To appear in 15th IEEE Computer Security Foundations Workshop, June 24–26, 2002

A Privacy Policy Model for Enterprises

Günter Karjoth and Matthias Schunter
IBM Research
Zurich Research Laboratory
{*gka,mts*}*@zurich.ibm.com*

Abstract

Privacy is an increasing concern in the marketplace. Although enterprises promise sound privacy practices to their customers, there is no technical mechanism to enforce them internally. In this paper, we describe a privacy policy model that protects personal data from privacy violations by means of enforcing enterprise-wide privacy policies. By extending Jajodia et al.*'s Flexible Authorization Framework (FAF) with grantors and obligations, we create a privacy control language that includes user consent, obligations, and distributed administration. Conditions impose restrictions on the use of the collected data, such as modeling guardian consent and options. Access decisions are extended with obligations, which list a set of activities that must be executed together with the access request. Grantors allow to define a separation of duty between the security officer and the privacy officer.*

1 Introduction

The move toward large-scale deployment of electronic commerce is hampered by privacy concerns of potential customers. To appease such concerns, enterprises publish privacy statements that promise fair information practices. Written in natural language or formalized using P3P [14], they are only promises but not necessarily enforced by technical measures [6]. As there is frequently only a vague understanding of the data flows of personal data within the enterprise and no automated controls, enterprises may nevertheless inadvertently violate their own published privacy statements.

These problems are amplified if personal data is used not only by the enterprise that collected the data, but also by secondary users such as partner organizations, census bureaus, and government agencies. These flows of data are complex. Threats to data privacy can come from inside (accidental disclosure, insider curiosity and subornation) as well as from the outside (uncontrolled secondary usage) of each organization. Putting customer information online further increases the risk of exposing private and sensitive information to outsiders.

Access control constrains what a (legitimate) user can do directly, as well as what programs executing on behalf of the users are allowed to do, in order to prevent activities that could lead to a breach of security [13]. The access control decisions are based on an access control policy defined by the security administrator of the system. Classical policies define which subjects (e.g., users or roles) can access which objects (e.g., files, applications) in which mode (e.g., read, write, execute). Privacy control, however, is usually not concerned with individual users. A customer[1] releases his data to the custody of an enterprise while consenting to the set of purposes for which the data may be used. Thus, a typical privacy policy statement such as

> We collect your gender *to customize* our entry catalog pages.

does not authorize a particular subject to access the customer's gender data but anybody (within the enterprise) acting for that purpose.

In general, a "privacy policy" defines what data is collected, for what purpose the data will be used, whether the enterprise provides access to the data, who are the data recipients (beyond the enterprise), how long the data will be retained, and who will be informed in what cases. Based on this specification, an access control system should enforce the policy stated by the enterprise.

A privacy control model should reflect that there are three entities that influence authorizations. The *security administrator* protects the interest of the organization, the customer acting as a *data subject* limits the use and dissemination of his personal data, and the *privacy officer* ensures that the organization correctly implements the data subject's policy as well as the legal privacy regulations.

The privacy statements found today on the Internet are very abstract, and, therefore, it is not possible to determine

[1] For economy of expression, we will assume that the customer is male and all other persons female.

exactly who is authorized to access which objects in what ways. There is the need of a language that is expressive enough to specify privacy rights and obligations that are being promised by privacy statements and mandated by a number of legislatures. The language must have a formal semantics so that the meaning of an authorization requirement stated in that language can be precisely determined. This way, the privacy officer is able to reconcile easily what should be authorized with what is actually authorized.

We focus in this paper on a formal model for authorization management and access control in privacy protecting systems. Taking a systems view of privacy, we elaborate on technical mechanisms to ensure that personal information is used only for authorized purposes. Our model is capable of precisely capturing the meanings of a wide variety of such privacy policies. We then define a privacy language whose semantics is defined with reference to the model. This ensures that every privacy policy has a clear and unambiguous interpretation that is defined without reference to any particular implementation of a privacy protecting system.

Although recent logic-based authorization languages, such as [2, 8] for example, are very expressive and flexible to use, they do not support all required elements of a privacy policy. Therefore we implemented our privacy language within the Flexible Authorization Framework (FAF) [8], enriched with the notions of grantors [2, 15] and obligations [7, 11]. Conditions impose restrictions on the use of the collected data, such as modeling guardian consent and options, or narrowing the set of accessing principals. Access decisions are extended with obligations, which list a set of activities that must be executed together with the access request. Whereas FAF assumes that authorization administration is the task of a single administrator, the *System Security Officer* (SSO), we add the *Chief Privacy Officer* (CPO).

The remainder of this paper is structured as follows. After related work is discussed in Section 2, we describe in Section 3 the basic elements of a privacy policy. These elements are then formalized in Section 4, based on the Authorization Specification Language ASL [8] extended with grantors and obligations. Section 5 elaborates on various administration policies, exploring the relationship between System Security Officer and Chief Privacy Officer. Section 6 illustrates the expressiveness of our model with some example policies from the literature. Section 7 describes our conclusions and future work. Throughout the paper, we use the privacy policy of a virtual online book store called “Borderless Books” as a running example.

2 Related Work

The Platform for Privacy Preferences (P3P) of W3C [14] enables a Web site to state its privacy policy in a standard, machine-readable format. P3P-enabled browsers can retrieve this policy automatically and compare it with the consumer’s set of privacy preferences. A P3P policy is an XML document that describes the data collection practices for a site. It provides a base schema for the data collected and a vocabulary to express purposes, the recipients, and the retention policy. Although it captures common elements of privacy policies, sites may have to provide further explanations in human-readable policies. However, P3P does not provide technical mechanisms to check a given access request against the stated privacy policy.

Fischer-Hübner augmented a task-based access control model with the notion of purpose and consent [5]. Data can be accessed in a controlled manner only by executing a task. A user can access personal data if this access is necessary to perform its current task and the user is authorized to execute this task. Additionally, the task’s purpose must correspond to the purposes for which the personal data was obtained or there has to be consent by the data subjects. A task consists of a set of certified operations representing the set of “necessary accesses” to object classes to perform that task. This work is the first complete model of privacy we are aware of. However, the model does not consider context-dependent access control or obligations.

