to data-subject access. E.g., $AccessMapActions = \{\texttt{read}, \texttt{update}\}$.

If the E-P3P data hierarchy contains no identifiable customer information, the P3P value for ACCESS is `nonident`. Otherwise, if $AccessMapAll$ is defined and appropriate authorizations exist for access to $AccessMapAll$ by $AccessSubject$ for executing any action in $AccessMapActions$ for purpose $AccessPurpose$, the P3P value is `all`. Otherwise, authorizations for the other defined sets are checked until a set is found which is defined and has corresponding authorizations.

4.7 Actions

Finally, we define as ‘P3P-relevant’ actions the ones that can be interpreted as ‘usage’ or ‘sharing’ in P3P; this will define which of the E-P3P rules need to be transformed. Of the set of E-P3P actions we used throughout the example: {`read`, `update`, `store`, `delete`}, `store` and `delete` are relevant for retention but need not be translated. For our example, the P3P-relevant actions $ActionMap \in A = \{\texttt{read}, \texttt{update}\}$. In addition, a rule about an action a will need to be transformed into a P3P statement only if the data user is not the dedicated $AccessSubject$.

4.8 Disputes, Contact and Other Policy-Specific Statements

Most of the general policy information (such as dispute and some contact information) can not or only partially be derived from the E-P3P policy and thus has to be added by the mapping information. An exception is the “`access`” element which can be derived from E-P3P rules (see Section 4.6).

Therefore, we define our last mapping set $GenMap$ that contains appropriate values for general policy information which is not present in E-P3P, such as: the name of the P3P policy, the location for a human-readable version, the URL for opting-in and opting-out, and information about dispute resolution and remedies: $GenMap = \{PolName, PolOptURI, \ldots\}$

4.9 The Transformation Procedure Summarized

The complete procedure for transforming a generic E-P3P policy to a corresponding P3P policy consists of following two preparation steps that need to be done once:

1. The designer of the transformation defines the P3P data schema to be used. It may be the base data schema or an enterprise-specific data schema. The mapping is easier and yields finer-grained results the more the data sets in the P3P data schema correspond to sub-hierarchies in the E-P3P hierarchy. Re-using the base data schema should result in better interpretation by some user agents.

2. The designer of the transformation defines the different mappings. Depending on the E-P3P policy, some of these mappings may be empty: for mapping elements such as $AccessPurpose$, $AccessSubject$, $RetTimeMap$ if may be impossible to define values if the E-P3P policy was not written with retention or access goals in mind. This leads to a `none` value for access, and to `indefinitely` values for retention.

Whenever a given E-P3P policy shall be translated into P3P, this information is then used in the actual transformation. The transformation consists of the following steps:

3. The E-P3P policy is translated into a fine-grained E-P3P policy.

4. The fine-grained E-P3P policy is transformed into a fine-grained P3P policy.

   The general P3P policy information is extracted partially from the E-P3P policy (e.g., contact information), partially from $GenMap$; and the data schema (or a pointer to it) is inserted. Each of the fine-grained E-P3P rules with a P3P-relevant action and with a data-user not being the designated data-subject, is translated into a P3P statement where data group, recipients and purposes correspond to the P3P labels of the corresponding E-P3P elements; and where retention as well as data, purpose and recipient optionality are determined as in Sections 4.4 and 4.5.

5. The fine-grained P3P can optionally be aggregated into a coarser-grained P3P policy.

   Optionally, an automatic (one statement per data-element) or semi-automatic (the administrator identifying data to be grouped in a statement) data aggregation process can aggregate statements about the same or multiple data elements into one statement, applying the aggregation procedures discussed in Section 5.2.

6. The resulting P3P policy is published on the web-site.

5 Implementation Details and Observations

5.1 Fine-graining an E-P3P Policy

An E-P3P rule contains allow and deny rules for hierarchical elements. A rule may have arbitrary conditions and obligations; it may also have a precedence attached to it. In E-P3P, a rule is applicable to a given tuple if its conditions are met and if its elements cover (directly or by inheritance) the given tuple. If there are applicable rules with different precedence values, the applicable rules with the highest precedence apply. All lower-precedence rules are ignored. If there are applicable "allow"- and "deny"-rules on the highest precedence level, then the "deny"-rule applies.

Fine-graining an E-P3P policy removes the deny rules, as well as the conditions and obligations that are not recognized by the translation. Of the conditions, only the specific opt-in or opt-out conditions that are defined in $OptMap$ are retained; of the obligations, only the delete obligations associated with store rules are retained. Deleting conditions and obligations will be done such that the resulting imprecision only causes 'worse-case' P3P statements.

