Towards Scalable Management of Privacy Obligations in Enterprises

Marco Casassa Mont
Trusted Systems Laboratory
HP Laboratories Bristol
HPL-2006-45
March 16, 2006*

privacy, privacy obligations, scalability, privacy management, privacy enforcement, identity management

Privacy management is important for enterprises that collect, store, access and disclose personal data. Among other things, the management of privacy includes dealing with privacy obligations: privacy obligations dictate duties and expectations an enterprise has to comply with, in terms of data retention, deletion, notice requirements, etc. This is a green area open to research and innovation. This paper provides an overview of the work we have done in this space to explicitly represent, enforce and monitor privacy obligations: this includes an obligation management model and framework, a working prototype and its integration both in the context of PRIME project and with an HP identity management solution. This paper then focuses on an important issue: how to make our approach scalable, in case large amounts of personal data have to be managed. Thanks to our integration work and the feedback we received, we learnt a few lessons on how users and enterprises are likely to deal with privacy obligations. We describe these findings and how to leverage them. Specifically, in the final part of this paper we introduce and discuss the concepts of parametric obligation and hybrid obligation management model and how this could help to make our system both scalable and flexible at the same time. Our work is in progress. Further research and development is going to be done in the context of the PRIME project and an HP Labs project.

* Internal Accession Date Only
Approved for External Publication

Towards Scalable Management of Privacy Obligations in Enterprises

Marco Casassa Mont

Hewlett-Packard Labs, Trusted Systems Lab
BS34 8QZ, Bristol, United Kingdom
marco.casassa-mont@hp.com

Abstract. *Privacy management is important for enterprises that collect, store, access and disclose personal data. Among other things, the management of privacy includes dealing with privacy obligations: privacy obligations dictate duties and expectations an enterprise has to comply with, in terms of data retention, deletion, notice requirements, etc. This is a green area open to research and innovation. This paper provides an overview of the work we have done in this space to explicitly represent, enforce and monitor privacy obligations: this includes an obligation management model and framework, a working prototype and its integration both in the context of PRIME project and with an HP identity management solution. This paper then focuses on an important issue: how to make our approach scalable, in case large amounts of personal data have to be managed. Thanks to our integration work and the feedback we received, we learnt a few lessons on how users and enterprises are likely to deal with privacy obligations. We describe these findings and how to leverage them. Specifically, in the final part of this paper we introduce and discuss the concepts of parametric obligation and hybrid obligation management model and how this could help to make our system both scalable and flexible at the same time. Our work is in progress. Further research and development is going to be done in the context of the PRIME project and an HP Labs project.*

1 Introduction

Enterprises that store, manage and process personal data must comply with privacy laws and guidelines and satisfy people's expectations on how their personal data should be used.

Privacy laws [1,2,3] dictate policies on how personal data should be collected, accessed and disclosed according to stated purposes, by keeping into account the consent given by data subjects (e.g. customers, employees, business partners) and by satisfying related *privacy obligations* including data retention, data deletion, notice requirements, etc.

The management and enforcement of privacy policies in enterprises is a green field: key requirements include automation, cost reduction, simplification, compliance checking and integration with existing enterprise identity management solutions. In particular the management of *privacy obligations* is open to research and innovation. Privacy obligations [4] dictate duties and expectations on how personal data should be managed. They require enterprises to put in place *privacy-aware information lifecycle management* processes.

During the last two years we have been active in the *privacy obligation management* [5] space by: (1) researching and defining an explicit model for privacy obligations; (2) formalising the representation of obligations; (3) proposing an obligation management framework to deal with the explicit scheduling, enforcement and monitoring of privacy obligations.

Based on this, we have built a working prototype - referred in this paper as "*obligation management system*". Part of this work has been done in the context of the PRIME project [6]. This prototype has also been integrated (as a proof of concept) with an HP identity management solution (HP Select Identity [7]) to demonstrate the feasibility of our approach and show that our solution can be deployed into state-of-the-art enterprise middleware – in particular in a context of user provisioning and account management [8]. This paper provides an overview of this work.