A language for use-based restrictions that allows one to state under which conditions specific data can be accessed has been developed by Bonatti *et al.* [4]. In their language, a data user is characterized as the triple user, project, and purpose. Projects are named activities registered at the server, for which different users can be subscribed, and which may have one or more purposes. Each user and project is associated with a profile, which captures properties such as name and address or title and sponsor. As they focus on data publishers, their language does not support obligations and consent.

The concept of provisional authorization [7, 10] shares similar objectives with privacy obligations. Added to the access decision, provisions are a kind of annotation that specify necessary actions to be taken. Modeled as a sequence of secondary access requests, they are executed by the user and/or the system under the supervision of the access control system. However, we do not try to describe the semantics of obligations by extending FAF’s logical system with a temporal logic, but simply model obligations as lists of terms returned to the reference monitor denoting actions that have to be executed within the scope of the application.

Concurrently and independently to our work, Bettini *et al* developed a finer and more formalized notion of obligations [3]. They distinguish between actions (*provisions*) that are to be performed before the decision is taken and actions (*obligations*) that will be taken after the decision. These actions are represented as two disjoint sets of predicates, assigned to logic rules. The system implementing the

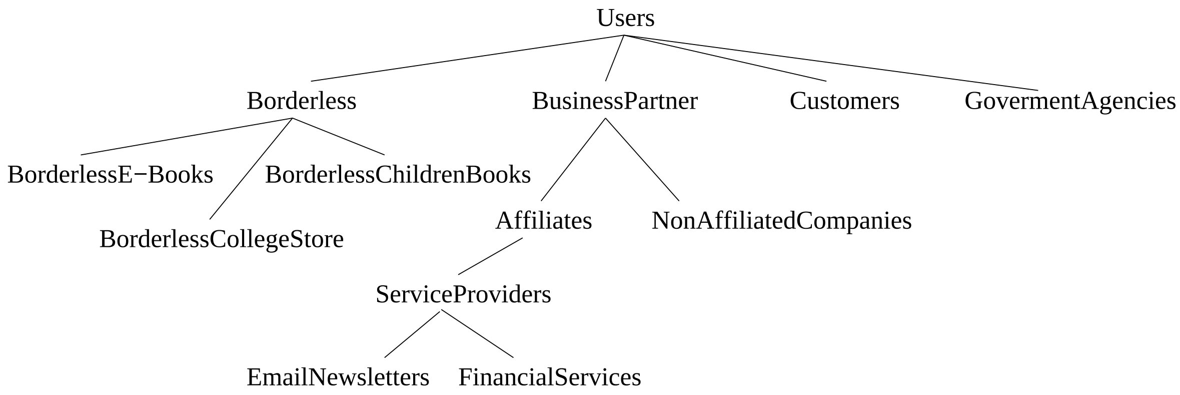

Figure 1. Hierarchy of groups.

policy rules must deduce what actions (if any) may be perform to gain access, and what promises (if any) that must be made after gaining the access. The system also monitors the progress of obligation fulfillment and, in case of failure, take compensatory actions. Provisions are structured and also have an associated weight that allows to select the weakest obligation thus considering semantic relations between them.

3 Elements of a Privacy Policy

In this section, we introduce the elements of our privacy model. We assume that the enterprise runs a Web server, which collects customer data (explicitly via forms and implicitly by analyzing HTTP traffic and cookies) and provides access to services for internal and external users. If data of a customer is stored, then the customer is registered at the Web server and can be authenticated. As a running example, we use an enterprise (Borderless Bookstore) that collects personal data from a data subject (Joe).

3.1 Principals

Information is owned by, updated by, and released to users. We assume that all users are registered at the enterprise, and thus are also system users and can be authenticated accordingly. By "owner" we mean the user who provided the personal data (also known as the "data subject"). The owner is usually a customer of the enterprise. Data users who access data are thus simply called "users". The privacy statement that customers are allowed to access their own data – such as for the correction of inaccurate data – makes an (implicit) reference to the identity of the requester.

Often, users are not explicitly named in privacy policies. The term "we" refers to the enterprise, the terms "business partners" or "others" refer to collections of users. In our model, the concept of groups abstracts from individual users. For example, users can be distinguished to be internal or external to the enterprise. Usually, the customer ("owner") is not an employee of the enterprise. Other external users are members of organizations to whom tasks such as billing have been delegated and thus need access to the data.

Figure 1 depicts an example of group hierarchy. The groups *Borderless*, *BusinessPartners*, *Customers*, and *GovermentAgencies* represent users from different domains and thus with different access rights to personal data. Borderless Bookstore is a family of companies, including the Borderless College Store for example, and these companies might not necessarily share customer information. Within the group hierarchy, a group inherits from its ancestors all their permissions. For example, the permissions of the group *Affiliates* are defined by the permissions of the groups *Users*, *BusinessPartners*, and the permissions of group *Affiliates* itself. In our example, Joe would be a member of group *Customers*; employees of the company providing the email newsletters service would be in group *EmailNewsletters*.

3.2 Data

There are three categories of *personal data* that are distinguished by the level of linkability to its data subject. Whereas personally identifiable information (PII) can be linked to a data subject, depersonalized information can only be linked if one knows additional information (e.g., to whom a pseudonym belongs), whereas anonymized information cannot be linked at all.

In our Borderless Bookstore example, any record that contains Joe's full name or exact address, or a reference[2] to it, is PII. For example, Borderless Bookstore might store

[2] Here we only regard identification numbers. Statistical correlation used on one or more factors specific to an identifiable person's physical, physiological, mental, economic, cultural, or social id category are excluded.

