Privacy policies, tools and mechanisms of the future

Vincent Naessens[1], Mehmet Tahir Sandikkaya[1], Jorn Lapon[1], Kristof Verslype[2], Pieter Verhaeghe[2], Girma Nigusse[2], Bart De Decker[2]

[1] Katholieke Hogeschool Sint-Lieven, Department of Industrial Engineering
Gebroeders Desmetstraat 1, 9000 Gent, Belgium
[2] Katholieke Universiteit Leuven, Department of Computer Science,
Celestijnenlaan 200A, 3001 Heverlee, Belgium

Abstract. Although many believe that we have lost the battle for privacy, protection of what's left of the user's privacy is all the more important. Not only should a user be able to minimize the disclosure of her personal data, she should also have rights to decide what happens with her data once they have been disclosed. In order to minimize user interaction when deciding whether or not to reveal personal data, privacy policy languages were developed. However, these languages are inadequate and cannot properly deal with the complex interactions between users, service providers, third parties, identity providers and others. Also, tool support for composing and verifying these policies and mechanisms for enforcing them are lagging behind. This paper argues the need for better privacy policies and proposes some solutions. Throughout the paper, our statements are applied to three sample applications in three different domains: e-health, banking and social networks.

1 Introduction

Some decades ago, the Internet was mainly used as a means to publish and disseminate information. Companies and organisations built static HTML-pages that could be interpreted by simple web browsers. A small percentage of individuals also hired web space to upload personal information. At the end of the twentieth century, the number of Internet users grew rapidly and application developers started to build more advanced applications. Today, companies and organisations in many domains offer interactive and highly dynamic services to an ever extending community of customers. Multiple characteristics of today's Internet services have an impact on the privacy of individuals. First, individuals are usually tempted into *releasing valuable personal information* during service consumption. This information can often be linked to data that uniquely identifies the individual. Moreover, the user looses all control over the data once they are released. This may introduce serious privacy dangers. For instance, an e-shop can draw conclusions about the health status or political preferences of an individual based on his purchase behaviour. This can lead to discrimination, especially when this information is sold to external parties (such as employers and/or insurance

companies). Moreover, the information that is released can also be used as evidence in a court in certain countries. An individual was already condemned to prison after he caused a car accident. One piece of evidence was information about his drinking behaviour that he published some months before the accident at a social networking site. Second, *many companies that offer advanced Internet services also collaborate with other entities.* E-shops typically collaborate with delivery companies and payment service providers; commercial e-health providers (e.g. companies that offer monitoring services to patients) can cooperate with insurance companies and hospitals; social networking sites can collaborate with advertisement companies and police forces. It is often unclear towards service consumers what personal information is actually revealed to which party that is involved in the application. Moreover, collaborations between organisations can be highly dynamic. For instance, an e-shop may decide to collaborate with another delivery company to reduce costs; the delivery company can, on its turn, be trading with another company or can delegate some tasks to a third party. Third, *personal information about other individuals* is released to service providers in many Internet applications. This often happens without (explicit) consent of that individual. For instance, users can post messages, tagged pictures or phone numbers on social networking sites that can compromise other individuals' privacy. More subtle examples can be found in other domains. A consumer can order a gift for a friend at an e-shop and needs to release his name and his address during that transaction; a doctor can add new patient records to a centralized e-health database, etc. Although the information is not necessarily made publicly available, it allows Internet providers to collect information about individuals without their explicit consent. In some cases, those individuals are even not aware that valuable personal information is collected by a company. Fourth, service providers permanently try to *optimize their business by compiling more advanced profiles.* Hence, they try to personalize their services. This often depends on the amount, type and quality of information that can be collected. In some cases, it can be beneficial to the consumers. Accurate medical profiles surely lead to better health provisioning; a book shop or travel organisation can offer products within the user's domain of interest. However, users are often not aware about personalized advertising practises and the information that is collected, stored and processed. Moreover, users often have no impact on the quality of information that is used by service providers when building a profile. Inaccurate or even false information may result in misleading profiles.

It is clear that many of today's Internet service providers offer services that support complex interactions between stakeholders. Consumption of these services can have an impact on the user's privacy. On the contrary, current privacy policies, tools and mechanisms offer only support for simple interactions and are quite static. Hence, they lack support to deal with the increasing complexity of interactions in advanced electronic services. This paper defines important challenges for future privacy policies and discusses mechanisms and tools to enforce those policies.

