To appear in *2nd Workshop on Privacy Enhancing Technologies*, Lecture Notes in Computer Science. Springer Verlag, 2002.

Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data

Günter Karjoth, Matthias Schunter, and Michael Waidner

IBM Research, Zurich Research Laboratory,
Säumerstrasse 4, 8803 Rüschlikon, Switzerland
{gka,mts,wmi}@zurich.ibm.com

Abstract. Enterprises collect a large amount of personal data about their customers. Even though enterprises promise privacy to their customers using privacy statements or P3P, there is no methodology to enforce these promises throughout and across multiple enterprises. This article describes the Platform for Enterprise Privacy Practices (E-P3P), which defines technology for privacy-enabled management and exchange of customer data. Its comprehensive privacy-specific access control language expresses restrictions on the access to personal data, possibly shared between multiple enterprises. E-P3P separates the enterprise-specific deployment policy from the privacy policy that covers the complete life cycle of collected data. E-P3P introduces a viable separation of duty between the three "administrators" of a privacy system: The privacy officer designs and deploys privacy policies, the security officer designs access control policies, and the customers can give consent while selecting opt-in and opt-out choices.

1 Introduction

Consumer privacy is a growing concern in the marketplace. Whereas privacy concerns are most prominent in the context of e-commerce, they are increasing for traditional transactions as well. Some enterprises are aware of these problems and of the market share they might lose unless they implement proper privacy practices. As a consequence enterprises publish privacy statements that promise fair information practices. Written in natural language or formalized using P3P [11], they merely constitute privacy promises and are not necessarily backed up by technological means.

In this article, we describe the Platform for Enterprise Privacy Practices (E-P3P). E-P3P defines technology that enables an enterprise to enforce the privacy promises made to its customers. It solves some of the most prominent privacy issues of enterprises that collect data from their customers:

- Enterprises store a variety of personally identifiable information (PII or *personal data* for short). Larger enterprises may not even know what types of personal data are collected and where it is stored.
- Enterprises may not know the consent a customer has given nor the legal regulations that apply to a specific customer record.
- Enterprises exchange customer data. Enterprises that process or store data collected by another enterprise are unable to enforce privacy consistently on behalf of the collecting enterprise.

Whenever an enterprise collects, stores, or processes personal information, E-P3P can be used to ensure that the data flows and usage practices of an enterprise comply with the privacy statement of that enterprise.[1] E-P3P can be used in the following areas:

- *Formalized Privacy Policies:* E-P3P enables an enterprise to formalize a privacy policy into a machine-readable language that can be enforced automatically. The natural text version is inspected by the customers whereas the machine-readable version is used for enforcement within the enterprise.
- *Formalized Policy Options:* An E-P3P privacy policy can identify opt-in as well as opt-out choices or options that depend on the collected data (e.g., whether the given data pertains to a child). These options enable a company to use a limited number of policies while still providing freedom of choice to its customers.
- *Customer Consent Management:* A privacy policy can be regarded as a contract between the individual customer and an enterprise. As a consequence, a customer needs to authorize the applicable policy as well as any applicable opt-in and opt-out choices that the policy offers. This requires recording of consent on a per-customer basis.
- *Policy Enforcement:* Given collected personal data and its policy, the policy needs to be enforced. Policy enforcement covers several cooperating enterprises if personal data is exchanged among them. The core technology is a scheme for privacy-enabling access control that allows only actions that are authorized by the applicable privacy policy. Besides granting or denying access, privacy obligations have to be enforced as well (such as "we delete collected data if consent is not given within 15 days").
- *Compliance Audit:* The handling of personal data should comply with the privacy policy in an auditable way. This enables a privacy officer to verify that the data was handled properly.

Note that this is only the technical core of privacy-enabled customer data management.

Another important building block is to provide additional customer privacy services. Customers should be enabled to inspect and update the data and usage logs stored about them. In addition, an enterprise may offer the option to delete the personal data. Ideally, customers should retain maximum control over their data. Once a privacy-management scheme has been implemented, it needs to be audited by external parties that are trusted by the customers. Together with resulting privacy seals, this can increase the trust of the consumers. Customer privacy requires secure systems. As a consequence, enterprises must implement continuous business processes to keep their systems secure. Owing to space restrictions we will not elaborate on these services in this article.