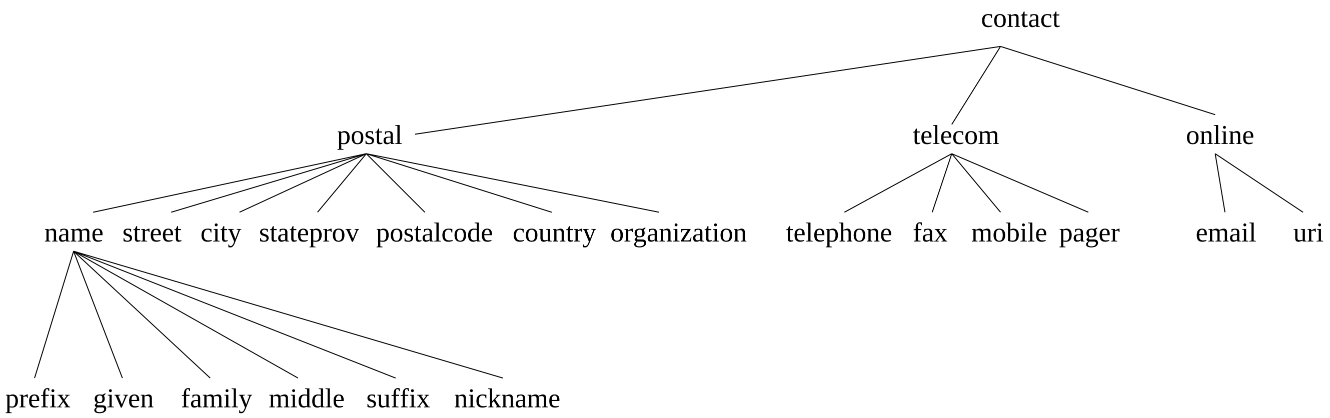

Figure 2. Basic data structures of P3P.

some of the collected data in a dataset accessible via an internal customer number. Although there is no name stored in the dataset, the data is personal data as there is a link from the person to its internal reference. Note that the identification of the individual may use information that is already in the possession of an organization or information that is likely to come into the possession of an organization [9].

Within the enterprise, collected personal data may be stored at different places and in different ways, including as parts of a database. At Borderless, Joe's data is stored in the subscription, billing, and marketing departments. Besides the collected data, there is also personal data that is generated within the organization. In the case of Joe, Borderless keeps a purchase history. Other data record information needed for the evaluation of the privacy policy are date of last access or parental consent.

We introduce *forms* as an abstraction of the actual storage of the data (membership form, purchase form, invoice form). A form is merely a set of fields together with a unique identifier.[3] For example, records in the subscription and billing databases may be interpreted as forms. Forms can be regarded as some sort of meta data, separate from the personal data itself, and used for cataloging and retrieval. Conceptually, a form represents the way a particular personal data has entered the enterprise. For simplification, we assume that a form pertains to only one person.

Types introduce an abstraction on data and thus many human readable privacy policies are expressed in terms of types: "Sale representatives can modify a customer's *purchase record*." This is also the case in our example, where Borderless uses demographics (gender) and billing (purchase history data) information to send monthly book selections to subscriber Joe.

The types of information collected may be structured into contact information (full name, mailing address, email address), financial information (credit/debit card data), medical information (medications, allergy), demographic information (gender, age, race, income level, city/county/state, political party), Internet Protocol (IP) information (IP address, browser type, personal identifiers, passwords) and information collected by or held in cookies.

P3P provides a basic data structure, whose elements are organized into a hierarchy. Figure 2 shows the elements of structure `contact`, which is used to specify contact information. As a data element includes all of the data elements below it in the hierarchy, it is a convenient way to address a group of related data elements. In P3P, data can be associated with categories (physical, online, uniqueid, purchase, etc.). However, there is no means to specify usage restrictions based on categories only.

3.3 Purposes

Any good privacy practice tells the customer how the collected data will be used. To implement this basic privacy principle, we structure the intended use of collected data into categories called purposes. For example, the privacy statement "we use customer contact information (from the registration form) to send you information about our company and/or to give you updates on products and/or services" defines two purposes for which customer contact information will be used:

1. to send company information, and

2. to give updates on products and/or services.

P3P defines a set of purposes including *admin* (web site and system administration), *develop* (research and development), *contact* (contacting visitors for marketing of services

[3] We might use the relational data model to represent forms. Other models would also be possible, an object model for instance, where a class corresponds to the form and the attributes are its fields.

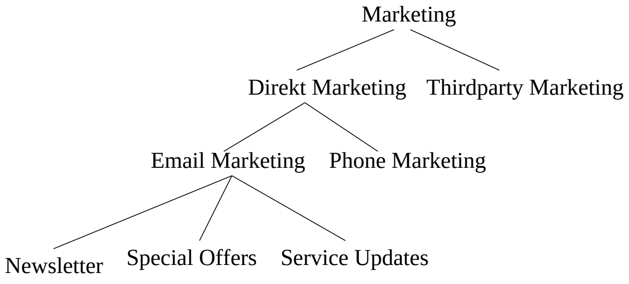

Figure 3. Hierarchy of purposes.

or products), and *telemarketing* (telephone marketing) [14].

In P3P, there are no hierarchical purposes. In many applications, however, there is a natural hierarchy of purposes, based on the familiar principles of generalization and specialization [4]. An example is shown in Fig. 3. The purposes of direct marketing and third-party marketing are specializations of the marketing purpose. A principal assigned to purpose direct marketing (or third-party marketing) will inherit privileges assigned to the more general purpose of marketing.

Purposes are also used to model opt-in/opt-out choices. In P3P, a purpose is of type mandatory (no choice), opt-in (customer may give consent), or opt-out (customer may withdraw consent).

3.4 Actions and Information Sharing

Access modes categorize privacy-relevant operations that an enterprise performs on personal data. Examples are `read` (`write`) that read (write) a data field. In general, we assume that operations do not store their output outside the system. However, privacy policies can also prescribe the dissemination of information. For example, Borderless' privacy policy allows it to disclose Joe's data to a marketing company.

With respect to an access control enforcement system, there is no fundamental difference between an operation that simply retrieves data ("read") and an operation that forwards ("disclose") data to an entity outside of the system. It is beyond the scope of this paper to model the disclosure of data between enterprises, i.e., exporting and importing data with their associated privacy policy from/into a system.