The main focus of this paper is on an important open issue that has been identified in our work and potential next steps to address them. Our current obligation management system allows end-user to customise - in a fine-grained way - their personal preferences: related privacy obligations are automatically generated and associated to users' data. However, this causes scalability issues when large sets of personal data must be managed as a large set of privacy obligations might be generated and then it has to be managed. This is very important for

enterprises that potentially have to deal with millions of records related to customers, employees or business partners.

The integration phase of our work and the feedback we received from third parties (customers, HP businesses, etc.) has helped us to better understand how users are actually likely to define their privacy preferences and which realistic support enterprises can provide in terms of handling privacy obligations. We describe these findings and highlight how they can actually be leveraged to address the scalability issues. The final part of this paper describes our related ideas, based on the concept of *parametric obligations* and *a hybrid model of privacy obligations*. This work is in progress and will be carried on in the context of PRIME and an HP Labs project.

2 Management of Privacy Obligations in Enterprises

Privacy obligations [4,5,9] are policies that dictate expectations and duties to enterprises on how to handle personal data and how to deal with its lifecycle management. Privacy obligations include: dealing with data deletion and retention, dealing with data transformation (e.g. encryption), sending notifications, executing workflows involving human and system interactions, etc.

It is important to notice that the management and enforcement of privacy obligations must not be subordinated to the management and enforcement of access control policies [4]. For example, deletion of personal data at a precise point in time has to happen independently from the fact that this data has ever been accessed.

Related work in the space of privacy obligations includes EPAL [10]. It defines a privacy language, inclusive of a placeholder for obligations, in the context of their Enterprise Privacy Authorisation architecture [11]. However, this work does not define obligation policies in detail and subordinate their enforcement to access control. Similar observations apply for the XACML [12] specification.

Our work aims at addressing these aspects by explicitly representing and managing privacy obligations. Comparisons of this work with other relevant work is provided in [4,5,9]. More details about our work follow.

2.1 Our Work

We have defined an *obligation management model* [4,5,9], where privacy obligations are “first class” entities, i.e. they are explicit entities that are modeled, managed and enforced.

This management and enforcement of privacy obligations is not subordinated to access control but handled by an obligation management framework [4,5,9].

In this model, a privacy obligation is an explicit “object” that includes the following aspects [9]: *Obligation Identifier*; T*argeted Personal Data* (e.g. data affected by the obligation); *Triggering Events* (e.g. time-based events); *Actions* (e.g. data deletion, sending notifications).

Different categories of privacy obligation [9] need to be managed and enforced by enterprises: *transactional obligations*; *data retention and handling obligations*; o*ther types of event-driven obligations*. A complementary classification of our managed privacy obligations is based on their activation timeframe and period of validity: *short-term obligations*; *long-term obligations*; *ongoing obligations* [9].

Figure 1 shows a very simple example of a privacy obligation (expressed in XML) dictating the deletion of a personal attribute (credit card detail) of a specific user (in the example having the *PSEUDO1* unique identifier) at a predefined period of time, along with the need to notify him/her via their e-mail.

```
<obligation ObligationId="OBLID1">
        <target  // Reference to the PII Data the obligation is associated to
            <data repository>databaseA <data repository>
            <data structure type=TABLE> CustomerTable </data structure>
            <data attr="ALL"  @key:UserId:PSEUDO1 </data>
        </target>
        <events operator="">
            <event id="e1">
                    <type>TIMEOUT</type>
                    <date now="no">
                          <year>2007</year>  <month>10</month> <day>13</day>
                           <hour>14</hour <minute>01</minute> <second>00</second>
                     </date>
            </event>
        </events>
        <actions>
              <action id="a1">
                  <type>DELETE</type>
                    <data attr="part">
                     // Reference to the PII Data attribute
                      <item>
                         @key:UserId:PSEUDO1|att:CreditCard
                      </item>
                     </data>
              </action>
              <action id="a2">
                       <type>NOTIFY</type>
                       <method>EMAIL</method>
                   // Reference to the PII Data attribute
                   <to>
                       @key:UserId:PSEUDO1|att:E-Mail
                   </to>
              </action>
        </actions>
</obligation>
```

Fig. 1. Simple Example of Privacy Obligation

We introduced and designed an *obligation management framework* (shown in Figure 2) to manage these privacy obligations [4,5,9], based on the following principles:

- **Data subjects** can explicitly define privacy preferences (e.g. on data deletion, notifications, etc.) on their personal data at the disclosure time (e.g. during a self-registration process) or at any subsequent time. These preferences are automatically turned into privacy obligations;
- **Enterprise privacy administrators** can further associate other privacy obligations to personal data, for example dictated by laws or internal guidelines.