The rest of this paper is structured as follows: In Section 2 we describe existing work related to privacy-enabled data management. In Section 3, we describe the E-P3P scheme and architecture for privacy-enabled customer data management. In Section 4, we take a closer look at the language for formalizing privacy policies as well as at the logic for evaluating privacy policies. We conclude in Section 5. A privacy policy for a hypothetical online bookstore called Borderless Books is given in Appendix A.

[1] Note that our scheme only protects against systematic privacy violations within the system. For example, it cannot prevent misuse by an employee with legitimate access.

2 Related Work

The Platform for Privacy Preferences (P3P) standard of W3C [11] enables a Web site to declare what kind of data is collected and how this data will be used. A P3P policy may contain the purposes, the recipients, the retention policy, and a textual explanation of why this data is needed. P3P defines standardized categories for each kind of information included in a policy. Compared to P3P, our model defines the privacy practices that are implemented *within* an enterprise. As this depends on internal structures of the company, it results in more detailed policies that can be enforced and audited automatically. Note that our policies can use P3P-compatible terminology to easily check that the P3P promises correspond to the enterprise-internal policies.

Current access control systems [12] only check whether a user is allowed to perform an action on an object. In [5], Fischer-Hübner augmented a task-based access control model with the notion of purpose and consent. Data can only be accessed in a controlled manner by executing a task. A user can access personal data if this access is necessary to perform the current task and the user is authorized to execute this task. In addition, 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. This work is the first complete model of privacy we are aware of. However, the model does not consider context-dependent access control nor obligations and is restricted to a single enterprise.

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.* [3]. 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. Conditions are used to define constraints that must be satisfied for the request to be granted.

Mandatory and discretionary access controls do not handle environments in which the originators of documents retain control over them after those documents have been disseminated. In [10], McCollum *et al.* define Owner-Retained Access Control (ORAC) that provides a stringent, label-based alternative to discretionary access control. This is of interest for user communities where the original owners of data need to retain right to the data as it propagates through copying, merging, or being read by a subject that may later write the data into other objects. The *originator-controlled access control* (ORGCON) policy [1] limits the authority of recipients of information to use or copy it.

The concept of provisional authorization [7, 9] 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.

Our concept of bundling data and policy is similar to the concept used in XACL [6]. An XACL document contains an access control policy for a particular XML document as well as the document to which the access shall be restricted.

A data format for disclosing customer profile data between enterprises has been defined by CPexchange [4]. CPexchange uses P3P-like privacy statements to define the policy accompanying the disclosed profile.

3 Platform for Enterprise Privacy Practices

The Platform for Enterprise Privacy Practices (E-P3P) is a scheme for privacy-enabled management of customer data. Its core is an authorization scheme that defines how collected data may be used.

3.1 Application Model and Prerequisites

In E-P3P, an enterprise runs *legacy applications* that use collected data. Each application can perform certain *tasks*. For example, a "customer relationship management system (CRM)" application may perform the tasks "create new customer record" or "update existing customer record".

Enterprise privacy policies reflect the authorized flow and usage of personal information within an enterprise. As a consequence, the flows and usages have to be identified in order to use the E-P3P system:

- A *business-process model* for the collection and use of customer data defines the scope of the data management system. The business-process model identifies the players that use collected data, the data they use, and how and for what purposes they use the data. The business model is formalized as the declaration of data, players and operations of an E-P3P privacy policy.
- A collection of *informal privacy policies* that govern the use of personal data in the business processes. They can be structured as bilateral privacy agreements that describe how data that is sent from one player to another may be used. Informal privacy policies are formalized as E-P3P privacy policy rules.

3.2 Policies and Separation of Duties

Privacy and security authorization in an enterprise involves at least four types of players. The *data subjects* are the players about whom personal data is collected. The most common data subjects are the customers of an enterprise. Other data subjects are employees or customers of cooperating enterprises. The next players are the *data users* within an enterprise who use collected data by executing *tasks* of *applications*. The other two players are the *privacy officer* (PO), who is responsible for privacy services, and the *security officer* (SO), who is responsible for security services.

E-P3P introduces the following intermediate abstractions and the corresponding policies (see Figure 1) in order to separate the duties of these players:

1. Personal data is collected in *forms*. A form is a set of fields, and fields have a type (e.g., string) as well as a PII type (e.g., "medical record", "address data", or "order data"). A form groups personal data and associates this data with its data subject. Examples of forms are "customer data", "purchase history", and "financial information". A "customer data" form, for example, may group the fields "name", "street", and "town".