Nevertheless, we can model two variants of disclosure. In the first case, we assume that data can leave the system only by the execution of operations characterized by access mode *disclose*, ensuring that disclosed data is properly exported into another privacy enforcement system that enforces the associated privacy policy. Except by using (digital) rights management systems, technology can do very little to ensure the person or enterprise rightfully receiving the information will handle it according to privacy standards. That depends on ethics and an effective supervisory and legal structure that provides sanctions against detected misuse [12]. In the second case, we regard disclosure as an act of authorization of external users. When Joe authorizes Borderless to disclose data for marketing purposes, there will be an authorization stating that Joe's data can be accessed for purpose "marketing". However, the real disclosure takes only place when group "Marketer" is authorized for purpose "marketing" or an employee of an business partner is added to group "Marketer".

3.5 Conditions

There are privacy statements that not only describe which (type of) data can be accessed for what purpose but also express some conditions to be satisfied before access is granted. In our example policy, Borderless Bookstore may use Joe's data only if it has received the consent of his parent. Thus, the access right depends on whether a certain event has happened previously. Privacy policies may also provide some freedom of choice in allowing the data subject to select whether that data can be used for a certain purpose (such as to provide a service or information). In the registration form, this manifests as a check box. In our running example, Joe gave explicit (opt-in) consent for marketing.

In another policy, the right to access private data may not only depend on the purpose of its use but also on the identity of the requester. For example, Joe might allow usage of his guardian's phone number for purpose *marketing* but only if the requester is his local bookstore agent. Whereas purpose *marketing* would allow anyone authorized to execute an application certified for that purpose to access Joe's personal record, the condition that the requester must be the local agent of the data owner restricts the set of authorized principals to those that act in the (dynamic) role. Note that this policy is different from the policy that "only local agents can access the telephone number of a customers' guardian." In the latter policy, any local agent could access a given customer's guardian's telephone number. In the former policy, however, access is only granted for the customer's local agent. Even a principal name like *Meyer* may be used, thus restricting access to an individual person.

From the examples above, we see that the scope of conditions is broad. It includes attributes of the object (owner, consent, etc.), the user accessing the object, other objects (for example the guardian of the object owner), the history of access to the object, and general attributes such as time or location. In our model, conditions are built over attributes defined in the context of the collected data of a person, i.e., all policy-relevant context information is stored within fields of the form. For example, there might be a field denoting the data subject, the collection date, or the parent's address. To model opt-in/opt-out choices, for each non-mandatory purpose there is a corresponding field in the form whose value reflects the consent of the data subject.

3.6 Obligations

Privacy policies may not only grant access to data but may also make statements about actions that have to be performed. For example, there is the policy that "Parents will be notified of the child's participation in promotions and surveys", or "If you delete your data, we will enforce deletion at all parties to whom the data has been disclosed." Thus, if access is granted, a subsequent action must be executed.

P3P's retention period represents an obligation, too. It cannot be expressed by a history-based authorization, as there has an activity to be performed. Obligations also model situations where the customer's consent must be sought. Other cases are exceptions to consent, where the enterprise is obliged to pass to a third party some information, the customer must still be notified of this information sharing. Such notifications may be delayed, as in the case of law enforcement access, but not omitted [1].

Obligation-based security policies can be enforced by reference monitors if they can be completely resolved inside an atomic execution [11]. Thus, their (future) execution relies on the application that provides a transactional environment. But privacy obligations may be independent from the application logic, and thus might require their own transactional environment. However, if there are compensatory actions [3] for obligations, monitoring of obligations might be sufficient.

In our model, obligations are simply activity names, possibly with parameters, such as *log*, *notify*, and *getConsent*. No (formal) semantics is given, as the behavior of these activities depend on the privacy practices of the enterprise. Also legal regulations leave room for different realizations of obligations. The 1998 Children's Online Privacy Protection Act (COPPA), for example, requires that in many cases Website operators must provide direct notice to parents and must have verified parental consent. However, the required method of consent varies based on how the collector uses the child's personal data, ranging from simple email to getting a signed form from the parent via postal mail or facsimile.

4 Formalization of a Privacy Policy

In this section, we formalize the privacy policy model described in the previous section, using the logical framework of the Authorization Specification Language ASL [8], extended with the notion of grantors [2, 15] and obligations [7].

4.1 Data System

The data system of our privacy model consists of users/groups, the data they are accessing, together with the purposes they act for, and the access modes they use. In particular, we define data items or groups of data items via the triple $OTH = \langle Obj, T, \leq_{OT} \rangle$ where Obj is a set of identifiers of fields in a form, T is a set of types (or set of objects), and $\leq_{OT}$ is a type hierarchy.

At Borderless Bookstores, there are forms for subscription, billing, and marketing. Every form is stored under a unique index id. Term /subscription denotes all subscription forms and /subscription[id] denotes the subscription form stored under index id. Correspondingly, term /subscription/account denotes the field account in all stored subscription forms and term /subscription[id]/account denotes field account in the subscription form stored under index id.

Users and groups are defined via a hierarchy $UGH = \langle U, G, \leq_{UG} \rangle$ where U is a set of user identifiers, and G is a set of group identifiers, and $\leq_{UG}$ is the group hierarchy where $x \leq_{UG} y$ iff all members of group x are also members of group y. In the same way, purposes are defined as a hierarchy $PH = \langle \emptyset, P, \leq_P \rangle$ where P is a set of purposes and $x \leq_P y$ iff x is a specialization of y. A purpose is a specialization of another purpose if it refers to more specialized usages.[4] Hierarchies OTH, UGH, and PH are disjoint.

There is a set A of actions or authorization modes. In our example, we use actions such as read, write, delete, disclose, and activate.

The last element of the data system is a set Rel of relationships, which are defined on different elements of the data system. Predicate $\mathtt{owner}(o, u)$ associates a unique user u with object o, the *owner* of o. The *owner* predicate abstracts from the specific implementation of the "data subject" concept. For instance, in an ACL-based system ownership might be expressed by a special permission. However, the data system itself may already provide with each object a link to the associated "owner".

We introduce the following privacy-specific relations on users and objects. Predicate $\mathtt{isMinor}(u)$ determines whether user u is a minor. The definition may depend on the citizenship of the owner or on the country where the Web server is located. Predicate $\mathtt{isGuardian}(u, u')$ associates a unique user u' with user u, the *guardian* of u. For convenience, we write $\mathtt{minor}(o) \equiv \mathtt{owner}(o, u) \wedge \mathtt{isMinor}(u)$ and $\mathtt{guardian}(o, u) \equiv \mathtt{owner}(o, u') \wedge \mathtt{isMinor}(u') \wedge$

[4] For instance, Email-marketing and Phone-marketing can be both seen as a specialization of Marketing (see Figure 3).