The rest of this paper is structured as follows. Section 2 classifies important challenges for policy specification. Section 3 defines a policy scheme.

Tools for policy enforcement are described in section 4. Section 5 points to related work. This paper concludes with an overview of the major contributions and directions for future research.

2 Challenges for privacy policies

2.1 Disclosure of information

Information may be disclosed by and to different types of parties. Some cases are particularly of interest with respect to privacy.

Disclosure of one's own personal data. In this case a user discloses information towards another party such as a service provider. Here, preserving the user's privacy may be addressed by controlling the disclosure of personal information (at the user's site); the outcome may be different depending on the kind of service and/or the service provider. For instance, an e-bank is allowed to request the user's bank account, while an e-shop is not.

Disclosure of personal data by another party to another party. We distinguish between disclosures to the public domain and to a service provider.

Disclosures by a friend to the public domain. In social networks users can tag other users. They can publish information about friends, possibly revealing personal information about them. Moreover, once one becomes a "friend", one has no control over whether one is listed in the friends-list of that friend or not. It should be possible to disallow that one's name is recorded in that list. As such, a user should have actual control over what a friend may publish about her. Current systems lack a proper way to define rules on who is allowed to access/publish personal information of others. Their privacy settings are often ad hoc and not user centric enough. Furthermore, especially in the case of social networks, privacy settings should by default be very restrictive. It becomes even more difficult when it involves data of multiple users, such as chat logs. In such cases, all parties should agree on a policy specifying whether or not that data can be distributed and/or published and under what circumstances.

Disclosures by a party to another party. Service providers may disclose information about their clients to third parties. Often the identity of that third party cannot be disclosed because of the commercial position of the company. However, a user should still be able to limit the transfer of his personal information by a service provider to third parties. It should be possible to express privacy rules for the use, disclosure, transfer and storage of personal information and these rules should be in force during the complete lifetime of that personal information. Hence, whenever a collector of personal data would like to further transfer the data to others, it should still be with the user's consent.

2.2 Trustworthiness

Users are willing to release more personal information (and certainly more sensitive PI) to service providers they trust than to service providers that have a questionable reputation or that are unknown. For instance, accredited e-health applications are usually considered more trustworthy than commercial service providers.
As such, privacy policies should be able to specify the trust level of a particular service provider, and restrict the disclosure of personal information depending on the assigned trust level.
For new service providers (SPs), the trust level may be chosen by the user or be derived from "privacy seals" awarded to these SPs by external accreditation organisations, based on regular audits.
On the other hand, service providers should also be able to trust the collected personal information. Often, users are required to reveal more personal information than strictly necessary, to allow the SP to verify the truthfulness of other information, or to be able to take corrective or disciplinary measures when abuse is detected. An SP may be willing to collect less personal information when it is ensured of the correctness of the collected information (e.g. because it is certified by a trusted party, which may also take certain liabilities).

2.3 Profiling/personalization/negotiation

Most service providers will maintain profiles about their users. These profiles are useful to personalize the SPs' services. For instance, an e-bookshop can present a customer a list of books that are similar to the ones bought before, or that have also been bought by buyers of the same books. However, such profiles also imply risks of revealing too much of the users' preferences, their range of thought, political ideas, or health status.
Hence, users should be able to control which personal information and which transactional data can be added to their profiles. For instance, the user's privacy policy may specify that only the "type" of book ("novel", "thriller", ..., but not the title) or the author of the book can be added to her profile, except for medical books which should be banned from the profile.
A step further is to disallow SPs to compile profiles altogether; instead, the user's profiles are maintained at the user's site, and SPs are allowed to query these profiles. This has two advantages: the local profiles are more extensive since they may contain transactional data of interactions with different SPs and the privacy policy can restrict the query results (e.g. delete sensitive information, summarize, generalize, etc.).

2.4 Possibility to access, update and delete data

Many countries have a privacy legislation that specifies that users have the right to access, rectify or delete their personal information at any time. However, this right is usually only provided via written letters,

and not via an e-service. The only exception is probably a subscription to a newsletter. Here, the user can usually unsubscribe from the mailing list by clicking on a link, which will remove the user's personal information (i.e. her email address) from the mailing list. However, this technique is –in general– inadequate and cannot be applied to other personal information that is collected and stored by SPs.
Therefore, a SP's privacy policy should specify how users can discover what personal information is kept by the SP, how this information can be accessed, modified and possibly deleted. Moreover, users should also be able to query what has been done with their data, who has accessed it, to which parties their information was forwarded, etc.
Note that exceptions exist. In some cases (e.g. e-health applications) users are usually not allowed to directly access their health records; also, the legislation may require that certain data should be kept for several years. In this case, deleting personal information means that the SP is no longer allowed to use that data.