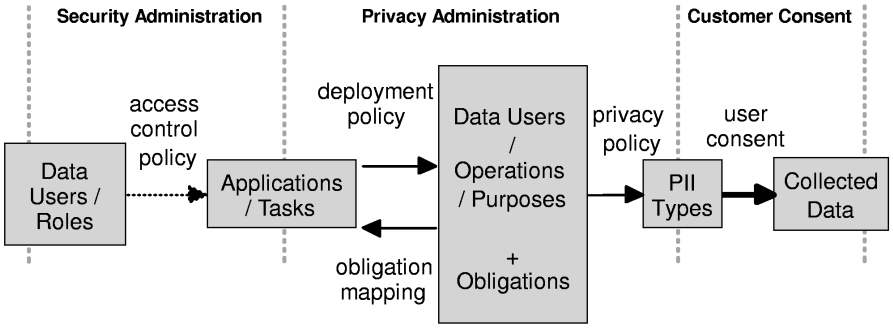

Fig. 1. Separation of duties for privacy authorization.

2. The PO defines a *privacy policy*. A privacy policy describes what *operations* for which *purpose* by which *data user* can be performed on each *PII type*. For example, the "marketing department" may be allowed to "read" the PII type "contact data" for purpose "e-mail marketing". In addition, a privacy policy may define opt-in and opt-out choices for the data subjects as well as certain privacy obligations such as "delete my data after 30 days unless parental consent has been given".
   The privacy policy should be enterprise- and application-independent. Enterprise-internals such as the role structure should not be used in order to enable exchange of policy-protected data between cooperating enterprises.
3. The PO and the SO define a *deployment policy* that maps legacy applications and their tasks onto the privacy-specific terminology used by the privacy policies. This mapping is specific to each enterprise. For example, whereas one enterprise maps a CRM system performing "product notification" as well as a printer for mass-mailings onto the action "read" for purpose "marketing", another enterprise, which uses a legacy application instead of the off-the-shelf CRM system, maps this legacy application onto "read" for "marketing".
4. The PO defines a *collection catalog* that identifies the sources where data is collected. For each source, the catalog defines the collected data, its PII types, and a default policy. This information is associated with an empty form for each particular collection point.
5. The PO defines an *obligation mapping* that translates application-independent obligations of the privacy policy (such as "delete") into specific implementations. For example, a delete may be translated into an "unsubscribe" of the mailing list.
6. The SO defines an *access control policy* that defines the roles and users within an enterprise. In addition, it defines which users or roles can execute which tasks of which applications.

3.3 Collecting Personal Data, Opt-in and Opt-out Choices, and Consent from a Data Subject

The collection catalog identifies an empty form for each collection point. At a given collection point, the data subject enters its data in the fields of the given form. The filled-out form contains the fields and PII types of the entered data as well as a default policy. The data subject may then choose opt-in and opt-out choices defined by the policy. By submitting the form, the data subject consents to the policy with respect to the selected choices. The choices are added to the form and the content of the form is stored.

Note that enterprise privacy policies that model access down to the employee level are usually too complex for end-users. As a consequence, it is advisable to present a coarser-grained privacy policy to the customer (either as text or P3P) and to implement an E-P3P policy for internal enforcement. For managing consent, a graphical user-interface is needed that enables the data subject to opt-in or opt-out of certain choices of the policy.

3.4 Granting or Denying Access

The form associating the collected data, the privacy policy and the selected options is used to decide whether an access shall be granted. Authorization is granted in two levels. Whereas access control focuses on restricting the access of employees to enterprise applications, privacy control restricts the access of applications to collected data.

Access Control: An employee acting as a data user with certain roles requests permission to perform a task of an application. The access control policy is used to verify that the data user with the given roles is in fact allowed to perform the requested task. If this is the case, the task is executed. This access control system is independent of the privacy authorization.

Privacy Control: Once a running task of a corresponding application has requested access to certain fields of collected data, the privacy enforcement system retrieves the form and uses it to allow or deny the given request as follows:

1. The request identifies the task of an application as well as the fields to be accessed.
2. The deployment policy maps the task onto a privacy-relevant operation and a purpose.
3. The form identifies the PII types of the requested fields.
4. The privacy policy and the data subject's choices are used to decide whether the operation for this purpose is allowed on the given PII types.
5. If the operation is denied, the access for the given task on the given fields is rejected.
6. If the operation is allowed and the privacy policy specifies a privacy obligation, the obligation mapping maps the obligation to a task of an application.[2]
7. If the operation is allowed, the task can be executed on the requested fields.

