Towards a UML Profile for Privacy-Aware Applications

Tania Basso[1], Leonardo Montecchi[2], Regina Moraes[1], Mario Jino[1], Andrea Bondavalli[2]
[1] State University of Campinas (UNICAMP)
Campinas, SP, Brazil
{taniabasso@ft, regina@ft, jino@dca.fee}.unicamp.br
[2] University of Firenze (UNIFI)
Firenze, Italy
{lmontecchi,bondavalli}@unifi.it

***Abstract*—Personal information is continuously gathered and processed by modern web applications. Due to regulation laws and to protect the privacy of users, customers, and business partners, such information must be kept private. A recurring problem in constructing web applications and services that protect privacy is the insufficient resources for documenting them. As web applications must be developed consistently with the statements of the privacy policy in order to enforce them, a structured documentation is necessary to model privacy protection during application design. To contribute with solutions to this problem, in this paper we propose a UML profile for privacy-aware applications. This profile helps building UML models that specify and structure particular concepts of privacy and, consequently, improve privacy definition and enforcement. After introducing the main privacy concepts, we describe how they are represented in the UML language. The profile's ability to model statements of realistic privacy policies is then demonstrated on a case study.**

Keywords — privacy; conceptual model; UML Profile.

I. INTRODUCTION

Personal information, in different forms, is continuously gathered and processed by modern web applications. This includes data that is required for accessing a certain service (e.g., email address, credit card number); additional personal data that the user may provide for an enriched user experience (e.g., pictures, connections to social networks); and data that can be automatically gathered by the service provider (e.g., usage pattern, approximate location). Once this information is made available it is no longer under the control of their owner regarding how it is used, which raises relevant privacy concerns. In our context, the term *privacy* means "*the right of an entity to be secure from unauthorized disclosure of sensible information that are contained in an electronic repository"* [1], and our focus is on information managed by web applications.

Nowadays, protecting the privacy of information manipulated by web applications and services is essential. The main reasons are related to regulation laws (the companies and organizations that hold private data have the obligation and responsibility to protect them), and competitive differentials (the more a company protect the privacy of its customers and business partners, the more credibility it gets).

A recurring problem in constructing web applications and services that protect privacy is the insufficient resources for documenting them [2]. The lack of integration of privacy protection in the application design and development makes privacy protection difficult, since privacy mechanisms have to be devised based on both the application and the privacy policy. Approaches for modeling privacy views of system applications through a structured model can help to better describe policy statements and resources or technologies to be used in order to enforce them. By using a popular language like UML [3], it would be easier to integrate such privacy information protection within the development process, helping service providers to fulfill the requirements of privacy policy. In this paper, we provide our contribution in this direction, by defining a UML profile for privacy-aware applications.

A UML profile is an extension of the UML metamodel, containing specializations for a certain domain, platform, or purpose [4]. The profile presented in the paper can be used for modeling views of the system including privacy protection concepts. Its main purpose is the documentation of privacy specifications of web applications, helping to structure particular concepts of privacy and improving privacy definition and enforcement. Models created using the profile are meant to be used both during the development phase of a web application, as well as after its deployment. During the development, models created using the profile help developers to keep track of privacy requirements and how they are implemented. After the deployment, the same model can provide a description to users of how the application will handle its private information.

The paper is organized as follows. Section II presents the main related work in the literature. Section III introduces the conceptual model that will form the basis for our profile, and then describes how concepts are actually mapped to UML constructs. The profile is then applied to a use-case in Section IV. Finally, conclusions are drawn in Section V.

II. RELATED WORK

The Unified Modeling Language (UML) [3] is a central resource in the development of modern software systems. The objective of UML is to provide software engineers with tools for analysis, design, and implementation of software-based systems, as well as for modeling other kind of processes. The UML2 specification defines a lightweight mechanism for extending the language, called "profiling". A UML profile is an extension of the UML metamodel containing specializations for a certain domain, platform, or purpose. UML profiling has been widely adopted, leading to a wide range of profiles for different purposes. Among them, the Object Management Group itself has published as OMG standards several UML

profiles related to non-functional properties, e.g. the QoS&FT [5], MARTE [6], and SysML [7].