2.5 Context information

Current privacy policies are context-oblivious. It is not possible to allow or restrict the disclosure of personal information based on certain context parameters: date/time, location of user and SP, communication channel (wireless/wired, (un)protected, authenticated/anonymous, ...), special conditions (e.g. emergency situation, legal investigation, ...), value of the transaction (high/low), size and content of current profile, etc.
It should be clear that the context in which the transaction takes place may ultimately influence the decision whether certain personal information can or cannot be disclosed. For instance, goods (e.g. movies) with export restrictions may require that users disclose (prove) the region in which they live; in case of a medical emergency, few restrictions on the disclosure of personal information (PI) will exist; when the user's profile is already extensive, the policy may not allow further discloser of PI, and hence, the user will have to create a new account to keep her profile as small as possible.

3 A Policy Scheme

This section presents a generic structured approach to define *privacy policies* and *privacy preferences*. First, we describe an approach to structure the privacy policies of the service provider. Next, we define how users can structure their privacy preferences. Finally, an approach to classify personal information items (PIIs) is presented.

3.1 Privacy policies of the service provider

Many organisations that offer electronic services already define and publish a privacy policy. Such a privacy policy typically expresses what personal information is collected, processed and possibly propagated to third

parties. However, current policy languages are too coarse-grained to support the complexity of interactions in advanced electronic services. First, many service providers try to optimize their business by personalizing services. The user may receive discounts based on previous transactions, the type of personal information that she is willing to disclose, her reputation, etc. Second, many companies offer multiple services. The type of information that an individual needs to release depends on the service that is accessed/consumed. For instance, individuals can *subscribe to a newsletter* or *book a flight* at a travel agency. The former requires that the user needs to release her *e-mail address* whereas the latter implies that she discloses (and possibly proves) more intrusive personal identifying information (such as her *name*, *address*, *age*, ...).

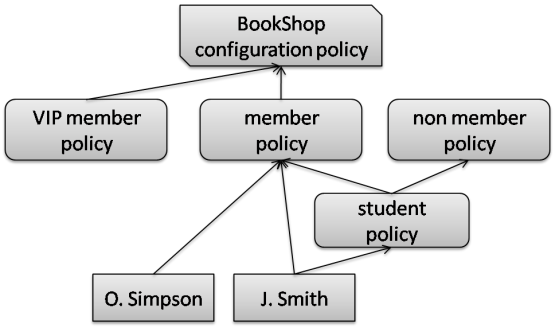

Fig. 1. Overview of the policy scheme of an *e-shop*.

The privacy policies of the service provider can be split into four categories:

- The **service provider configuration policy** defines the rules that will be applied to all services of one service provider. For instance, an e-shop can define an upper limit to the retention time of personal information after it is released by a customer. This policy can be refined by other policies. For instance, the e-shop may define that it will store information for a maximum of *12 months* whereas it will store information related to the ticketing service only for *6 months*.
- The **service configuration policies** define the service-specific rules that apply to all users. For instance, a travel agency can define that information related to purchases will not be disclosed to other parties without explicit consent of the customer. Similarly, a service provider that offers multiple communication services (chat service, e-mail service, phone service, ...) can specify to keep logs related to phone conversations for *18 months* whereas chat messages will only be kept for *1 month*.
- The **group-specific policies** allow for a differentiation of the privacy policy based on the group to which a user belongs. A user

can either be a *VIP member*, a *normal member* or a *non-registered user* (i.e. a *guest*). VIP members typically need to disclose more personal information and allow for processing more detailed information whereas non-registered users can remain anonymous. In return, the former group gets more benefits (discounts, more disk space, personalized information, etc).

- The **user-specific policies** allow to define rules that apply to a particular individual (or a specific third party). For instance, an e-shop can build profiles based on the history of transactions that occurred during the last 12 months. However, each VIP member can choose to increase the retention period to get an even better quality of service. For instance, user X can allow the e-shop to store all personal information for a period of 2 years. Similarly, an e-health service provider will not propagate medical information by default to anybody except in case of emergency. However, each patient can compose a list of individuals (e.g. a few doctors and/or family members) who can access certain personal medical information at any time. To support user-specific policies, users must be able to communicate their personal preferences to the service provider. For instance, the service provider can send a template to new users. Note that, in some cases, users may want to modify their personal preferences. Hence, a user-specific policy is typically more dynamic than a service (provider) configuration policy.