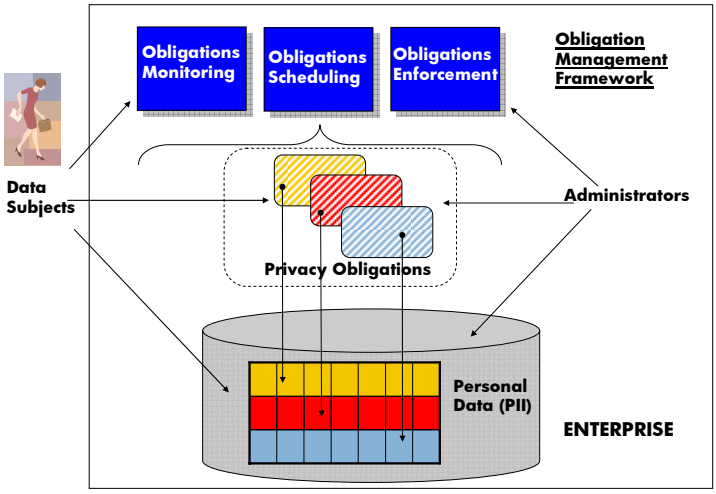

Fig. 2. Obligation Management Model

Our *obligation management framework* handles privacy obligations by providing the following core functionalities:

- **Scheduling the enforcement of privacy obligations**: the system schedules which obligations need to be fulfilled and under which circumstances (events);
- **Enforcing privacy obligations**: the system enforces privacy obligations once they are triggered. Enforcement may range from execution of simple actions to complex workflows involving human intervention;

- **Monitoring the fulfilment of privacy obligations**: the system monitors and audits the enforced obligations, at least for a predefined period of time, to ensure that the desired status of data is not changed and to report anomalies.

More details can be found in [4,5,9]. These functionalities can be accessed by enterprise privacy administrators and potentially also by data subjects, for example to monitor their personal data and check for privacy compliance. Figure 3 shows the high-level architecture of our obligation management system derived from our obligation management framework.

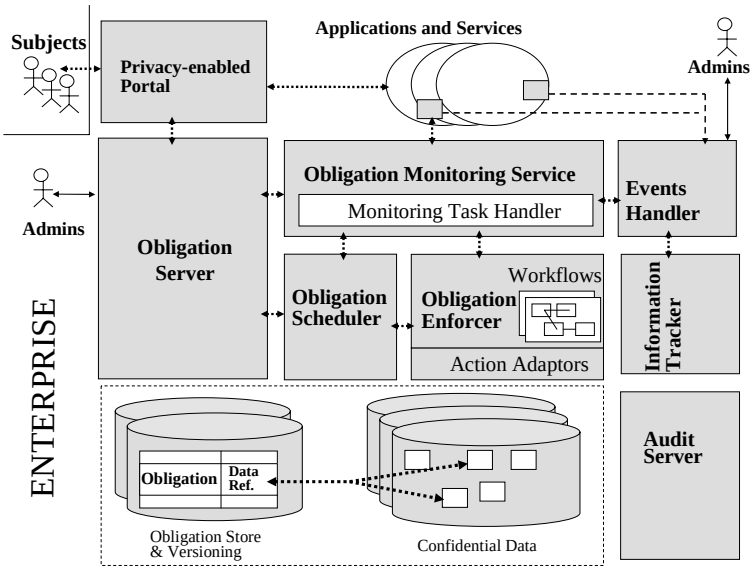

Fig. 3. High-level Architecture of our Obligation Management System

This system consists of the following modules:

- **Obligation Server**: this deals with the authoring, management and storage of obligations. It turns privacy preferences into privacy obligations via a template –based mechanism. It explicitly manages the association of privacy obligations to personal data and their tracking and versioning. Active obligations (i.e. obligations to be fulfilled) are pushed to the "Obligation Scheduler";
- **Obligation Scheduler**: this knows which obligations are active, ongoing obligation deadlines, relevant events and their association to obligations. When events/conditions trigger the fulfilment of one or more obligations, this component activates the corresponding "workflow processes" of the "Obligation Enforcer" that deals with the enforcement of the obligation;
- **Obligation Enforcer:** this is a workflow system containing workflow processes describing how to enforce one or more obligations. The enforcement can be automatic and/or could require human intervention, depending on the nature of the obligation;
- **Events Handler**: this is in charge of monitoring and detecting relevant events for privacy obligations and sending them to the obligation scheduler. It coordinates its activities with other instrumented components;
- **Obligation Monitoring Service**: this is orthogonal to the scheduling and enforcement components and monitors enforced obligations and the expected status of data;
- **Information tracker**: this intercepts events generated by data repositories, databases and file systems containing confidential data and provides this information to the event handler;
- **Audit Server**: this logs the various events and messages generated by all the components.

As a proof-of-concept, a working prototype has been fully implemented and integrated in the context of the EU PRIME project [6]. To demonstrate the feasibility and applicability of this work within enterprises, we also integrated it [8] with HP OpenView Select Identity (an HP state-of-the-art identity management solution [7]) to manage privacy preferences and related privacy obligations during user provisioning and user account management.

3 Scalability Issues

Our system (see Figure 2) provides flexible, fine-grained mechanisms to end-users (and enterprise privacy administrators) to express their privacy preferences (e.g. deletion preferences, notification preferences, etc.): it automatically turns them into privacy obligations (by means of translation rules) to be managed by our obligation management system.

The side-effect of this flexibility (at least in the current implementation) is that for each piece of personal data disclosed by a user, one or more privacy obligations could be generated, each of them with its own specific properties and requirements. For example, each user of an e-commerce site could potentially specify different privacy preferences (deletion date, notification preferences, encryption of data, data minimisation, etc.) and privacy constraints (among the ones supported by the enterprise) on their personal data. As a consequence, in the current system, one or more explicit privacy obligations (i.e. obligation objects) are generated for each user's personal data and then managed by our obligation management system. Figure 4 shows this approach (architectural details are omitted for simplicity).

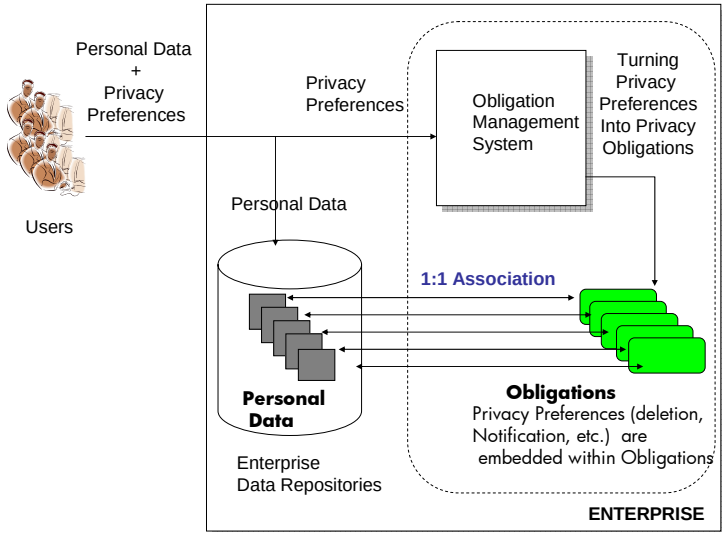

Fig. 4. Current Model: Direct Association of Privacy Obligations to Personal Data

In case of large amounts of users, large amounts of privacy obligations are created and subsequently they must be scheduled, enforced and monitored by our obligation management system. In general, the number of managed privacy obligations linearly grows with the number of managed users. Despite the fact that the components of our system can be replicated and distributed [9], the overhead of managing all these obligations could be overwhelming, both in terms of computation and in terms of human-based administration.