[2] Applications are responsible for managing their data. As a consequence, they are required to implement tasks that correspond to obligations in the privacy policy.

3.5 Sticky Policy Paradigm

An important aspect of E-P3P is the management of the data subject's consent on a per-person and a per-record basis. This is done by the *sticky policy paradigm*: When submitting data to an enterprise, the user consents to the applicable policy and to the selected opt-in and opt-out choices. The form then associates the opt-in and opt-out choices as well as the consented policy with the collected data. This holds even if the data is disclosed to another enterprise.

Note that policy management on a per-user basis is useful if consent and different sources are issues to be considered. Examples are managing data of different policy versions (e.g., due to different collection times), different user roles (e.g., paying users vs. users funded by advertising), or users from different jurisdictions (e.g., Europe and US).

3.6 Systems for Enterprise Privacy Enforcement

The authorization procedure described in Section 3.4 is structured into the privacy enforcement components depicted in Figure 2. The components interact as follows to

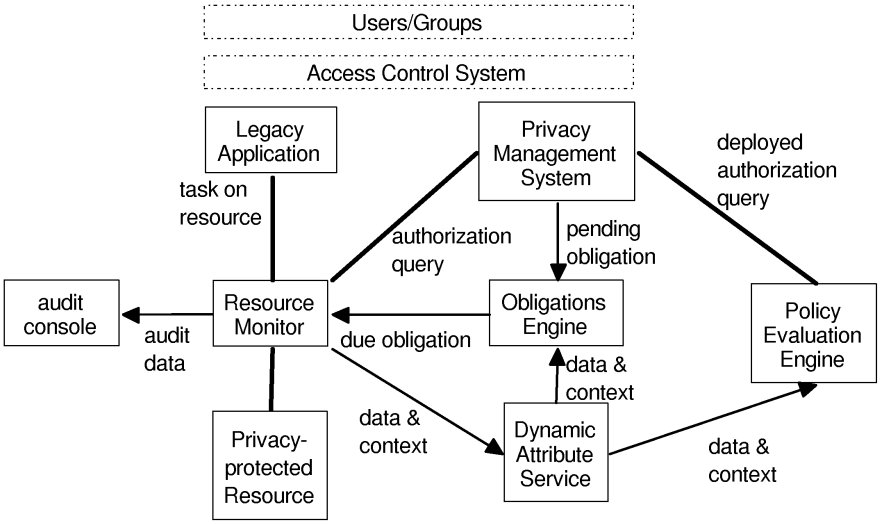

Fig. 2. Architecture for privacy enforcement

decide whether a task executed by a legacy application is allowed to access a protected resource:

1. A *legacy application* tries to execute a task on a protected resource.
2. A resource-specific *resource monitor* shields the resource and captures the request of a certain task for certain fields. For each task, it asks the privacy management system for authorization.

3. The resource-independent *privacy management system* obtains an authorization query identifying the fields to be accessed by a certain task of a certain application. It performs steps 1 to 3 of the authorization procedure in Section 3.4 to deploy the authorization query.
4. The *policy evaluation engine* performs step 4 of the authorization procedure. The policy evaluation engine needs context and data to evaluate conditions. The resource monitor abstracts from resource and storage details by using a dynamic attribute service [2] that provides values for data and context variables on request. The policy evaluation engine returns the decision as well as any resulting obligations to the privacy management system.
5. The *privacy management system* returns the decision of the policy evaluation engine to the resource monitor. If obligations were returned, the applications are mapped onto tasks (step 6) and sent to the obligations engine.
6. The *resource monitor* performs the tasks if it has been authorized. If not, the task is denied. In addition, the resource monitor sends log data to the audit monitor.
7. The resource-independent *obligations engine* stores all pending obligations. It evaluates the associated conditions based on values obtained from the dynamic attribute service. When a cancel-condition becomes valid, the obligation is removed. When a start-condition becomes valid, the obligation is sent to the resource monitor for execution.

4 E-P3P Privacy Policy Language

In a typical privacy policy, there are such statements as "*We* will use *your* address for e-mail marketing if *you* are not a minor." Usually, "we" denotes the data user and "you" denotes the data subject. We present a language to formalize such privacy policies.