3.2 Privacy preferences from the user

Individuals typically define privacy preferences. Many users want to update their privacy preferences after awhile. However, it should also be possible to constrain the modification of privacy preferences. For instance, parents must be able to define the privacy preferences on behalf of their children. A child must not be able to change those settings without explicit consent of their parents. For instance, a parent may define that it is forbidden to disclose a visa card number during any interaction between the child and a service provider.

The privacy preferences at the user side can be split into four categories:

- The **user configuration policy** defines privacy preferences that apply to all domains. The policy can define that the user's *unique national identification number* is never revealed without explicit consent. Similarly, the user's *gender* and *country* may be revealed to any service provider without prior consent. The policy may also specify that the user will assign a domain to a service provider the first time a service is used.
- **Domain specific policies** allow to differentiate privacy preferences based on the domain to which the service provider belongs. Typical examples are the `medical` domain, the `commercial` domain, the `governmental` domain, etc. As such, the policy can allow to release the user's *blood group* to an e-health service whereas disclosing that attribute to a service provider that belongs to the commercial domain is forbidden. Similarly, the policy may specify that the *name*

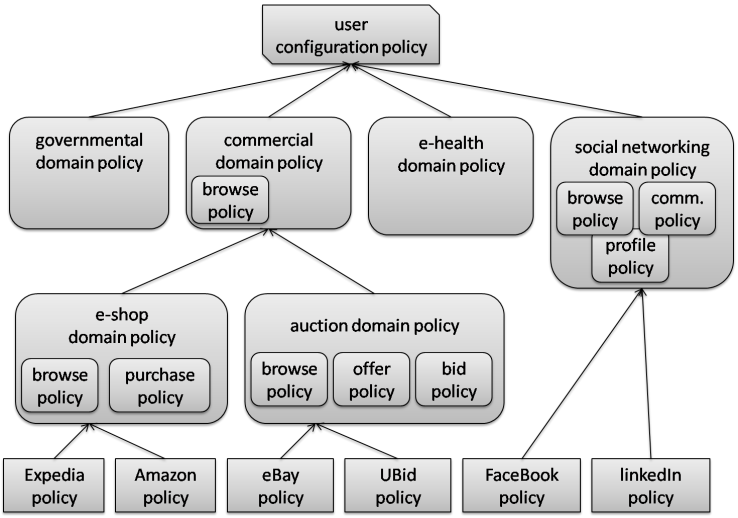

Fig. 2. Structure of the user's privacy preferences.

and *e-mail address* can be released without restrictions to any governmental service provider whereas the user's explicit consent is required when those attributes are requested by a commercial service provider. The policy can also specify that *no personal information* will be released to commercial service providers for which no acceptable purpose is given.

- **Service specific policies** allow for differentiation of privacy preferences within a particular domain based on the specific service that is consumed. For instance, a user may define different policies for multiple services offered by e-shops. An individual is typically willing to release more information when buying a book than when browsing a list of books that are available at the book shop.
- **Service provider specific policies** allow for differentiation of privacy preferences within a particular domain based on the specific service provider. Some service providers may be trusted more than others (because of past experiences or special seals awarded by trusted third parties). The policy should allow to release more or less information to certain service providers. For instance, whereas e-mail addresses are usually kept private, it may be revealed to a trusted e-shop in order to subscribe to a newsletter. Also, the policy should allow the user to specify what transactional information (e.g. the goods bought, ...) can or cannot be included in the user's profile. Another example is the release of personal information on social networking sites. The policy may specify that some specific friends can release some personal information about the user (e.g. her e-mail address) but only on *LinkedIn* and not on *FaceBook*.

Note that a *user configuration policy* is typically more restrictive than *a domain specific policy*. An individual can define that unique identifiers may not be released to any service provider. The policy for the governmental domain can override this rule. Moreover, *service provider specific policies* are more dynamic than *domain specific policies* and *user configuration policies*. The latter will only change frequently for a short period after initialisation (or a learning process by the privacy agent) whereas the former may change over time (e.g. based on the SP's reputation). For instance, if the service provider's site has been hacked, individuals may make the *service provider specific policy* more restrictive for some period (until the company can convince individuals that is has taken appropriate countermeasures). Note also that multiple service providers can be involved in one application. A typical example is a book shop that relies on a payment service for financial transactions. Hence, multiple domain specific policies need to be taken into account.