Related to the latter aspect, current GUI administrative tools [9] to manage privacy obligations within enterprises can potentially display all the managed privacy obligations along with their current status (to be enforced, enforced & compliant, enforced & violated, etc.). See Figure 5 for details.

These GUI tools already allow administrators to focus on sub-set of managed obligations, based on some of their properties. However, in case of large amounts of managed privacy obligations, the task of selecting the relevant privacy obligations or having an overall view of the status of monitored obligations could be difficult to achieve.

To summarise, addressing the scalability problem requires to:

- Deal with large amount of personal data (potentially millions of records) and related privacy obligations;
- Do it in efficient and practically usable way;
- Provide adequate administration and obligation lifecycle management capabilities.

These issues were known at the design time of our current prototype: however more urgent and preliminary work was required to research the very concept and properties of privacy obligations. Our first prototype was meant to demonstrate the feasibility of our approach and use it as a starting point to make further experiments.

The remaining part of this paper describes our thoughts and ideas on how to address the scalability problem, based on a few lessons that we have learnt and how to move forwards.

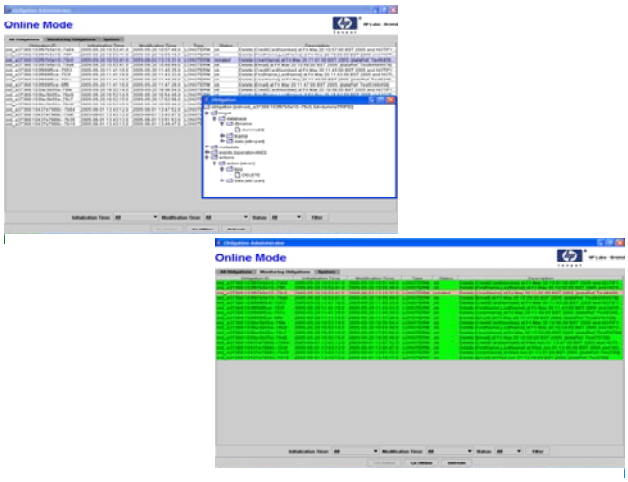

Fig. 5. Enterprise Administration tools for Managed Privacy Obligations

4 Towards Scalable Management of Privacy Obligations

As described in the previous section, the main cause of the current scalability problem is that (in order to allow users to define their privacy preferences in a fine grained way) our obligation management system generates one or more privacy obligations every time personal data is disclosed: these obligations can potentially be different in their structure and declared constraints.

In the last months, we have learnt a few lessons thanks to: the integration of our system in PRIME and with the HP identity management solution; the overall feedback we received. This has provided us with more insights and ideas on how to address the scalability problem – in a way we can leverage and extend our current work. Next sections provide more details about these two points.

4.1 Learnt Lessons

Our obligation management system has been integrated with the PRIME system [6] to provide a comprehensive privacy-enhanced identity management solution both at the user side and the enterprise side.

At the integration time, it has been clear that it would have not been feasible for the enterprise to support users in defining any arbitrary combination of privacy preferences and constraint specifications. This because of the involved costs, the complexity of developing a general purpose solution and usability aspects for users.

We have learnt that it would be preferable to provide users with a list of predefined "types" of privacy obligations that can be supported by an enterprise (for given types of personal data to be disclosed). Each type of privacy obligation clearly states which relevant *privacy preferences* a user can specify (e.g. data deletion time, notification preference, etc.). In the integrated PRIME system, the enterprise side describes these "types" of privacy obligations by means of "*Obligation Templates*".

An "*Obligation Template*" is graphically rendered to users at the time they need to disclose their personal data. In doing this, users can intuitively instantiate the related privacy preferences (these required preferences and information to be instantiated are expressed in the template with the "*[?]*" notation – see Figure 6).