Before listing the main related work, we note that, in our view, privacy-related policies can be organized in a hierarchy: highest-level policies are described in natural language; lowest-level policies are specified in machine-readable format, and used by the application itself to e.g., perform access control. In principle, lower-level policies describe a refinement of higher-level policies. The authors of [8] identified seven layers: legal, business, process, application, information, system, device, and network. In this perspective, the scope of the work in this paper ranges from the legal to the application layer. The extension to the lower layers is one of the planned extensions of this work.

As opposite to the objectives of this paper, a lot of research has focused on low-level approaches [8]. Such works aim at producing machine-readable specifications, which can be directly used as input for software enforcement frameworks. Most of the work in this category addresses access control, e.g., XACML [9], PRBAC [10], or Ponder [11]. Another notable contribution within this category was P3P [12] and EPAL [13], languages for specifying user privacy preferences in the web domain. EPAL, inclusive, shows a non-normative UML overview over the elements of its policy. A survey of languages at this level of detail can be found in [14].

With the aim of raising the level of abstraction, and integrate privacy protection in system development, some work defined UML profiles for specifying role-based access control properties [15] [16]. However, these works are mainly limited to describe access control policies, and possibly derive code for enforcing such rules.

Contributions to a higher-level specification of privacy concepts are few. Antón and Earp [28] provide a taxonomy based on possible vulnerabilities to privacy, tailored to the specification of privacy requirements of websites. The work from Solove in [27] synthesizes American law and legal literature with the objective to enable courts and policymakers to improve privacy regulations. While some concepts are in common with our work, both these taxonomies are more tailored to privacy violations, rather than privacy policies and enforcement means.

The Privacy Engineer's Manifesto [25] provides a systematic engineering approach (a set of methodologies, models, and patterns) to develop privacy policies. Similarly, the Dataflow Ontological Approach [26] presents an approach based upon data flow modeling, coupled with standardized terminological frameworks, classifications and ontologies to properly annotate and describe the flow of information into, out of and across systems, from a privacy point of view. Both these approaches foresee the use of UML-like diagrams, but mainly for modeling existing data flows, without explicit relations to the software architecture. Also, they do not introduce specific UML extensions for privacy modeling.

The contribution which is the closest to the one presented in this paper is perhaps the work in [8], where the authors, besides proposing the previously mentioned classification of privacy policy layers, describe a conceptual model to express privacy policy requirements and user preferences. While some concepts are shared, the work in [8] does not aim at defining a UML profile, neither at integrating privacy requirements with the development process. This paper focuses exactly on these two aspects. Finally, the authors of [17] focus instead on the comparison between privacy policies in a SOA context. Comparison of privacy policies for checking their compatibility is not explicitly considered in this paper, and will be part of our future work.

III. CONCEPTUAL MODEL

In this section we outline the privacy elements required to provide specifications about how applications should handle privacy. The goal is to have a model of the domain concepts that are required for modeling views of the system where privacy management and protection are applied.

The proposed elements are created based mainly on two main sources: privacy principles and reference models. Regarding the privacy principles we adopted, more specifically, for this conceptual model, the privacy principles described by the fair information practices developed by the Organization for Economic Cooperation and Development (OECD) [18] and the Global Privacy Standard [19]. The guidelines created by the OECD were adopted because they have been used as the model for most of the privacy legislation throughout the world [20]. The Global Privacy Standard was selected because it attempts to develop a single privacy instrument, i.e., a set of universal privacy principles. Regarding the reference models, we based mainly on the ISO/IEC 29100 [21], the ISO/IEC 29101 [22], the OASIS (Organization for the Advancement of Structured Information Standards) Privacy Management Reference Model [23] and the reference architecture proposed by Basso et al. [24].

The ISO/IEC 29100 [21] describes a high-level framework for the protection of Personal Identifiable Information (PII), sets a common privacy terminology, defines privacy principles and categorizes privacy features. The ISO/IEC 29101 [22] describes a reference architecture with best practices for a technical implementation of privacy requirements. It covers the various stages in data life cycle management and the required privacy functionalities for PII in each data life cycle, as well as positioning the roles and responsibilities of all involved parties in information and communication systems development and management. The OASIS Privacy Management Reference Model [23] is a conceptual model which helps understanding and implementing appropriate operational privacy management functionalities and supporting mechanisms capable of executing privacy controls in line with privacy policies. Finally, the privacy reference architecture in [24] was adopted because it describes, through a privacy logical layer, features that should be considered during the development of an application in order to protect the privacy of personal information. Since we want to consider privacy enforcement in our model, these resources could help in the specification of this kind of element. Also, in [24] the authors consider users to inform their privacy preferences about their PII, agreeing or not with the policies or part of them. This feature allows users to make more thoughtful choices about the use of their personal information online and is also interesting for our model.