3.3 Classification of personal information items

To reduce the complexity of instantiating privacy policies and/or privacy preferences, **personal information items (PI-items)** should be classified according to an (extensible) ontology. This classification should start with the **PI-class name** to which it belongs followed by a **PI-type name** (e.g. *myBloodType* is a `medical.bloodtype` while *myFaceBookEmail* is a `social.emailaddress`). A `null` class is used for personal information items that exceed a particular domain (e.g. *mySurname* is a `null.identity.surname` and *myHomeStreetName* is a `null.home.streetname`). It should also be possible define **structures** in which PI-types are aggregated. Structures of interest are those that present a higher level of confidence (or trustworthiness), because they are (self-)signed, incorporated in certificates or embedded in (anonymous) credentials. In all three cases, a signature is present and the trustworthiness of the signed personal data will depend on the issuing procedure of the signer (whether the signed data has been verified properly) and on the liabilities taken by the signer. Multiple values can be assigned to PI-classes, PI-types and PI-structures:

- A **sensitivity level** (SL) expresses *how important a PI-class or PI-type is to the user, from a privacy perspective*. PI-types without an SL inherit their SL from the class they belong to. SLs will be taken into account by the privacy agent (cfr. further) when personal information needs to be disclosed. The agent will select the PI-items or structures that have the lowest accumulated sensitivity.

- A **threshold of minimal trustworthiness** (TMT) of the receiver of PI-items expresses *how trustworthy a receiver of PI should be to be allowed to receive that type of PI*. For example, `medical.genetic.*` items can only be communicated to service providers that have been accredited by the national social security organisation. Again, PI-types inherit the TMT from their domain if no TMT is defined. Trustworthiness will always be expressed in terms of seals or credentials.

4 Tool support

We assume that applications are **designed to be privacy friendly**. Formal models for expressing privacy requirements can be used during the design phase of the application. Also, **code annotations** should indicate where personal information is collected and what the purpose of the collection is. A special-purpose **plugin** for an integrated development environment (IDE) can then extract these annotations from the code and compile the basic structure for the application's privacy policy. Also, other plugins can trace where the personal information is processed in the code, where it is stored and retrieved from databases and where it is sent and received through communication channels. This will help the administrator to compile a consistent privacy policy, but also the auditors who have to assess the organization's compliance with that policy.

4.1 Policy Management Tools

A generic policy management tool should aid both the compilation of privacy policies and the verification of these policies.
In this paper, a distinction is made between policy tools for administrators and tools for the user (customer). All tools should be able to define a *hierarchy of policies*: for administrators, the hierarchy starts at the root with the organisation, then the website, the applications, (web)services, the groups and the specific users; for users, this hierarchy starts at the root with general rules, then domain rules, the rules for services and finally service provider specific policy rules. Moreover, the tools should also support *a hierarchy of personal information items*.
A typical **user's privacy policy tool** should have the following functionalities:

- The tool should also allow the users to specify which personal information types/classes/structures can be communicated with which service provider domains. For instance, `medical` personal information can only be disclosed to service providers from the `medical` domain.
- Users should be able to specify how a domain is assigned to a service provider (SP) or to its different services. This can be done by the user on the first contact with that SP, or automatically based on certified seals presented by the SP. For instance, a bookshop could have a *Browse*-service in which a user can view the catalogue and a *Buy*-interaction, in which one or more books can be bought. *Browse* would be assigned `Anonymous` while *Buy* would belong to `Home/Work` and `Financial`);
- The tool should allow the users to assign sensitivity-thresholds to domains and service providers. Whenever this threshold is exceeded, the privacy agent will intervene and request the user's explicit consent.
- Users should be able to specify per service provider and per service, which PI-items and PI-structures can or cannot be exchanged during a particular interaction; radio buttons would allow to change a `block` into `allow` or `request consent`.

- The tool should allow users to specify whether release of personal information by other parties (friends) is allowed or not, and whether the user's consent is required or not.
- The tool should allow users to specify what should happen with transactional data collected by SPs. Whether this data can be added to the user's profile and in what format (complete, partial, summarized, ...). For SPs with which the user has not dealt with previously, it is often unknown which data is actually collected. Hence, an option `userConsent` should be available that will request the user's consent at the end of a service interaction.