Figure 6 shows a simple example of *Obligation Template* (related to the example of privacy obligation shown in Figure 1), defined by the enterprise, allowing users to specify their preferences in terms of deletion of their financial information at a specific point of time and being notified.

```
<obligation ObligationId="OBLID1">
        <target  // Reference to the PII Data the obligation is associated to
            <data repository>databaseA <data repository>
            <data structure type=TABLE> CustomerTable </data structure>
            <data attr="ALL"  @key:UserId:[?] </data>
        </target>
        <events operator="">
            <event id="e1">
                <type>TIMEOUT</type>
                <date now="no">
                    <year>[?]</year>  <month>[?]</month> <day>[?]</day>
                     <hour>[?]</hour <minute>[?]</minute> <second>[?]</second>
                </date>
            </event>
        </events>
        <actions>
            <action id="a1">
                <type>DELETE</type>
                <data attr="part">
                 // Reference to the PII Data attribute
                  <item>
                     @key:UserId:[?]|att:CreditCard
                  </item>
                 </data>
            </action>
            <action id="a2">
                    <type>NOTIFY</type>
                    <method>EMAIL</method>
                 // Reference to the PII Data attribute
                 <to>
                    @key:UserId:[?]|att:E-Mail
                 </to>
            </action>
        </actions>
</obligation>
```

Fig. 6. Simple Example of Obligation Template

Once privacy obligations have been instantiated (with the relevant privacy preferences) they are processed by our obligation management system as described in section 2. For example, the instantiation of the Obligation Template in Figure 6 would be a privacy obligation similar to the one shown in Figure 1.

This approach to "*predefine and standardise*" types of managed obligations is also consistent with the feedback we received (by customers, HP business divisions and third parties) and our direct experience in integration our system with the HP identity management solution: in these cases the main drivers where simplification of the overall specification and management processes, both for the enterprise and users.

By using this approach, all the obligations derived from a predefined "type" (obligation template) have the same structure (i.e. the same template, describing the same types of events and actions): the only aspects that differentiate them are the privacy preferences provided by end-users. These preferences are *embedded* within these obligations.

Of course, in case of large amounts of personal data (of related users), our obligation management system had still to deal with a large number of privacy obligations – hence again the scalability issue. At this point, however, we realised that each set of structurally identical obligations requires the same type of management, enforcement and monitoring: as such, each set can be represented by just *an abstract obligation* that is *parametric* to the related privacy preferences expressed by users.

This introduced the concept of P***arametric Privacy Obligation***: its properties and the implication for our obligation management model are described in the next section.

4.2 Model of Parametric Privacy Obligations

This section describes our current thoughts and ideas on how to address the scalability issues by leveraging the concept of *parametric obligation*. A *parametric obligation* is an obligation containing a parametric definition of its sub-components, i.e. *Target*, *Events* and *Actions*. Its structure is based on obligation templates defined by enterprise privacy administrators.

In this context, privacy preferences *are not anymore embedded* within obligations (as it happens in the current system). These privacy preferences are still managed by the obligation management system but they are stored in a separated, explicit *data structure* (e.g. database tables in a relational database) – referred in this paper as *"Privacy Preferences" data structure* - along with a reference to the personal data they are associated to. Hence, in this model, a parametric obligation is associated to a set of privacy preferences and related personal data – as shown in Figure 7.

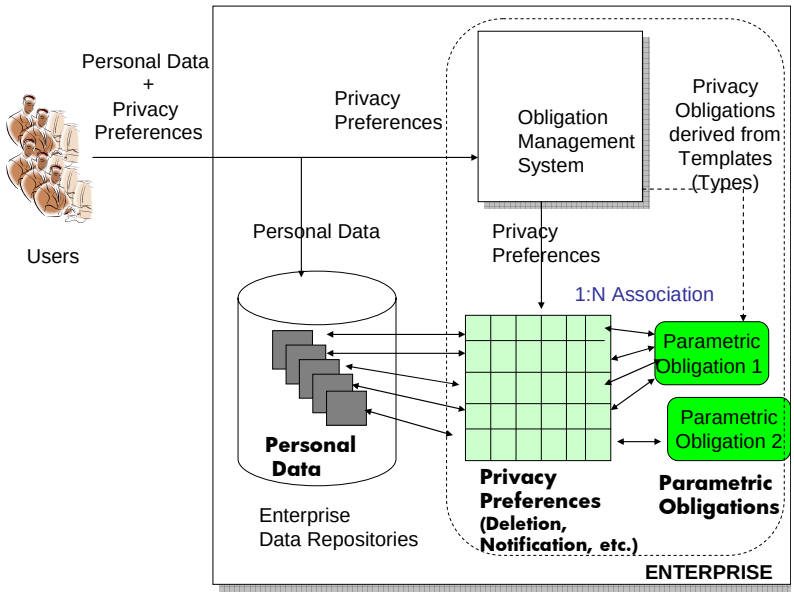

Fig. 7. Association of Parametric Obligations to Personal Data

The key components of a parametric obligation are specified as follows:

- **Target**: this specifies the set of personal data and associated privacy preferences the parametric obligation refers to. For example, users' data could be stored in one or more tables (e.g. in an enterprise relational database). An additional table can be allocated to store privacy preferences. The Target section will contain information about these tables, their relationships and the subset of data of relevance;
- **Events**: they contain references to where relevant preferences (e.g. deletion time) are stored, within the "Privacy Preferences" data structure. Each reference is a "generic" reference (e.g. it could be the name of the column in the preference table containing the relevant information) and it is valid for multiple pieces of personal data subject to the same type of obligation;
- **Actions**: they also contain references to where relevant preferences (e.g. notification preference) are stored, within the "Privacy Preferences" data structure. The same comments made for the "Events" section apply.

In this model, each parametric obligation dictates an identical set of duties and expectations to be fulfilled on a potentially large set of personal data, individually customised by associated privacy preferences.

As privacy preferences are not embedded within parametric obligations, the resulting effect is that the set of parametric obligations is now reasonably small, depending on the different types of obligations to be managed by the enterprise. In other words, given a predefined set of obligation types (i.e. obligation templates), the obligation management system will have to manage a correspondent set of parametric obligations. As these sets are meant to be small, this is a step towards addressing the scalability problem.

The current *obligation management system* needs to be extended to deal with these parametric obligations. For each managed parametric obligation it has to:

- Focus on the targeted set of personal data and related preferences;
- Capture and manage the events that are relevant to all this data;
- Check if any of these events can trigger the execution of specific actions. If so, execute these actions.

On one hand, this extended obligation management system will have only to manage a restricted set of (parametric) obligations. On the other hand, however, each parametric obligation could be associated to a potentially large set of personal data along with their related preferences. For each piece of personal data, this system must remember relevant "operational" information (related to associated parametric obligations), such as the local status of the events that might trigger the execution of actions. In case of composite events [9] (including stateful events, such as access counters) additional intermediate information must be stored. This can be done in additional data structures managed by the obligation management system.

Despite the fact that the management of events and actions might relate to a potentially large amount of data, we believe that these operations can now be optimised by using appropriate data structures and ways to manipulate this data via standard data access mechanisms. For example indexed tables could be used within relational databases to store the relevant information (personal data, preferences and auxiliary data) and (optimised) SQL

queries used to make inferences, extract and update the relevant information. Research is in progress on these aspects.

4.3 Hybrid Obligation Management Model

Our current obligation management system (described in section 2.1) needs to be *extended* to schedule, enforce and monitor parametric obligations. Nothing prevents that "traditional" privacy obligations and parametric obligations coexist in the same system: this introduces a hybrid model and framework to manage privacy obligations.

This model can provide users and enterprises with a comprehensive and flexible solution that can adapt to varying needs and requirements.

In case large amounts of personal data need to be handled, the support for parametric obligations will allow enterprises to deal with scalability issues by containing the number of managed obligations. Nevertheless in those cases where more flexibility and customisation is required by users when defining privacy obligations, this will still be supported and managed by the system.

Hence, depending on the context and requirements, a mixture of the two capabilities can be provided to address at the best needs for scalability, flexibility and customisation.

5 Discussion

We believe that the proposed model does not limit the control that users have in specifying their privacy preferences: it actually makes the overall process more effective by allowing enterprises to declare upfront which types of privacy obligations they can support and letting users make their informed decisions.

In addition to refining the concept of parametric obligation and understanding (in more details) how to extend our current obligation management system, work also needs to be done to better understand how to provide more suitable administrative and GUI tools.

Current GUI tools allow administrators to administer one-by-one every privacy obligation, by displaying their properties and current status (to be enforced, enforced & satisfied, enforced & violated). In case of parametric obligations this capability has to be extended, as a parametric obligation can potentially refer to a large set of personal data (and related preferences): for each piece of personal data the properties and status of the parametric obligation could be different.