As we consider only 'privacy-friendly' obligations (notifying a user, asking a user's consent), deleting or ignoring obligations can only cause a P3P policy to be less privacy-friendly ('worse-case'). As for conditions, ignoring a condition on an allow rule can only make the policy 'worse-case': an action being allowed promises less privacy than an action conditionally being allowed. Ignoring a condition on a deny rule may have the inverse effect: an action not being allowed promises better privacy than an action not being allowed under certain conditions. Thus, in the following, when evaluating which allow or deny rules are applicable to a certain tuple, the general approach for conditions which cannot be modeled by P3P is as follows: an allow rule with such a condition is always applicable, while a deny rule with such a condition by default is not. An exception case where we can conclude applicability of a deny rule with (a) generic condition(s), is where it overrides an applicable allow rule (with the same precedence) with the same (or more) generic conditions: if the conditions in the allow rule are met, then the conditions in the deny rule are met as well.

After deleting obligations (other than delete obligations), fine-graining consists of two steps, the first one dealing only with usage rules, the second one with store rules and delete obligations. The basic idea of the first step is to iterate over any leaf tuple of data category, purpose, data user, and action in the domains of these elements. For each tuple, the algorithm tries to assess whether access is allowed or not and whether a opt-in or opt-out conditions exist. This is done as follows:

1. Check for the highest-precedence allow-rule that covers the tuple (directly or via inheritance). Check for opt-in or opt-out conditions.

2. Check whether there exists a same- or higher-precedence deny-rule that covers the tuple. This deny rule is only applicable if it either has no conditions or else a sub-set of the conditions in the allow-rule.

3. Add the tuple and the opt-in and opt-out condition to the fine-grained policy if an allow-rule but no applicable deny-rule exists.

4. If the given precedence level did not lead to a tuple being added to the fine-grained E-P3P policy, restart while ignoring the allow-rules that have already been considered.

In the second step, using a similar iteration procedure, one fine-grained store rule is created for each leaf data category. If, on the highest precedence level, more than one store rule (with different conditions or obligations) apply to the same leaf data category, they are merged in one store rule containing the intersection of their delete obligation sets: as we cannot know which would apply at run-time, this is a ‘worst-case’ representation of guaranteed delete conditions.

5.2 Aggregation and Conflict Resolution of Fine-grained P3P

The P3P policy that results from the transformation in Section 4 is ‘fine-grained’: multiple P3P statements may govern the use of any given data element. Each statement defines one particular use of the data element. Such a P3P policy is syntactically correct but may be difficult to interpret by user agents. Through re-aggregation, a fine-grained P3P policy can always be transformed into one where each data element occurs in one statement only, or even where multiple data elements are grouped in a statement. This generally causes a loss of granularity: grouping all the statements about `customer.home-info.postal` in the Web merchant’s P3P policy into one would give recipient `delivery` the right to `contact(opt-in)` the customer for marketing.

This aggregation in general will group multiple statements into one and thus will necessarily produce a coarser-grained P3P policy. Re-aggregation also needs to resolve potential ambiguities that could arise from allowing data to appear in multiple statements. Examples of ambiguities in the fine-grained P3P policy are:

- A data element which is non-identifiable in one statement but identifiable in another statement;
- A data element which is optional for a certain purpose and given recipient in one statement, and mandatory in another statement for the same purpose and recipient;
- A data element which has a longer retention in one statement than in another.

Some of these ambiguities can already be avoided in the transformation step: e.g., as ‘non-identifiable’ is a feature of the data category, we could enforce consistency already during translation. For others, it may be easier to allow them in the fine-grained P3P policy as long as their resolution during re-aggregation is well-defined (e.g., the retention policy of a data element is the maximum of the retention statements of its different occurrences).

The re-aggregation procedure may:

- group statements about the same data by defining unions of its sub-elements (e.g., the union of two “optional” values is their logical AND; the union of “`opt-in`” and “`opt-out`” is “`opt-out`”; the union of “`opt-out`” and “” is “”.
- make statements about parent data types resulting from equal statements for children.
- group statements about groups of data collected together if so required, by using the same union mechanisms.

5.3 P3P and E-P3P Have Different Semantics on Data Elements

When translating E-P3P into P3P, one might be tempted to translate the data category hierarchy directly into the data element hierarchy of P3P.[6] We now explain why this

[6] The temptation would only apply for data elements since these are the only hierarchical elements in P3P.

direct translation of non-leaf nodes of the hierarchy is not possible.