$\texttt{isGuardian}(u', u)$ to denote whether the owner of object o is a minor and who the guardian is.

Predicate $\texttt{consent}(o, p, u)$ defines whether user u has consented that object o can be processed for purpose p. We write $\texttt{opt-in}(o, p) \equiv \texttt{owner}(o, u) \wedge \texttt{consent}(o, p, u)$ as shorthand to denote the consent given by the data subject. The term "opt-out" means that *unless and until* the customer informs the enterprise that he does not want his data be used for that purpose, the enterprise it free to do so. The "opt-in" provision, on the other hand, says that the enterprise cannot use the data *unless* the customer has consented. Accordingly, $\texttt{opt-out}(o, p) \equiv \neg\texttt{opt-in}(o, p)$ is the dual to *opt-in*. Predicate $\texttt{lastAccess}(o, time)$ relates the time (of type *duration*) elapsed since object o has been accessed last.

We model obligations as lists of terms, being activity names and possibly parameterized. Such activities must be performed within the scope of the execution of the requested and granted operation. We fix the set of *obligations* C to contain activity notify:u, which sends a notification to user u, and activity anonymize:o, which anonymizes object o. The empty list of obligations $[\,]$ is always fulfilled. Let c range over obligations.

4.2 Authorizations

The set AS of *authorization subjects* consists of users, processes, groups, and purposes. Likewise, the set AO of *authorization objects* consists of objects, types, and purposes – the latter are included in objects as they can be assigned to subjects. The set $SA = \{+a, \ -a \mid a \in A\}$ denotes *signed actions*, actions that are either authorized (+) or denied (-).

There is a subject hierarchy ASH obtained by placing the graphs of UGH and PH side by side. Likewise, the object hierarchy OSH is obtained by placing the OTH and the inverse of PH side by side. With the inverse of PH, we express the assumption that if some action is allowed for a subpurpose of purpose p, it is also allowed for purpose p. Using the purpose hierarchy expressed in Figure 3, for example, we say if data can be collected for purpose DirectMarketing then it is also allowed for purpose Marketing (but not for purpose ThirdpartyMarketing).

An *authorization* is a quintuple of the form

$$\langle o, s, \langle sign\rangle a, c, g\rangle,$$

where $o \in AO$, $s, g \in AS$, $c \in C$, $a \in A$, and "sign" is either "+" or "-". A positive authorization $\langle o, s, +a, c, g\rangle$ states that subject g authorizes subject s to perform action a on object o provided obligation c will become true. We restrict negative authorizations to contain only default obligation $[\,]$.

Examples for authorizations relative to form /subscription given by the Chief Privacy Officer (cpo) are:

- $\langle \texttt{contact/homeAddress}, \texttt{promotion}, +\texttt{read}, [\,], \texttt{cpo}\rangle$.
  We ask you for your address to send you promotional material.
- $\langle \texttt{demographics/gender}, \texttt{thirdParty}, -\texttt{read}, [\,], \texttt{cpo}\rangle$.
  No outside party will have access to your income figures.
- $\langle o, \texttt{government}, +\texttt{disclose}, [], \texttt{cpo}\rangle$.
  We share user information with governmental authorities when legally required to do so.
- $\langle o, \texttt{statistics}, +\texttt{disclose}, [\texttt{anonymize:}o], \texttt{cpo}\rangle$.
  We disclose user information for statistical purposes only in anonymized form.

Negative authorizations are very appropriate to express the natural language policy above in a concise way. Groups thirdParty and government represent user domains outside of the enterprise (see Figure 1). In particular, group government abstracts from the way cooperation with governmental authorities for law enforcement is authenticated in practice.

4.3 Authorization Specification Language

ASL is a logical language that contains different types of rules that are inserted by the security administrator, representing direct authorizations (cando) or authorizations derived by the system using logical rules of inference: dercando rules describe the propagation of information, do rules define conflict resolution strategies and error rules define integrity constraints.

ASL includes the relations Rel of the data system and additional predicates, called hie-predicates *dirin* and *in* that capture the direct and indirect membership relationship between subjects, and predicate *typeof* that captures the grouping relationship between objects. Predicate done represents events that happened in the past: if $\texttt{done}(o, u, a)$ is true then user u has executed action a on object o.

Let a term be a constant or a variable instantiated with any value of a given set of the data system. If p is an n-ary predicate symbol, then $p(t_1, \ldots, t_n)$ is an atom. A literal is an atom or the negation of an atom. An *authorization rule* is a rule of the form

$$\texttt{cando}(o, s, \langle sign\rangle a, c, g) \leftarrow L_1 \wedge \ldots \wedge L_n.$$

where $o \in AO$, $s, g \in AS$, $a \in A$, $c \in C$, "sign" is either "+" or "-", $n \geq 0$, and $L_1, \ldots, L_n$ are done, hie-, or rel-literals. The intuition underlying an authorization rule is that the authorization $\langle o, s, \textit{(sign)}a, c, g\rangle$ is added if the literals evaluate to true.

Examples for direct authorization rules are shown in Figure 4. In the first rule, members of the group businesspartners can read the customer's email address if the customer has consented. The second rule also makes the access

- $\texttt{cando}(o, \mathsf{business\text{-}partners}, +\mathsf{read}, [\,], \texttt{cpo}) \leftarrow \texttt{opt-in}(o, \texttt{thirdparty-marketing})$.
  We will give you the option of receiving e-mail, telephone calls, or written service from our business partners.

- $\texttt{cando}(\texttt{/subscription}[id]\texttt{/gender}, \mathsf{Email\text{-}marketing}, +\mathsf{read}, [\,], \texttt{cpo}) \leftarrow \neg\texttt{opt-out}(id, \texttt{Email-marketing})$.
  Users may opt-out of receiving future mailings.