A typical **administrator's policy tool** should start from the basic structure extracted from the annotated code.

- The tool should allow administrators to **refine the purpose** of the personal information collection, how that data will be processed and stored, with whom the data will be shared, etc.
- Administrators should be able to specify the **minimal trustworthiness** of PI-items or -structures. For instance, they can list the TTPs or CAs by which the PI has to be certified.
- The tool should allow administrators to restrict the disclosure of personal information of someone else, and make this disclosure conditional to the consent of the person concerned.
- Administrators should be able to specify how customers can request access to their personal data and what rectification is allowed.

Privacy policy **verification** tools allow users or service administrators to check for conflicts and verify whether the policies cover everything. Also, the tool should show graphically which personal information will be exchanged during a particular interaction. When conflicts are discovered, the user/manager would get the opportunity to resolve the conflict or leave it and have it resolved at run time.
Finally, appropriate tools are necessary to distribute these privacy policies to the different components that need them.

4.2 Runtime support mechanisms

A number of mechanisms can assist the user at runtime with the purpose of increasing the user's privacy.
Policy enforcement and **decision** points are necessary wherever personal information is disclosed. Therefore, a generic **privacy agent** [1] will assist the user whenever PI needs to be disclosed. If the agent cannot unambiguously determine the action to perform, the user should be queried. A checkbox should allow to extend the privacy policy in accordance to the user's reply.
A profile manager keeps track of what data has been disclosed to whom under what pseudonym. The manager maintains a copy of the user profiles that are collected by the service providers and, hence, can warn the user when she risks to loose too much of her privacy. Ideally, the user should be able to forbid the service provider (SP) to compile a profile, and instead have the SP query the locally kept profile. The profile manager can then decide which information to disclose (depending on the trustwortiness of the SP).

Negotiation protocols should try to match the service provider's policy to the user's policy. Based on this matching process, a selection of the personal properties to be disclosed is made. Aspects such as privacy friendliness of different credentials and trustworthiness of both the service provider and the different credentials are taken into consideration.

5 Related work

5.1 Current policy languages

Current policy languages that are related to automating the private data disclosure on the internet are P3P [2] and APPEL [3]. P3P is a policy language that creates a machine-readable interpretation of the privacy practices of the service providers. On the other hand, tightly coupled with P3P, APPEL offers a machine-readable language where users can define their privacy preferences. XPref [4], which covers APPEL, is more compact and flexible than APPEL.
At the policy side, using P3P, service providers can define their purpose on collecting personal data, the recipients of the personal data, the retention time for erasure and finally a method and contact to resolve conflicts manually. P3P cannot indicate the domain of the service provider, technical specifications of the connection or the methods to enforce the policies.
At the preference side, using APPEL or XPref, users can define actions for the statements based on data collection purpose, data recipients and retention time. Still, domain support is lacking and users cannot indicate technical requirements.
However, the aim of P3P was to interpret the natural language privacy policies to machine-readable form; mentioned additional features seems to be helpful in the privacy preserving context. Extending the languages or merging P3P/APPEL/XPref policies with another set of configuration policies may be a solution.

5.2 PrimeLife

PrimeLife [5] is a project that builds on the experience of the PRIME [6] project. PrimeLife aims at bringing sustainable privacy through addressing the problem of the digital footprints left in one's life time. In these projects policies are emphasized as an enabler for privacy, identity, and trust management. These projects help users to keep control over their personal data through privacy-aware access control solutions named the PRIME policy languages [7]. In addition to the traditional access control policies, the PRIME policy languages define release policies and data handling policies [8]. The release policies govern the release of properties, credentials, and personal identifiable information (PII), and also restrict these release conditions. The data handling policies regulate how users' private data will be handled by service providers. Most importantly, PRIME's XML based policy languages exploits cryptography [9], hence, supports attribute-based restrictions, credential definition

and integration, and anonymous credentials [10]. These projects have presented privacy policy related prototypes for Web browsers, such as bookmark lists with icons for privacy preferences (PrivPrefs), PRIME Console, PRIME `Send Personal Data?`, PRIME assurance evaluation (Privacy Functionality Check), and PRIME Data Track [11, 12]. For instance, the PRIME visiting cards can automatically disclose data if the user's preferences are fulfilled, otherwise, a confirmation box, i.e. `Send Personal Data?` will popup to ask for additional preference settings and confirmation. Thus, using `Send Personal Data?`, individuals are able to define which data to whom and for what purpose should be disclosed [12]. On the other hand, data tracks serve as a history function of the PRIME system, where users can look up what private data they have disclosed to which service provider [11].