The reason is caused by the fact that the semantics of non-leaf nodes of the data hierarchy is different in P3P and E-P3P. In E-P3P using an inner data node in a rule means that all parts of this node can be used. As a consequence, the E-P3P authorization engine will allow access to an inner data node only if access to all its parts are allowed. In P3P, using a inner data element in a statement defines that this element or its children may be used. From a policy point of view, this means that in P3P, "allow" rules inherit up and down while in E-P3P, they only inherit down.

Assume the E-P3P policy allows read access of `/all/user/contact/postal` to marketer for `/all/service/non-tele-marketing` purposes, and to `/all/user/contact/homephone` to telemarketer for `/all/service/tele-marketing` purposes. If we want the P3P policy to convey only information about the use of contact information for aggregated marketing purposes, can we derive this statement by evaluating access by marketer, respectively telemarketer, for `/all/service/marketing` to `/all/user/contact`? The answer is no: the E-P3P authorization answer will be 'no' in both cases as not all contact information can be used for all marketing purposes, while the P3P statement should be 'yes': *some* contact information can be used for *some* marketing purposes.

5.4 Alternatives for Storing the Mapping Information

The mapping information $MapInfo$ that summarizes the maps for each element type can be stored in a number of ways:

- It can be stored outside of the E-P3P and P3P data and policy files (e.g., in a separate XML file describing only the correspondence of elements between P3P and E-P3P). This allows the most flexibility for implementing different conversion directions and does not impact any of the P3P and E-P3P policy data schemas.
- It can be added to (inserted into) the P3P data schema definitions, by adding elements to P3P data definitions and categories indicating which E-P3P category they correspond to. This could be useful if converting from P3P to E-P3P but is not considered here.
- It can be added to the E-P3P data category definition, by adding extensions to the E-P3P data category elements which can be interpreted by an E-P3P to P3P policy converter.

The latter representation corresponds well to the 'labeling' way of illustrating different subsets of $MapInfo$, as we describe in the following sections.

5.5 P3P-aware E-P3P editing

Writing the E-P3P policy and writing $MapInfo$ are independent processes and may be performed by different administrators. The more interaction there is between these processes (at least initially), the more likely it is that policy writers will formulate the E-P3P rules such that they indeed enforce and create an acceptable or desired P3P policy. Without any feedback from the mapping and transformation process, an E-P3P policy writer is not likely to create the exact rules or definitions that can support P3P access or limited retention statements; and $MapInfo$ may be incomplete because some of its elements (e.g., $AccessPurpose$ or $AccessSubject$) are impossible to define, resulting in 'worst-case' P3P values.

Even in the presence of interaction and feedback, it may be very difficult for a policy administrator to create the fine-grained policy such that it enforces all the desired P3P features. If so desired, it is easy to modify the mapping transformation procedure such that it helps a policy writer to write the correct E-P3P policy, given a set of desired P3P outputs. Using such a procedure, the policy writer would specify that certain data should always be kept for `stated-purpose`; which could trigger the generation of the appropriate storage rule with delete obligations. Or, he could specify that the data subject should have access to `all` his data; this would trigger the generation of the appropriate authorization rules for $AccessSubject$.

5.6 Reverse Direction

So far, we discussed how E-P3P practices could be expressed in P3P. We found that the best P3P policies can be achieved if the E-P3P policy writer is aware of some of the targeted P3P policy features: a P3P-aware E-P3P policy editor could even help generating certain E-P3P definitions and rules based on desired P3P outcome. One example was the specification of P3P retention policy in Section 5.5 and the automatic generation of 'store' rules with 'delete' obligations; another example would be the derivation of `data-user` E-P3P rules given a specific value of $AccessMapAll$ value for a required value of `all` for the P3P "ACCESS" element.

Building on that approach, one can also envisage a transformation in the other direction, i.e., from P3P promises to an E-P3P policy enforcing them. In the remainder of this section, we discuss the feasibility of such a transformation.

Starting from a generic P3P practices file, an associated data schema, E-P3P definitions and mapping information, the transformation would generate E-P3P rules. As, of the individual mapping sets in $MapInfo$, only $GenMap$ is obviously direction-specific, we will reuse the $MapInfo$ mapping information.

When transforming from P3P to E-P3P, we take the 'most restrictive' approach: the E-P3P policy should enforce the most strict policy the user or user agent could interpret. E.g., if, for the same data element, the purpose is labeled `opt-out`, and the recipient `opt-in`, then the resulting E-P3P rule should enforce an `opt-in` policy.