- $\texttt{cando}(\texttt{/subscription}[id]\texttt{/account}, u, +\mathsf{write}, [\,], \texttt{cpo}) \leftarrow \texttt{owner}(id, u)$.
  Consumers can change all of their personal account information including their address, telephone number, email address, password as well as their privacy settings.

- $\texttt{cando}(\texttt{/subscription}[id]\texttt{/account}), \mathsf{contest}, +\mathsf{read}, [\,], \texttt{cpo}) \leftarrow \neg\texttt{minor}(id)$.
  We will use your account information while conducting contests if your are not a minor.

- $\texttt{cando}(o, u, +\mathsf{delete}, [\,], \texttt{cpo}) \leftarrow \texttt{guardian}(o, u)$.
  We allow parents to remove from our database at any time the information collected about their child.

- $\texttt{cando}(o, s, +\mathsf{write}, [\,], \texttt{cpo}) \leftarrow \texttt{guardian}(o, u) \wedge \texttt{consent}(id, \mathsf{approval}, u)$.
  No information should be submitted to or posted at Borderless by persons under 18 years of age without the consent of their parent or guardian.

- $\texttt{cando}(o, \mathsf{Borderless}, +\mathsf{create}, [\mathsf{notify}{:}u], \texttt{cpo}) \leftarrow \texttt{guardian}(o, u)$.
  When we receive information from minors, we notify their parents.

- $\texttt{cando}(o, s, -\mathsf{read}, [\,], \texttt{cpo}) \leftarrow \mathsf{lastAccess}(o, d) \wedge d > \texttt{P1Y}$.
  Personal data becomes inaccessible if not used for more than one year.

Figure 4. Example authorizations.

to gender information conditional on the consent of the data subject. The third rule allows users to change their own personal data stored in the user account. The next four rules are authorizations regulating access to personal data of minors. The last but one rule gives an example where an obligation has to be executed by the system. Finally, the last rule prevents access to data after one year of user inactivity.

Note that the last policy of Figure 4 is only an approximation of the statement that "personal data must be erased at some date", assuming that inaccessible data would be eventually garbage-collected. Modeling erasure of data would require a more sophisticated obligation management using conditional obligations [3].

Derived Authorizations. In the case that no explicit authorization exists for a given request, derivation rules define how authorizations propagate "downwards" in the AOH and ASH hierarchies. A *derivation rule*

$$\texttt{dercando}(o, s, \langle \textit{sign} \rangle a, c, g) \leftarrow L_1 \wedge \ldots \wedge L_n.$$

is like an authorization rule, except that literals can also include cando and dercando rules, the latter only in positive form.

In our system, we use the next three rules to specify how rights flow along the subject and object hierarchies.

$$\begin{aligned}
&\texttt{dercando}(o, s, a, c, g) \\
&\qquad \leftarrow \texttt{cando}(o', s, a, c, g) \wedge \texttt{typeof}(o, o'). \\
&\texttt{dercando}(o, u, a, c, g) \\
&\qquad \leftarrow \texttt{cando}(o, u', a, c, g) \wedge \texttt{in}(u, u'). \\
&\texttt{dercando}(o, p, a, c, g) \\
&\qquad \leftarrow \texttt{cando}(o, p', a, c, g) \wedge \texttt{in}(p', p).
\end{aligned}$$

Decision Rules. A *positive decision rule*

$$\texttt{do}(o, u, +a, c) \leftarrow L_1 \wedge \ldots \wedge L_n.$$

is like a derivation rule without grantor, except that every variable that appears in any of the L_i's also appears in the head of the rule.

The two decision rules below enforce consistency and completeness for the authorizations derived from a single administrator. For conflict resolution, we employ a "denials take precedence" strategy under a closed world assumption.

$$\begin{aligned}
&\texttt{do}(o, u, +a, c) \leftarrow \\
&\quad \texttt{dercando}(o, u, +a, c, g) \wedge \neg\texttt{dercando}(o, u, -a, [\,], g).
\end{aligned}$$

The closure rule says that the only authorizations granted are those explicitly derived after conflict resolution [8].

$$\texttt{do}(o, u, -a, [\,]) \leftarrow \neg\texttt{do}(o, u, +a, c).$$

Up to now we only presented privacy-specific authorizations defined by the Chief Privacy Officer. In the next section, we discuss additional decision rules establishing specific administration policies that regulate competences between privacy and security administrators.

Integrity Rules. Security administrators can define constraints that must hold for the authorization specifications or the actual access execution. An *integrity rule* is a rule of the form

$$\texttt{error} \leftarrow L_1 \wedge \ldots \wedge L_n.$$

where $L_1, \ldots, L_n$ are cando, dercando, done, do, hie-, or rel- literals. Integrity rules check the authorization specification at run-time, thus taking dynamic aspects into account.

5 Deriving Authorizations from different Administration Domains

In this section, we examine different approaches to jointly administrate privacy and access control. Using the grantor element of an authorization rule, we define several administration policies, which vary in the separation of duty between access control and privacy control.

Centralized Administration. In our first scenario, we assume that the privacy officer directly assigns purposes to users, expressed by predicate $\texttt{active}(u, p)$, which can then be activated like roles [8].

$$\begin{aligned}&\texttt{dercando}(o, u, +a, c, \texttt{cpo})\\ &\quad\leftarrow \texttt{active}(u, p) \wedge \texttt{cando}(o, p, +a, c, \texttt{cpo}).\end{aligned}$$

Here purposes are – like roles – named collection of privileges. However, there is a subtle difference. Whereas roles identify the tasks that users need to execute to perform organizational activities, purposes identify the tasks for which data can be used. Authorization to a role/purpose facilitates access to the information associated with the role/purpose.

In addition, we may not only want to “authorize” users but also to “certify” tasks.

$$\begin{aligned}&\texttt{dercando}(o, s, +a, c, \texttt{cpo}) \leftarrow\\ &\quad\texttt{dercando}(o, p, +a, c, \texttt{cpo}) \wedge\\ &\quad\texttt{dercando}(p, s, +\texttt{activate}, c, \texttt{cpo}).\end{aligned}$$

If action a can be performed on object o for purpose p under obligation c and subject s is certified for that purpose then subject s can perform action a on object o.