4.1 Structure of a Privacy Policy

A privacy policy contains three elements. The first element is a header that contains information describing the policy such as a name, an author, and a version. The second element is the declaration that declares the identifiers, such as PII types, operations, and purposes. The third element are the authorization rules. The authorization rules can express what *operations* for which *purpose* by which *data user* can be performed on a given *PII type*. An example of a rule is that a "nurse" can "read" the PII type "medical record" for "care-taking" purposes. Rules can contain *conditions* that evaluate data in a form or external context variables. If an authorization rule contains a condition, the rule only applies if the condition evaluates to true. This includes opt-in and opt-out choices that are stored in the fields of a form. Examples include expressions such as "`Age>18`", "`OptInToEmailing=True`" or "`8am<currentTime<5pm`". Authorization rules can contain *obligations*. Obligations are required consequences of performing the authorized operation. An obligation, for example, can define that after storing data, it must be deleted after 30 days unless parental consent is obtained.

4.2 Declarations of a Policy

The first part of a privacy policy declares the data model as well as the identifiers used in the authorization rules. The declared obligations and context variables have to be supported by an implementation in order to be able to interpret a policy. This data-model can be specific to a single enterprise. If enterprises exchange data, they are required to agree on a common terminology before being able to exchange policy-protected data.

Mandatory fields and context attributes. Mandatory fields must be declared for each PII type if field names are used as variables in conditions. The declaration comprises the field name and its type. For example, the PII type "customer data" may mandate the fields "name", "street", and "town" of type "string". In addition, a declaration can define a list of opaque external context variables that can be used for constructing conditions. Table 1 lists the variables that can be used for constructing conditions.

Table 1. Fields and mandatory context attributes that can be used as variables in conditions.

Condition Variable	**Instantiated with**
`context.currentTime`	the current time
`context.executor`	the data user performing the operation
`context.operation`	the identifier of the requested operation
`context.collectionTime`	the time at which this particular form has been collected from Data Subject
`field.[fieldname]`	A variable for each mandatory field.
`argument.[argumentname]`	A variable instantiated with each argument of a declared operation that is currently executed.

Data users. The data user declaration defines the data users that are covered by the policy. Data users can either be distinct enterprises that use the data or different departments within an enterprise. Data users are not structured in a hierarchical manner. Note that the access-control notion of roles is different from our notion of data users. Whereas roles model enterprise-internals, data users represent entities that are different from a data subject's point of view.

Purpose hierarchy. Purposes are strings that identify the purposes for which an operation is executed. A privacy policy, for example, may authorize data disclosure for "research purposes" but not for "marketing purposes". Purposes are ordered in a hierarchical manner. We use a directory-like notation for purposes (e.g., marketing and its sub-purposes `marketing/email-mailings` and `marketing/postal-mailing`). If an operation is allowed for a given purpose, we assume that it is allowed for all sub-purposes.

Operations on the form. An enterprise uses the collected personal data by performing operations on it. Operations in our sense represent the privacy viewpoint regarding the use of data. A policy may use such terms as "use", "read", "disclose", or "anonymize",

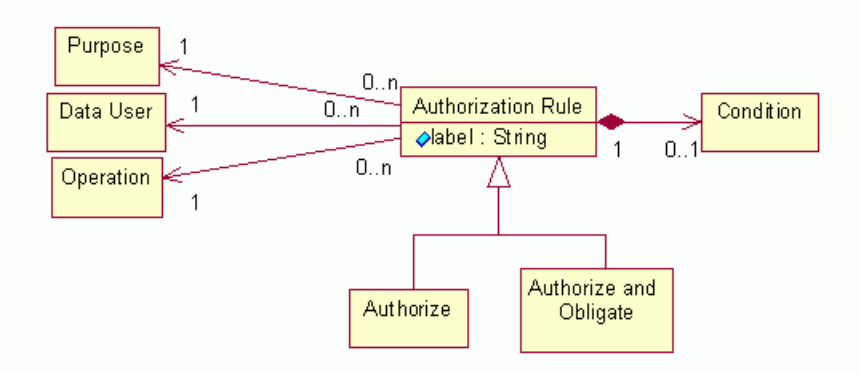

Fig. 3. Rules authorize purpose and operation under a condition.

even if "use" and "read" are both implemented by SQL statements. Each operation description contains a list of argument descriptions. Each argument description contains a unique name identifying this argument. Prefixed with "`argument.`", the argument name can be used as a variable for constructing conditions that evaluate this argument. Note that operations are different from purposes: A policy may authorize the reading of certain data for billing purposes but may not allow the same data to be disclosed for billing purposes to another enterprise.

Obligated operations. The set of obligated operations declares the operations that can be used in defining obligations of an authorization rule. The most important operations are deleting forms, getting consent, and notifying the data subject.