The transformation procedure consists of the following steps:

- A transformation of a generic P3P policy to a fine-grained version (the *tuple set*, Section 5.6.1;
- Generation of E-P3P usage rules, Section 5.6.2;
- Generation of E-P3P store rules, Section 5.6.3;
- Generation of E-P3P data subject access rules, Section 5.6.4;

5.6.1 The P3P Fine-grained Tuple Set

First, the P3P practices file is pre-processed to produce a fine-grained *tuple set*. This is necessary to allow optional (opt-in, opt-out) flags to be attached to tuples (data, recipient, purpose) as opposed to recipients or purposes. This step creates a set of tuples (data element, purpose, data user, retention, non-identifiable, optional) for each data element in the statement. The tuple set is the cross-product of the sub-elements of the statement, according to following guidelines:

- The value of optional can be `always`, `opt-in` or `opt-out`:
  - If any of data user or purpose in the tuple was marked `opt-in`, required = `opt-in`. Else, if any of data user or purpose in the tuple was marked `opt-out`, required = `opt-out`. Else, required = `always`.
  - If a data element itself is marked 'optional', all the tuples resulting from this data element have required = `opt-in`.
- If the data-group in the original statement had the flag non-identifiable, then all the resulting tuples are non-identifiable. Else, all the tuples are identifiable.
- Retention for all tuples is copied from the retention value for the statement.

In a second pass, potential conflicts are resolved (this is only necessary if the original P3P policy already contained more than one statement for the same data element): any two tuples with identical element, purpose, data user but non-identical retention or optional values, are merged into one, by using the 'most restrictive' approach.

The P3P tuple set is used to produce usage rules, store rules and data subject access rules as described in the following paragraphs.

5.6.2 The E-P3P Usage Rules

For each tuple in the tuple set, we determine the set of E-P3P data categories, E-P3P purposes, E-P3P recipients and E-P3P actions (using inverse mappings) and create a set of E-P3P rules from their cross-product, according to following guidelines:

- An E-P3P rule is generated only if the tuple set contains all the tuples that would be the result of an E-P3P to P3P cross-product mapping. In other words, a tuple about purpose `current` will only result in a rule about purpose `order` if there is similar tuple about `individual-decision` (see Figure 8): without this condition, we would create a rule about `order` would in its turn allow use for `individual-decision`, which may violate the initial P3P policy if such a statement is not present.
- A data element in an identifiable tuple can only be mapped to an E-P3P data category which is identifiable, and vice versa. Thus, if the same P3P data element is associated with both an identifiable and a non-identifiable E-P3P data category, the non-identifiable tuples will be mapped to rules about the non-identifiable E-P3P category, and vice versa.
- For each tuple, an `opt-in` or `opt-out` results in corresponding rules having opt_in_cond or opt_out_cond.

5.6.3 The E-P3P Store Rules

In a next step, we create the necessary E-P3P store rules with delete obligations. We discard from the full tuples set all the tuples with retention = `none`. For each of the data elements in remaining tuples, and for each of the E-P3P categories associated with them (taking the non-identifiable distinction into account), we create one store rule with data-user a dedicated user in the E-P3P hierarchy (e.g., `/all/internal` in our example) and a purpose which is the union of all the (E-P3P) purposes the data is used for. If at this point an E-P3P category occurs in more than one store rule, the store rules are merged by merging their purposes. We then discard from the tuple set all the tuples with retention = `indefinitely` and retention = `business-practices`, as they would not create any delete obligations. We also discard any tuple with retention = `legal-requirement` and a purpose other than $RetLawMap$, as this is invalid and is considered to be equivalent with `indefinitely`. The remaining tuples all have retention = `stated-purpose`. For each distinct set of (data element, purpose, non-identifiable), we create a delete obligation in the corresponding rule(s) (rules about the appropriate E-P3P categories) according to the retention time in $RetTimeMap$.

Note that, using this procedure, a data element which occurred in the original E-P3P statement with both `stated-purpose` and `indefinitely` retention values, will be deleted according to the `stated-purpose` policy, which is consistent with our 'most restrictive' transformation.

5.6.4 The E-P3P Data Subject Access Rules

In a following step, we generate rules for data subject access. A value of `nonident` does not need any rules; a value of `all`, `contact-and-other`, `ident-contact` or `other-ident` creates rules allowing for data categories in $AccessMapAll$, $AccessMapContactAndOther$, $AccessMapIdentContact$ or $AccessMapOtherIdent$, allowing the actions in $AccessMapActions$ by the data user $AccessSubject$ for purpose $AccessPurpose$; if the appropriate mapping is not defined, an error is generated.

5.6.5 Discussion

The resulting set of E-P3P rules enforces the original P3P practice statement. However, it contains all the authorizations which are possibly allowed by the (more coarse-grained) P3P policy; this may be more than needed or used by the enterprise's business processes.