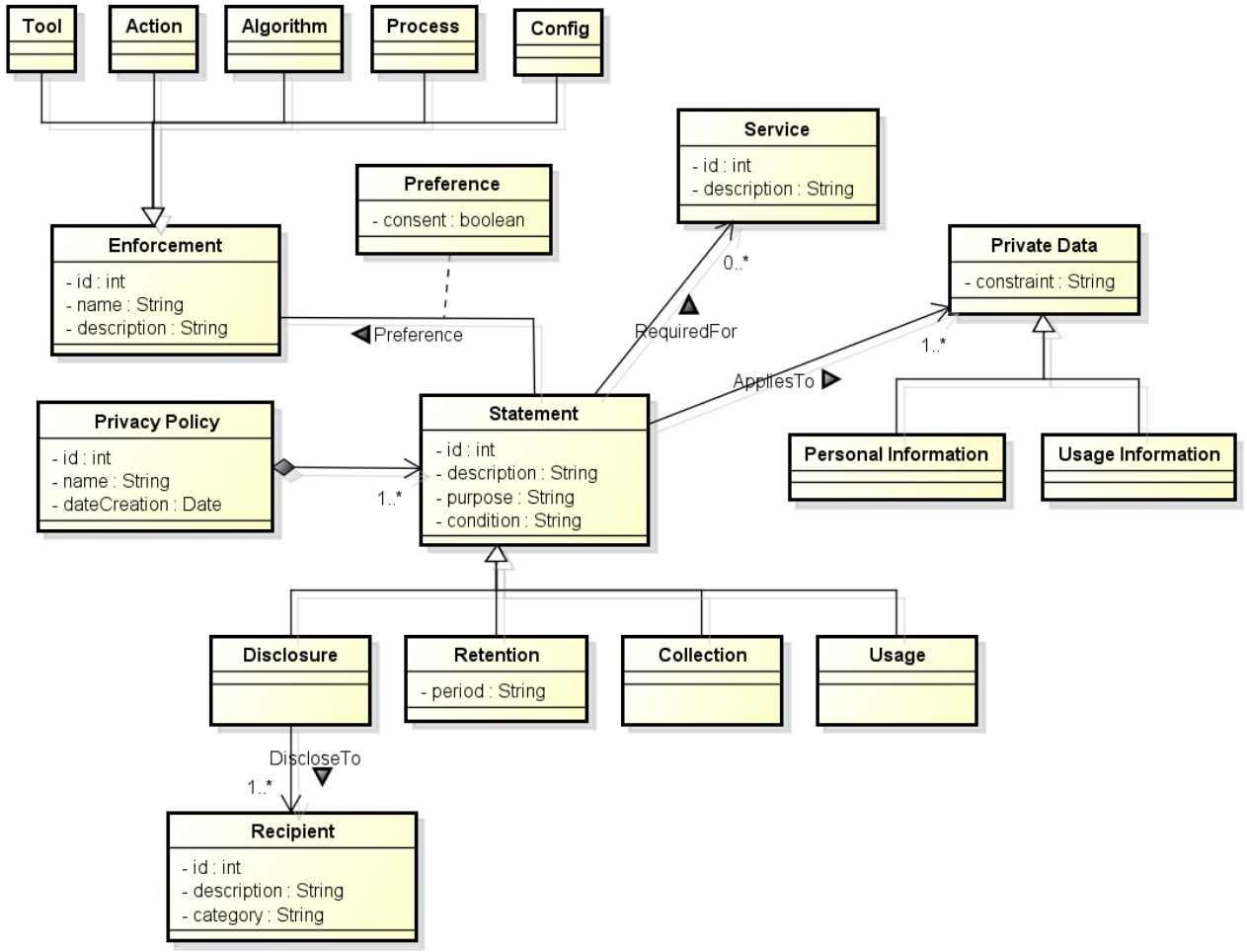

Fig. 1. The Privacy Conceptual Model.

The identified privacy elements and their relations are organized in a conceptual model, which is represented in Figure 1. The model is described in the following.