Distributed Administration. In a privacy-aware enterprise, there is a Chief Privacy Officer, who is responsible for the development and implementation of the enterprise's privacy policies and procedures. To be independent from general IT business, the Chief Privacy Officer is usually a high-level management or officer position. A System Security Officer, on the other hand, is responsible for data security within the enterprise.

To distribute authorization between System Security Officer and Chief Privacy Officer while guaranteeing the desired separation of duty, we allow each to manage a certain subset of the authorizations. The System Security Officer authorizes tasks and users to perform operations on objects. An example is the authorization

$$\texttt{cando}(\texttt{order-tracking}, \texttt{jack}, +\texttt{x}, [\,], \texttt{sso}) \leftarrow .$$

that enables user jack to execute task order-tracking. The Chief Privacy Officer on the other hand authorizes purposes to perform operations on data. This reflects the privacy policy that has been consented by the data subject. An example is the rule

$$\texttt{cando}(o, \texttt{billing}, +\texttt{read}, c, \texttt{cpo}) \leftarrow .$$

that enables purpose billing to read object o under obligation c. In addition, the Chief Privacy Officer certifies tasks to perform certain purposes, either expressed by the predicate $\texttt{certified}(t, p)$ or by an explicit authorization $\texttt{cando}(p, t, +\texttt{certified}, [\,], \texttt{cpo}) \leftarrow$. Using the first alternative, this intuitive separation of duty is formalized by the following rule:

$$\begin{aligned}&\texttt{do}(o, u, +a, c) \leftarrow\\ &\quad\texttt{dercando}(o, p, +a, c, \texttt{cpo}) \wedge \texttt{certified}(t, p) \wedge\\ &\quad\texttt{dercando}(t, u, +\texttt{x}, [\,], \texttt{sso}) \wedge \texttt{currentTask}(u, t).\end{aligned}$$

Above rule defines that the access request of an user u can be granted if the user's current task t has been certified by the Chief Privacy Officer for purpose p and that this purpose is authorized by the Chief Privacy Officer to perform the desired access, and that the System Security Officer has authorized the user u to execute task t.

$$\begin{aligned}&\texttt{error} \leftarrow \texttt{in}(u, \texttt{sso}) \wedge \texttt{in}(u, \texttt{cpo}).\\ &\texttt{error} \leftarrow \texttt{dercando}(o, p, +a, c, \texttt{sso}).\\ &\texttt{error} \leftarrow \texttt{dercando}(o, u, +a, c, \texttt{cpo}).\end{aligned}$$

By adding above consistency rules, we also mandate that the roles System Security Officer and Chief Privacy Officer are in fact played by different users, that only the System Security Officer authorizes purposes to access objects, and that only the System Security Officer assigns purposes to tasks.

Precedence of authorizations. The previous example shows how policies given by two officers are combined: access is granted if a positive authorization can be derived from both rule sets administrated by the Chief Privacy Officer and the System Security Officer. But there might be cases when officers define conflicting policies. The simplest approach is to flag these situations:

$$\begin{aligned}\texttt{error} \leftarrow & \texttt{dercando}(o, u, +a, c, \texttt{cpo}) \wedge \\ & \texttt{dercando}(o, u, -a, [\,], \texttt{sso}).\end{aligned}$$

However, it is also possible to allow one officer to overwrite the policy of the other officer. For example, the System Security Officer should not be able to reject an access request granted explicitly by the Chief Privacy Officer to the owner, as for example in the policy where the data subject is allowed to read and/or change its personal data.

$$\begin{aligned}\texttt{do}(o, u, +a, c) \leftarrow \texttt{dercando}(o, u, -a, [\,], \texttt{sso}) \wedge \\ \texttt{dercando}(o, u, +a, c, \texttt{cpo}) \wedge \texttt{owner}(o, u).\end{aligned}$$

Note that the formulation of above policy is only possible because data subjects are also system users in our model.

6 Formalized Example Policies

To illustrate the expressiveness of our language, we specify several published privacy policies.

Task-based Privacy. In [5], Fischer-Hübner states the following privacy policy:

> A subject may only have access to personal data if this access is necessary to perform its current task, and only if the subject is authorized to perform this task. The subject may only access data in a controlled manner by performing a (well-formed and certified) transformation procedure, for which the subject's current task is authorized. In addition, the purpose of its current task must correspond to the purposes for which the personal data was obtained or consent must be given by the data subjects.

Let "diagnosing" be a process ("task") and "treatment" be a purpose in a hospital system.

$$\texttt{cando}(\texttt{medications}, \texttt{treatment}, +\texttt{read}, [], \texttt{cpo}) \leftarrow . \quad (1)$$

$$\texttt{cando}(\texttt{treatment}, \texttt{diagnosing}, +\texttt{activate}, [], \texttt{cpo}) \leftarrow . \quad (2)$$

$$\texttt{cando}(\texttt{diagnosing}, \texttt{Joe}, +\texttt{execute}, [], \texttt{sso}) \leftarrow . \quad (3)$$

Rule 1 states that data of type medications can be read for purpose treatment. Rule 2 states that task diagnosing can act for purpose treatment. Rule 3 states that user Joe is authorized to perform task diagnosing.

The Privacy Officer provided authorization rules 1–2; the security officer gave authorization 3. In addition, the patient implicitly sanctioned authorization rule 1 when he accepted the enterprise's privacy policy.

$$\begin{aligned}\texttt{error} \leftarrow \texttt{certified}(\texttt{diagnosing}, p) \wedge \\ \texttt{certified}(\texttt{diagnosing}, p') \wedge p \neq p'.\end{aligned} \quad (4)$$

By adding rule 4, the Privacy Officer strengthens the privacy policy stating that the above task serves exactly one purpose.

Information Flow. To prevent authorized users of "misusing" information, it is necessary to control the ability of an authorized user to copy a data item. Here we apply a simple information flow policy within a clinical information systems expressed by Anderson [1]:

> Where two records with different access control lists have the same owner, then the only information flow permissible without further consent is from the less to the more sensitive record. This means that information derived from record A may be appended to record B if and only if B's access control list is contained in A's.