This is bad practice from a privacy point of view; it is also expensive for the enterprise to have more authorization rules than needed, as it will have to deal with a stack of delete obligations related to unused purposes.

The set of E-P3P rules can be pruned by an administrator with knowledge of business processes and their required data usage. A good definition of business processes and their data usage could automate this process by deleting usage rules (and possible associated store rules with delete obligations). Ideally, the business process information is part of the transformation input, such that generation of usage and store rules is guaranteed to be consistent.

6 Lessons Learned

6.1 The Platform for Privacy Preferences (P3P)

The goal of P3P is to describe the privacy promises of a site in a unambiguous format that can be interpreted by user agents. We feel that this goal is only partially met.

Complexity: P3P is too complex to be easily communicated to end-users. Two complicated aspects are that a statement authorizes the cross-product of all elements in a statement. This gets even worse if some opt-in/opt-out elements are contained in the statement or if cases with unclear semantics are addressed. One way to improve this is to associate opt-in/opt-out with groups of statements. A opt-in choice then consents to a whole lot of permissions. E.g., consenting to an abstract notion of 'direct marketing' would correspond to opting into a list of statements.

Semantics: There exists no clear semantics of P3P, i.e., there is a lot of freedom how user-agents and enterprises interpret a policy. A major ambiguousness are overlapping statements. An example is collecting the same data for the same purposes and recipient under different retention promises or once as identifiable and once as non-identifiable. We feel that there should be at most be one statement for each data/purpose/retention/id or else, there should be a well-define resolution mechanism.

Anonymization Since the same data can be collected anonymized and non-anonymized for the same data user depending on the purpose or even customer-choices, the data cannot be anonymized at collection time. We feel that P3P should be augmented to define what data user can get identifiable versions of data and what data user can get only anonymized ones. The usage by data-users for purposes can then be defined independently.

6.2 The Platform for Enterprise Privacy Practices (E-P3P)

E-P3P aims at formalizing enforcable enterprise-internal privacy policies and interoperability between enterprises. This implies that E-P3P defines a clear syntax and a well-defined semantics.

E-P3P is well suited for enterprise-internal use. For projection into P3P, some additional data has been needed. For interoperability between enterprises, E-P3P still lacks features that consider multiple policies. It currently assumes that all enterprises enforce the same policy. It would be useful to extend E-P3P with concepts that allow to compare multiple policies. This would enable us to implement automated comparisons that derive whether another E-P3P policy corresponds to `ours` (an agent), or `same` (equivalent practices). A first step in this direction would be to extract statements concerning the exchanged data from two E-P3P policies and verify that both policies are equivalent with respect to this data.

7 Conclusions

We have been able to define a transformation between E-P3P Privacy Practices and P3P Privacy Promises. This transformation guarantees that changes of the enterprise-internal privacy practices are reflected by an updated P3P policy. Since the process is automated and E-P3P driven, it may not produce the 'desired' P3P statements like 'we grant data subject access to all its data'. As a consequence, it can be useful to adopt the E-P3P policy with the transformation in mind in order to achieve the desired results.

A major obstacle we had to resolve is the unclear semantics of P3P. In order to describe a sound mapping, we made several assumptions that fill ambiguities in the P3P specification.

We feel that this transformation of policies is a first but important step into the direction of Enterprise Privacy Management, which will enable enterprises to manage privacy like they manage systems security today.

Acknowledgements

We thank Birgit Pfitzmann, Günter Karjoth, Michael Waidner, and Calvin Powers for helpful discussions. This work has been partially funded by the IBM Privacy Institute (see www.research.ibm.com/privacy).

References

[KSW02a] G. Karjoth, M. Schunter, and M. Waidner. From privacy promises to privacy management : A new approach for enforcing privacy throughout an enterprise. In *To appear: ACM New Security Paradigms Workshop*, Virginia Beach VA, September 23-26 2002. ACM Press.

[KSW02b] G. Karjoth, M. Schunter, and M. Waidner. The Platform For Enterprise Privacy Practices – Privacy-enabled Management Of Customer Data. In *Proceedings of the Privacy Enhancing Technologies Conference*, San Francisco, CA, April 14-15 2002.

[TRU] TRUSTe. Privacy Certification. Available at www.truste.com.

[W3C99] W3C. Xml path language (xpath 1.0), 1999. Available at www.w3.org/TR/xpath.

[W3C02a] W3C. A P3P Preference Exchange Language 1.0 (APPEL1.0), 2002. Available at www.w3.org/TR/P3P-preferences.

[W3C02b] W3C. Platform for Privacy Preferences, 2002. Available at www.w3.org/TR/P3P.