In Figure 1, the *Privacy Policy* element represents the artifact that must be defined and presented to the user. This is done in order to inform specifically what user's information will be collected and kept confidential, shared or even sold to third parties (i.e., other partners, companies and organizations), as well as the time period they will be stored. A *Privacy Policy* element can be defined by means of its attributes: *id* (identification of the policy), *name* (name of the policy) and *dateCreation* (the date that the policy was created).

The *Privacy Policy* element is composed by one or more *Statements*. The *Statement* element represents the description of one of the rules that are specified in the privacy policy. The attributes that identify the statements are: *id* (identification of the statement); *description* (description of the rule), *purpose* (the purpose for which the data is collected or managed, e.g. research and development, or contacting visitors for marketing of services or products, i.e., it specifies why the action described by the statement is needed); *condition* (prerequisites to be met before any action can be executed, e.g.: before collecting, using or disclosing personal information from a child, an operator must obtain verifiable parental consent from the child's parent).

In addition to the generic *Statement* element, there can be four main specialized types, which describe the kind of action that is performed on private data. These specializations are described in the following, without a particular order: *Disclosure* (specifies which data will be disclosed and to whom), *Retention* (specifies the period the data will be retained), *Collection* (specifies what information, i.e., what private data will be collected) and *Usage* (specifies how the private data will be used). Based on the statements, users inform their privacy preferences.

The *Disclosure* element is related to the *Recipient* one. *Recipient* represents who will access the data to be disclosed. Its attributes are *id* (identification of the recipient), *description* (a textual description), and *category* (used to classify the recipient according to a given taxonomy, e.g., internal or external groups, individual or organization, etc.).

Still in Figure 1, the *Service* element represents the services offered by the company, which are related to statements, i.e., the services that a person can use if he/she provide his/her private information to the applications' company. The *Service's* attributes are *id* (identification of the service) and

description (description of the actions and results provided by the service). There is an association between a *Statement st* and a *Service sv* if the utilization of *sv* is subordinated to the acceptance of *st* by the user.

Also, related to the *Statement*, the privacy conceptual model has the *Private Data* element. It represents the data to be collected and managed by the application according to the privacy policies statements. Its *constraint* attribute can be used to narrow down the kind of private data it represents; this feature is useful to model statements that apply only to data having specific characteristics (e.g., "only *anonymous* data is collected"). The *Private Data* can be of two types: *Personal Information* (information that the user provides to the system) and *Usage Information* (data the system collects, e.g. links accessed, user's actual location, search strings, etc.). The association between a *Statement* and a *Private Data* element keeps track on which private data each statement applies.

Besides the *Statement*, another key element of the privacy conceptual model is the *Enforcement* element. It represents the resources that can be used in order to specify how to enforce the privacy policy statements, respecting the data subjects' preferences. These resources can be represented as *Tool* (e.g. tracking activities tool, intrusion detection tool), *Action* (e.g. allow access, deny access, anonymize data, remove from storage devices, logging actions, encrypt data), *Algorithm* (e.g. k-anonymity – for anonymizing data, RSA – for encrypting data), *Process* (e.g. identity management, access control, auditing), *Config* (e.g. web browser security configurations, software updates, changes in default configurations). The attributes of the *Enforcement* element are *id*, *name*, and *description*, which represent, respectively, the identification, the name and the description of the resource to be used.

Statements can be associated with the resources that are adopted for its enforcement (*Enforcement* elements). The association is performed through the *Preference* relation; such relation has an attribute, *consent*, which is used to specify which kind of user preference (*true* or *false)* that resource relates to. Each statement may be associated to one or more *Enforcement* elements.

From the privacy conceptual model we created the privacy UML profile, to customize UML models including privacy protection. This is explained in the next subsection.

A. Mapping the Conceptual Model to the UML Profile

This proposal considers the integration principles that UML suggested through the profiles idea. Creating a profile allows creating new modeling elements, bringing specific concepts related to privacy protection to the UML language. Profiles are defined using *stereotypes*, *attributes*, and *constraints*. *Stereotypes* are the main construct in a profile and help identifying elements of interest in a model. A stereotype is an extension of an existing UML metaclass or other stereotypes, possibly defining a set of additional *attributes* (i.e., properties). When a stereotype is *applied* to an instance of a UML metaclass, values can be specified for its attributes. Finally, a UML profile may define additional *constraints*, i.e., statements that need to be satisfied for the model to be well-formed according to the profile. Table 1 shows the element of our Privacy UML Profile.