4.3 Authorization Rules

An E-P3P policy contains a set of authorization rules for each PII type. Each authorization rule (see Figure 3) states an operation that can be performed by a data user for a given purpose. Unless superseded by a more specific rule, each rule also holds for all sub-purposes of the declared purpose. A simple rule, for example, can state that an *enterprise* is authorized to *read* the data for *statistics*.

In addition to the basic authorization of *(purpose, data user, operation)* on a *PII type*, a rule may specify a condition that must hold. If no condition is included, the rule always holds. We distinguish two types of rules depending on their outcome:

- Authorize: The operation is authorized.
- Authorize and Obligate: The operation is authorized. Executing the operation results in the list of obligated tasks contained therein.

An *Authorize*-rule authorizes the triple (operation/purpose/data user) on a PII type if the condition is satisfied. An *Authorize and Obligate*-rule does the same but, in addition, it obligates a certain operation. In other words, if the authorized operation is performed, the obligation must be enforced as well. An example of such a rule is "the enterprise

may store my data for handling orders" with the obligation that "data be deleted within 30 days unless parental consent is obtained".

Conditions on data and context. Conditions define when an authorization rule can be applied. The standard cases for privacy policies are covered in the authorization triples (operation/purpose/data user) on a PII type. Policies that require more complicated authorizations are augmented by conditions that evaluate the data and context listed in Table 1. Variables and constants are evaluated using operators that depend on their type. The resulting Boolean expressions can then be further combined using Boolean operators. Examples of conditions are "`(thisForm.age>18) OR (thisForm.consent=True)`". and "`context.executor=field.PCP`" .

As conditions can evaluate fields containing choices, an explicit mechanism for opt-in and opt-out choices is not necessary. For example, by enabling an opt-in choice to e-mail direct marketing, a policy can declare a Boolean field "yes-to-emailing". The corresponding rule authorizes an operation "send mail" for purpose "direct marketing" only if this data field is set to true.

Descriptions of obligated tasks. An "authorize and obligate" rule specifies a list of obligated tasks. Each obligated task contains a list of operation descriptions (operation name and arguments) together with a `start`- and `cancel`-condition. The intuition is that the system queues obligations that result from performing an authorized operation. The `start`-condition then defines a precondition. If this condition is satisfied, the obligation must be executed. The `cancel`-condition defines a postcondition. If this condition is satisfied, the obligation is no longer necessary and may be deleted from the queue. Obligation conditions are expressed in the language for expressing authorization conditions with the extension that obligations can use the prefix "`^`" for variables with deferred instantiation. Those variables are not instantiated during authorization but every subsequent time the obligation conditions are evaluated. When the obligation conditions are evaluated, the non-deferred variables have already been instantiated and are now used like constants. The variable "*today*" is instantiated when the authorization condition is evaluated, whereas the variable "`^today`" is instantiated when a start or stop condition is evaluated. As a consequence, an obligation condition "`^today > today + 1day`" becomes true one day after the authorization rule has been applied.

An example of such a complex obligated task is the obligation that a form must be deleted after 30 days if parental consent is missing. For this task, the operation is "`delete`", and the start condition is "`^today > today + 30d`", whereas the cancel condition is "`^ParentConsent=True`".

4.4 Policy Evaluation Logic

We now describe how privacy policies are evaluated. This algorithm refines the policy evaluation (step 4 in Section 3.4) for the refined privacy policies. The algorithm answers a request whether a data user is authorized to perform a certain operation for a given purpose and returns any resulting obligations. The algorithm assumes that for a given purpose, data user, PII type, and operation there is at most one rule with obligations and

a condition that evaluates to true for any given data and context. For a given policy, the decision can depend on the following external inputs:

1. The data user, the requested operation, its purpose, and the PII type on which the operation shall be performed.
2. The data contained in the data fields that are declared in the policy.
3. The context information that has been declared in the policy.

The first item is a mandatory input for any authorization request. The latter two are additional information that is retrieved on request using the dynamic attribute service. The policy evaluation engine returns its decision and resulting obligations.

Authorizing Access. The authorization request is evaluated as follows:

1. The engine retrieves all rules that apply to the given tuple (PII type, data user, operation).
2. The engine retrieves the set of most specific rules for the given purpose. This is a longest-matching prefix search on the purposes of the applicable rules. The result is a reduced set of applicable rules.
3. The engine evaluates the conditions of each remaining applicable rule. To evaluate the conditions, the authorization engine instantiates the variables contained in the condition using the context attributes that are retrieved via the dynamic attribute service. All rules with conditions that evaluate to false are removed from the set of applicable rules.
4. If there is more than one remaining rule with obligation, the request is denied due to inconsistencies in the policy.