We are currently investigating how to provide incremental details on managed parametric obligations via graphical tools that can drill-down the relevant information. For example, given a parametric obligation, an administrator can obtain summarised information about the set of related data and preferences this obligation is associated to (e.g. the size of the set), the percentage of times where this obligation has been "enforced & satisfied" and the percentage of times where it has been "enforced & violated". For each category, the administrator can then dig down the details by potentially focusing on specific cases along with the status of associated personal data and privacy preferences.

6 Next Steps

Our model of parametric obligations and the extension of our obligation management framework must be properly researched and refined. This includes: formalizing the format of parametric obligations; designing the engine that processes these obligations; ensuring that our system evolved towards a hybrid system that can support both "traditional" obligations and parametric ones.

We plan to do this work in the context of the PRIME project and an HP Labs project. We are also planning to get further feedback and input by engaging in technological trials with customers.

7 Conclusions

Privacy management is important for enterprises that handle personal data. In particular the management and enforcement of related privacy obligations is a green area open to research and innovation. In this paper we provided an overview of our R&D work done in this space to explicitly represent, schedule, enforce and monitor privacy obligations – in a flexible and customizable way.

The prototype that we have built and its integration with both the PRIME system and the HP identity management solution showed the feasibility of our approach: this also helped us to further understand this space and highlight a potential scalability problem. This problem is particularly relevant when large amounts of personal data have to be processed: in this context, our current system will generate a large amount of privacy obligations with a consequent management overhead. In this paper we described in more details this issue and the causes.

The lessons we learnt during the integration phase and the feedback we received from third parties helped us to better understand how users will specify their privacy preferences and how privacy obligations should be expressed and generated within an enterprise. As a consequence, in this paper we introduce the concept of parametric obligation as a way to drastically reduce the number of managed obligations and allow the obligation management system to scale. We described our current thoughts on how parametric obligations can be implemented and how they can coexist with "traditional" obligations in a hybrid obligation management model - to provide the right blend of scalability and customisation. Our work is in progress. Further research and development is going to be done in the context of PRIME and an HP Labs project.

References

1. Rotemberg, M., Laurant, C.: Privacy International: Privacy and Human Rights 2004: an International Survey of Privacy Laws and Developments, Electronic Privacy Information Center (EPIC), Privacy International. http://www.privacyinternational.org/survey/phr2004/, 2004
2. OECD: OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data. http://www1.oecd.org/publications/e-book/9302011E.PDF, 1980
3. Online Privacy Alliance: Guidelines for Online Privacy Policies. http://www.privacyalliance.org/, Online Privacy Alliance, 2004
4. Casassa Mont, M.: Dealing with Privacy Obligations: Important Aspects and Technical Approaches, TrustBus 2004, 2004
5. Casassa Mont, M.: Dealing with Privacy Obligations in Enterprises, HPL-2004-109, 2004
6. PRIME Project: Privacy and Identity Management for Europe, European RTD Integrated Project under the FP6/IST Programme, http://www.prime-project.eu.org/, 2005
7. Hewlett-Packard (HP): HP OpenView Select Identity: Overview and Features, http://www.openview.hp.com/products/slctid/index.html, 2005
8. Casassa Mont, M., Thyne, R., Chan, K., Bramhall, P.: Extending HP Identity Management Solutions to Enforce Privacy Policies and Obligations for Regulatory Compliance by Enterprises - HPL-2005-110, 2005
9. Casassa Mont, M.: A System to Handle Privacy Obligations in Enterprises, HP Labs Technical Report, HPL-2005-180, 2005
10. IBM: The Enterprise Privacy Authorization Language (EPAL), EPAL 1.2 specification. http://www.zurich.ibm.com/security/enterprise-privacy/epal/, IBM, 2004
11. Karjoth, G., Schunter, M.: A Privacy Policy Model for Enterprises. IBM Research, Zurich. 15th IEEE Computer Foundations Workshop, 2002
12. OASIS: Extensible Access Control Markup Language (XACML) 2.0, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml, 2005