In Table 1, the conceptual elements from the privacy conceptual model (see Figure 1) are mapped to UML *stereotypes*, and listed in the first column; for completeness, also abstract stereotypes are included in the table. Abstract stereotypes cannot be directly used in UML models; their purpose is only to be supertypes of other concrete stereotypes, which can instead be directly used in the modeling process. As stereotypes extend UML *metaclasses* or other stereotypes, the base element of each stereotype is listed in the second column. It should be noted that a stereotype may also extend another newly introduced stereotype. Finally, stereotype *attributes* are listed on the last column.

The <<PrivacyPolicy>> stereotype extends the *Artifact* metaclass, which represents the specification of a physical piece of information that is used or produced by a software development process, or by deployment and operation of a system [3]. Also, it extends the *Class* metaclass. In UML, a *Class* describes a set of objects that share the same specifications of features, constraints, and semantics [3].

The <<Statement>> stereotype extends the *Class* metaclass. In UML profiling, *Class* is often selected as a "default" base metaclass, and it is typically adopted for stereotypes that do not represent software elements as well. The <<Statement>> stereotype is further extended by stereotypes that characterize the nature of the statement of the privacy policy: <<Disclosure>>, <<Retention>>, <<Collection>> and <<Usage>>.

TABLE 1. PRIVACY UML PROFILE.

Stereotype	Base Metaclass or Stereotype	Attributes
<<PrivacyPolicy>>	Artifact, Class	id (int), name (string), dateCreation (date)
<<Statement>>	Class	id (int), description (string), purpose (string), condition (string)
<<Disclosure>>	*Statement*	
<<Retention>>	*Statement*	period (string)
<<Collection>>	*Statement*	
<<Usage>>	*Statement*	
<<PrivateData>>	Property, Class	constraint (string)
<<PersonalInformation>>	*PrivateData*	
<<UsageInformation>>	*PrivateData*	
<<Enforcement>> *(abstract)*	Class	id (int), name (string), description (string)
<<Tool>>	*Enforcement*	
<<Action>>	*Enforcement*	
<<Algorithm>>	*Enforcement*	
<<Process>>	*Enforcement*	
<<Config>>	*Enforcement*	
<<Preference>>	Association	consent (boolean)
<<Service>>	Component, Port	id (int), description (string)
<<Recipient>>	Actor	id (int), description (string), category (string)

The <<PrivateData>> stereotype extends both the *Property* and the *Class* metaclasses. The *Property* metaclass is a structural feature which represents an attribute [3], i.e., a portion of data; the *Class* in this context is seen as an aggregation of multiple elements of information. <<PersonalInformation>> and <<UsageInformation>> are used to mark data that is regarded to as personal information or usage information, and they extend <<PrivateData>>.

The <<Enforcement>> stereotype and its descendants represent solutions that are used to enforce the statements described in the privacy policy. Ideally, the profile should allow the modeler to relate enforcement solutions directly to elements in the model of the software architecture. Depending on the context, an enforcement solution (e.g., an algorithm) may be described by either a structural (e.g., a *Component*) or a behavioral feature (e.g., an *Activity*). In order to be able to cover both cases, our <<Enforcement>> stereotype extends the *Class* metaclass, which is a common ancestor of both the *Component* and *Behavior* UML metaclasses [3]. The <<Enforcement>> stereotype is then extended to better categorize the nature of the enforcement solution: <<Tool>>, <<Action>>, <<Algorithm>>, <<Process>>, <<Config>>. We defined the <<Enforcement>> stereotype as abstract because we want that how statements are enforced is explicitly specified, i.e., we want the user to explicitly use the more detailed enforcement elements. Directly using the generic <<Enforcement>> element would not say much on how the statement is actually enforced.

The <<Preference>> stereotype extends the *Association* metaclass, which specifies a semantic relationship that can occur between typed instances, in our case elements of the <<Statement>> and <<Enforcement>> elements. Such association relates an enforcement solution with a statement for which it is needed, also detailing for which kind of user preference (*opt-in, opt-out*) is actually needed.