If the set of applicable rules is empty (i.e., if the conditions were not satisfied), the request is denied. If there is only one remaining rule with an obligation, the request is allowed and the obligation is returned. If none of the remaining rules contains any obligation (i.e., if multiple rules authorize the request but none imposes obligations), the request is granted.

5 Conclusion

We have described the first scheme for enterprise privacy management. The enterprise-independent privacy policies are comprehensive and more expressive than existing proposals for enterprise-internal privacy policies. The deployment scheme enables enforcement of a common privacy policy for a variety of legacy systems.

The viable separation of duties between the privacy officer and the security administrator enables secure and efficient management in practice. The intuitive consent-management paradigm enables customers to retain greater control over their personal data.

Our methodology protects personal data within an enterprise with trusted systems and administrators against misuse or unauthorized disclosure. It cannot protect data if the systems or administrator are not trusted. Therefore, it merely augments a privacy-aware design of enterprise services that minimizes the data collected. In the desirable

(but unlikely) scenario where an enterprise can offer its services without collecting personal data, our privacy management methodology would be rendered obsolete.

To correctly specify privacy rights and obligations that are being promised by privacy statements and mandated by a number of legislatures, the privacy officer must be able to reconcile easily what should be authorized with what is actually authorized. Therefore, we have developed a formal model for authorization management and access control in privacy protecing systems [8].

Acknowledgments

We thank Kathy Bohrer, Nigel Brown, Jan Camenisch, Calvin Powers and Els Van Herreweghen for valuable comments.

References

1. M. Abrams. Renewed understanding of access control policies. In *16th National Computer Security Conference*, pages 87–96, 1993.
2. K. Beznosov: Information Enterprise Architectures: Problems and Perspectives. School of Computer Science, PhD Thesis, Florida International University, June 2000.
3. 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*. Paris, France, June 2001.
4. K. Bohrer and B. Holland (eds.): The Customer Profile Exchange (CPexchange) Specification; Version 1.0, International Digital Enterprise Alliance, October 20, 2000 (from `www.cpexchange.org`).
5. S. Fischer-Hübner: *IT-Security and Privacy*. Lecture Notes in Computer Science 1958, Springer-Verlag, 2001.
6. S. Hada and M. Kudo. XML Access Control Language: Provisional Authorization for XML Documents, Tokyo Research Laboratory, IBM Research, October 16, 2000 (from `www.trl.ibm.com/projects/xml/xacl/`).
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. G. Karjoth and M. Schunter. A Privacy Policy Model for Enterprises. In *15th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, 2002.
9. M. Kudo and S. Hada. XML Document Security based on Provisional Authorizations. In *7th ACM Conference on Computer and Communications Security*, pages 87–96, 2000.
10. C. J. McCollum, J. R. Messing, and L. Notargiacomo. Beyond the pale of MAC and DAC – defining new forms of access control. In *IEEE Symposium on Security and Privacy*, pages 190–200, 1990.
11. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Recommendation, 16 April 2002 (from `www.w3.org/TR/2002/REC-P3P-20020416/`).
12. R. Sandhu, E. Coyne, H. Feinstein, and C. Youman: Role-based Access Control Models, *IEEE Computer*, 28/2 (1996) 38-47.

A Example – Privacy at Borderless Books Inc.

In this section, we outline the most important elements of a privacy policy for a hypothetical online bookstore called Borderless Books.

Table 2. Fields of personal data collected by Borderless Books.

Fieldname(s)	Data Type	PII Type	Description
Name, Surname, Address, PostalCode, Phone, Email, ParentName, ParentEmail, CardType, CardOwner, CardNumber	String	CP	Fields containing collected PII.
Birthdate, CardExpiry	Date	CP	Birthdate for determining the age.
Birthdate, CardExpiry	Date	PD	Credit card expiry data.
YesToMarketing	Boolean	CP	Opt-in choice for personalized marketing.
ParentConsent	Boolean	CP	Set to true if parent consent has been given.
ParentID	String	CP	Identifier of the parent.
Password	String	CP	Password for authentication of the data subject.
OrderHistory	List	OD	The order history managed by the enterprise.
DataSource	String	MD	The source where the form has been obtained.
UserName	String	MD	The user name of the data subject.