6 Conclusions and future work

This paper touches the challenges for future privacy policies, which are currently not dealt with in the available privacy policy languages. Users have a clear need to express their privacy preferences in terms of sensitivity of their personal information and the trustworthiness of the service (provider). Also, disclosures of personal information by friends or PI-collectors should be subject to the users' consent. Current privacy policies cannot deal properly with sophisticated mechanisms such as anonymous credentials, neither can service providers impose minimal trustworthiness qualities on the collected data ("The data must have been certified by a trusted credential issuer"). User profiles are usually kept outside of the privacy policies; nevertheless, they often carry an enormous privacy risk. Therefore, users should be able to express what personal/transactional data can be included in such profiles, or whether such profiles should be maintained at the user's site instead. Context (location, date and time, communication channel, transaction value, etc.) may influence the decision on whether personal information is to be disclosed or not.

Future privacy policy schemes should allow to easily classify personal information (PI). PI belongs to a certain type, has a particular sensitivity level and can only be communicated to service providers which are trustworthy to handle this kind of PI. Moreover, PI can be aggregated into structures with additional trustworthiness properties, because they have been certified by external parties. The policy languages should allow for fine-grained specifications of disclosure/collection of PI. To make the management of policies easier, the paper proposes a hierarchy of privacy policies, both for service providers as for users. For SPs, the policies go from SP configuration, service configuration, over group-specific to user-specific policies. Likewise, user policies are structured from configuration policies over domains, services to service provider specific policies.

Evidently, proper tools are necessary to allow for easy management and verification of the policies. Also, proper runtime mechanisms are necessary to enforce these policies and allow for policy negotiations. The user's privacy is best protected when as little as possible personal information is disclosed. Hence, the maintenance of local profiles (on the user's host) is preferable to profiles maintained by SPs.

A hierarchy of policies allows for a flexible compilation of these policies. However, conflicts are more likely to occur and need to be dealt with. This is outside the scope of this paper. Also, the structure and working of the privacy agent is subject to future research.

Acknowledgements

This research is partially funded by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy and the Research Fund K.U.Leuven, the IWT-SBO projects DICOMAS and ADAPID and the IWT-Tetra project e-IDea.

References

1. S. Gevers and B. D. Decker, "Privacy friendly information disclosure," in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops (Meersman, R. and Tari, Z. and Herrero, P., eds.), Lecture Notes in Computer Science*.
2. L. Cranor, B. Dobbs, S. Egelman, G. Hogben, J. Humphrey, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle, M. Schunter, D. A. Stampley, and R. Wenning, "The platform for privacy preferences 1.1 (p3p1.1) specification," November 2006. `http://www.w3.org/TR/P3P11/`.
3. L. Cranor, M. Langheinrich, and M. Marchiori, "A p3p preference exchange language 1.0 (appel1.0)," April 2002. `http://www.w3.org/TR/P3P-preferences/`.
4. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Xpref: a preference language for p3p," *Computer Networks*, vol. 48, no. 5, pp. 809–827, 2005.
5. `http://www.primelife.eu/`.
6. `https://www.prime-project.eu/`.
7. C. Ardagna, M. Cremonini, S. D. C. di Vimercati, and P. Samarati., "A privacy-aware access control system," *Journal of Computer Security (JCS)*, vol. 16, no. 4, pp. 369–392, 2008.
8. C. Bournez and G. Neven, "Draft requirements for next generation policies," 2008. Draft version (PrimeLife Project).
9. P. Samarati, "First research report on research on next generation policies," 2009. Version 1.0 (PrimeLife Project).
10. C. Bournez and P. Bichsel, "First report on standardisation and interoperability overview and analysis of open source initiatives," 2008. Deliverable (PrimeLife Project).
11. J. S. Petterson, S. Fischer-Hbner, N. D. J. Nilsson, M. Bergmann, T. Kriegelstein, S. Clau, and H. Krasemann, "Making prime usable," in *In Proceedings of the Symposium of Usable Privacy and Security (SOUPS)*, 2005.
12. J. S. Pettersson, "Hci guidelines," 2008. Final version (Prime Project).