Below rule derives a denial for a subject to write to object o if the subject has read an object o' before and that object o' is more sensitive than object o.

$$\begin{aligned}\texttt{do}(o, u, -\texttt{write}, [], g) \leftarrow \texttt{done}(o', u, \texttt{read}) \wedge \\ \texttt{owner}(o, u) \wedge \texttt{owner}(o', u) \wedge \\ \texttt{do}(o, u', +a, c') \wedge \neg\texttt{dercando}(o', u', +a, c, g).\end{aligned}$$

We say that o's access control list is contained in o''s if all authorizations for o also hold for o'; i.e., there is no user u' who can access o but not o'. As this "inclusion" relation occurs negated in the last two literals of above rule, it triggers denial of access.

7 Conclusions

In this paper we presented a privacy policy model for enterprises that can serve as the basis for an internal access control system to handle received data in accordance with privacy standards. Thus, the data subject providing his/her personal data has the assurance that the enterprise receiving the information will handle it according to the stated privacy policy. The enterprise as well can verify that its business practices are not in conflict with the privacy policies posted on its Web site, which are usually considered a binding contract between the site owner and the people visiting the site.

This privacy policy model is the first that combines user consent, obligations, and distributed administration. Conditions impose restrictions on the use of the collected data, such as modeling guardian consent and options, or narrowing the set of accessing principals. Access decisions are extended with obligations, which list a set of activities that must be executed together with the access request. We showed how “real world” privacy statements can be expressed in our authorization language. They are expressive enough to accommodate an enterprise’s own privacy practices, possibly parameterized by the local country laws.

Although our privacy policy model is preliminary, it provides a sound basis for privacy protection officers or consumer associations to design “popular polices” conforming with strict laws whose implementation within an enterprise can be certified in the scope of “seal programs” of trusted third parties. Financial and health-care institutions, in particular, have stringent and complex privacy policies in place. We are currently investigating how well our model is suited to describe and enforce these policies.

More work is needed for a better description of the sharing of personal data. In the present model, personal data is shared by giving access to people outside the enterprise. But data is not physically copied. However, when personal data is transferred to another organization, a privacy policy should be attached to it that precisely reflects the access limitations for that data derived from the overall enterprise privacy policy.

The presented model is too simplistic in the way it deals with “meta data” assuming that everything is stored in a single form. This may be the case for a newly designed application where all entries of personal data are known. However, this is certainly not the case for a larger company with multiple DBMS systems and dozens of applications. Nevertheless, the specification of privacy policies is quite independent of the IT infrastructure used to store and process personal data, as most individual data systems are instantiations of the general definition used [8]. A policy specification, which is decoupled from data systems such as relational databases or object-oriented systems, also allows personal data to be encapsulated with its associated privacy policy when passed to other companies, such that the access control system of the receiving company can interpret and enforce the policy.

Acknowledgments

The authors would like to thank Paul Ashley, Jan Camenisch, Birgit Pfitzmann, Calvin Powers and Michael Waidner for the many critical observations and fruitful discussions on this work. The members of IBM’s Enterprise Privacy Architecture team were a valuable source not only of stringent requirements for but also solid expertise on data privacy.

References

[1] R. Anderson. A security policy model for clinical information systems. In *IEEE Symposium on Research in Security and Privacy*, pages 30–43. IEEE Computer Society, 1996.

[2] E. Bertino, B. Catania, E. Ferrari, and P. Perlasca. A logical framework for reasoning about access control models. In *6th ACM Symposium on Access Control Models and Technologies (SACMAT 2001)*, pages 41–52. ACM Press, 2001.

[3] C. Bettini, S. Jajodia, X. Wang, and D. Wijesekera. Obligation monitoring in policy management. In *3rd International Workshop on Policies for Distributed Systems and Networks (POLICY 2002)*. IEEE Computer Society, 2002.

[4] P. Bonatti, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. An access control system for data archives. In *16th IFIP-TC11 International Conference on Information Security*. 2001.

[5] S. Fischer-Hübner. *IT-Security and Privacy – Design and Use of Privacy-Enhancing Security Mechanisms*. Lecture Notes in Computer Science 1958. Springer, 2001.

[6] R. Grimm and A. Rossnagel. Can P3P help to protect privacy worldwide? In *International Multimedia Conference*, pages 157–160. ACM Press, 2000.

[7] S. Jajodia, M. Kudo, and V. S. Subrahmanian. Provisional authorization. In A. Ghosh, editor, *E-commerce Security and Privacy*, pages 133–159. Kluwer Academic Publishers, 2001. Also published in Workshop on Security and Privacy in E-Commerce (WSPEC), 2000.

[8] S. Jajodia, P. Samarati, M. L. Sapino, and V. Subrahmanian. Flexible support for multiple access control policies. *ACM Trans. Database Syst.*, 26(2):214–260, June 2001.

[9] S. Johnston. The impact of privacy and data protection legislation on the sharing of intrusion detection information. In W. Lee, L. Mé, and A. Wespi, editors, *RAID 2001*, Lecture Notes in Computer Science 2212, pages 150–171. Springer, 2001.

[10] M. Kudo and S. Hada. XML document security based on provisional authorizations. In *7th ACM Conference on Computer and Communications Security*, pages 87–96. ACM Press, 2000.

[11] C. N. Ribeiro, A. Zúquete, P. Ferreira, and P. Guedes. SPL: An access control language for security policies with complex constraints. In *Network and Distributed System Security Symposium (NDSS’01)*, pages 89–107, 2001.

[12] T.C. Rindfleisch. Privacy, Information Technology, and Health Care. *Communications of the ACM*, 40(8):93–100, 1997.

[13] R. Sandhu and P. Samarati. Access control: Principles and practice. *IEEE Communications MAGAZINE*, 32(9):40–48, 1994.

[14] W3C. The platform for privacy preferences 1.0 (P3P1.0) specification, Jan. 2002. W3C Proposed Recommendation, `http://www.w3.org/TR/P3P`.

[15] H. Wedde and M. Lischka. Modular authorization. In *6th ACM Symposium on Access Control Models and Technologies (SACMAT 2001)*, pages 97–105. ACM Press, 2001.