A.1 Designing a Privacy Policy

The main business of Borderless Books is to sell books. Customers compose an order using a shopping basket. To process an order a customer enters his user name and password. This identifies a customer profile. The order is stored as part of the customer profile. Name and credit card number of the form are sent to a payment processor for authorization. Borderless collects four types of PII in its customer database: CustomerProfile (CP), PaymentDetails (PD), OrderDetails (OD), and enterprise-internal management data (MD). The entry points are a Web page for creating customer profiles as well as the shopping basket. Both entry points are governed by a single policy.

A customer profile contains the fields UserName, Password, Name, Surname, Address, Phone, E-mail, and Birth date as well as the credit card information (CardType, CardNumber, Expiry Date, CardOwner). The former are of PII type CustomerProfile whereas the latter is PaymentDetails. The OrderHistory is of PII Type OrderDetails. If the data subject is a minor, he also has to provide the name and email of a parent. The customer can give permission to Borderless to use his PII for marketing, to send the PII to third-party marketers, or to use depersonalized data for statistics.

Identifying Privacy Practices The following list contains a list of purposes for which PII is used as well as the corresponding operations:

Profile Management: A customer consents to the data being stored. If a customer is a minor and if no consent by the parents is given, the data is deleted within 30 days. Upon request of the customer, the data is deleted.

The purpose profile management uses the following operations: The operation `Store` is used to store the data at the enterprise. The operation `Update` enables an application that acts on behalf of the data subject to update the collected data. The operation `Delete` deletes a form and all copies. Before deleting the form, the operation sends a delete request to all parties to whom the data has been disclosed.

Processing Orders: When Borderless Books processes Joe's order, it sends the credit card company an invoice for payment. Borderless Books is authorized to disclose payment data from the slip to the credit card company, but not to put the titles of the ordered books on the invoice.

This purpose uses the following operations: The operation `Read` reads the data for local use by Borderless Books. The operation `Write` writes an enterprise data field. The parameters are a field name ("field"), a new value ("value"). Operation `SendDisclosure` sends PII to another enterprise. The input parameters are an identifier of the party to whom the data shall be disclosed (the "disclosee" of type Role), the purpose ("purpose"), as well as a subset of the fields that shall be disclosed ("fields"). Operation `StoreDisclosure` stores a form that has been obtained from another enterprise. The input parameters are a form ("form") and a purpose ("purpose").

Personalized Marketing: If the customer consented to personalized marketing, his data (including the order history) can be disclosed to third-party marketers.

The data model is depicted in Table 2. It shows the collected fields as well as their PII type together with a brief description of their intended use.

A.2 Formalizing the Privacy Policy

The goal is to formalize the following policy:

> *Borderless Books uses your data only for processing orders.*
> *The data is not disclosed to any other party except for processing payments. The payment processor is obliged to delete the data within 1 day.*
> *If you consented to personalized marketing, we disclose the data to Direct Marketing Inc., which is not allowed to disclose the data further.*
> *For minors, we will delete the data if no parental consent is given within 30 days.*

Table 3 contains the rules that formalize this policy. The necessary declarations can be derived from this table.

Table 3. Rules defining Borderless Bookstore's privacy policy.

PII Types	**Action**	**Data User**	**Purpose**	**Condition**	**Obligation(s)**
All	Read	Borderless	processing order		
All	Read	Borderless	parental access	initiator=ParentID	
PD	Read	CreditCardInc	payment processing		
All	Store	Borderless	creating profile	today-birthDate≥18y	
All	Store	Borderless	creating profile	today-birthDate<18y	Delete at (today+30d) unless ^ParentConsent
CP, OD	Write	Borderless	processing order	(fieldname=OrderHistory OR fieldname=ParentConsent)	
All	SendDisclosure	Borderless	personalized marketing	disclosee="*DirectMarketingInc*" AND YesToMarketing=True AND (today-birthDate ≥ 18y OR ParentConsent)	
PD	SendDisclosure	Borderless	payment processing	disclosee=CardType AND (today-birthDate ≥ 18y OR ParentConsent)	
PD	ObtainDisclosure	CreditCardInc	payment processing		Delete if ^today > today+1d
CP	Update	Borderless	updating collected data		
CP	Update	Borderless	adding consent	field=ParentConsent AND initiator=ParentID	
All	Delete	Any	none	initiator IN {ParentID, DataSubject, DataSource, DataUser}	