The <<Service>> stereotype extends the *Component* and *Port* metaclasses. A *Component* describes a modular part of a system that encapsulates its contents, i.e., without focusing on its internal implementation, but only on the service(s) it provides. A *Port* may be used to specify in more details the services a classifier provides (requires) to (from) its environment. Finally, the <<Recipient>> stereotype extends the *Actor* metaclass. This metaclass specifies a role played by a user or any other system that interacts with the subject.

The *constraints* needed to express our domain concepts are limited to relationship multiplicities (see Figure 1); no additional constraints are included in the profile. The constraints expressing the multiplicities are instead summarized in Table 2.

TABLE 2. PRIVACY UML PROFILE CONSTRAINTS.

For each *Statement* element there must be at least one association with a *Private Data* element.
For each *Disclosure* element there must be at least one asssociation with a *Recipient* element.
For each *Privacy Policy* element there must be at least one containment association with a *Statement* element.

A case study was developed to instantiate the proposed UML profile. The idea is to apply the profile to a realistic and popular privacy policy. It will be addressed in the next section.

IV. CASE STUDY

The case study that we show is derived from the real privacy policy of Google services [29], i.e., we are using a concrete example for evaluating the proposed Privacy UML Profile. Although there are some discussions about Google's privacy policy when describing how it uses personal data gathered from its web services and products [30][31], we decided to model this policy because it is very popular and used by many users and client applications around the world. To be sure that privacy protection is respected, it would be necessary to perform a thorough analysis to identify if the policy complies with the established laws and principles, which is however out of the scope of this paper. Nevertheless, the same would be required also with other policies that are available on the Internet. Therefore, we assume that Google's policy is acceptable and we try to deal with its problems (like fuzzy or unclear statements), and focus on our goal to apply the proposed profile to a real use-case.

For the purpose of this paper, we will focus on specific parts of Google's privacy policy, among those that are more informative to the user (i.e., discarding general statements like "*we collect information to provide better services to our users*"). The Google privacy policy [29] is described in natural language and consists of 8 pages and more than three thousand words. From all the material we isolated a set of 9 statements of interest for our case study; the statements that were extrapolated are listed in Table 3.

TABLE 3. STATEMENTS FROM THE GOOGLE'S PRIVACY POLICY.

ID	Statement
ST1	*Many of our services require you to sign up for a Google Account. When you do, we ask for personal information like your name, email address, telephone number, or credit card.*
ST2	*We may collect device-specific information, unique device identifiers and mobile network information.*
ST3	*When you use a location-enabled Google service, we may collect and process information about your actual location.*
ST4	*If other users already have your email or other information that identifies you, we may show them your publicly visible Google Profile information, such as your name and photo.*
ST5	*We will share personal information with companies, organizations or individuals outside Google when we have your consent to do so.*
ST6	*If your Google Account is managed for you by a domain administrator, then your domain administrator […] will have access to your Google Account information (including your emails and other data).*
ST7	*We provide information to our affiliates […] to process it for us, based on our instructions and in compliance with our Privacy Policy.*
ST8	*We may share aggregated, non-personally identifiable information publicly and with our partners […]. For example, […] to show trends about the general use of our services.*
ST9	*We restrict access to personal information to Google employees, contractor and agents who need to know that information in order to process it for us.*

In the following, we describe how the profile described in this paper can be used to model such statements, and keep track of them and their requirements during the development of the application. It should be noted that, although we have complete access to the Google privacy policy [29], we have no information on the means that are applied by the company to actually enforce its privacy policy. Consequently, while most of the elements in the following models are derived from the real world, *Enforcement* elements used in the example are fictional, and serve for the purpose of describing the application of the profile. Furthermore, we interpreted fuzzy statements like those containing "*we may*" sentences in their worst-case meaning, e.g., if a statement says "*we may collect data*" we interpret it as "*we do collect data*".

Due to space restrictions we present the modeling of the statements ST5, ST6 and ST9 (see Table 3). These statements were selected because they are related to the disclosure of personal information to third-parties that can be related or unrelated to the company which has the private information (in this case, Google). The transfer of private information requires the data subject's knowledge and consent and respecting their individual consent is still a recurring problem.

For the sake of organization, we split the diagrams in two figures: one for illustrating ST5 and ST6, and another to illustrate ST9. Both are shown in the next subsections.

A. Statements ST5 and ST6

The model for statements ST5 and ST6 is detailed in Figure 2.

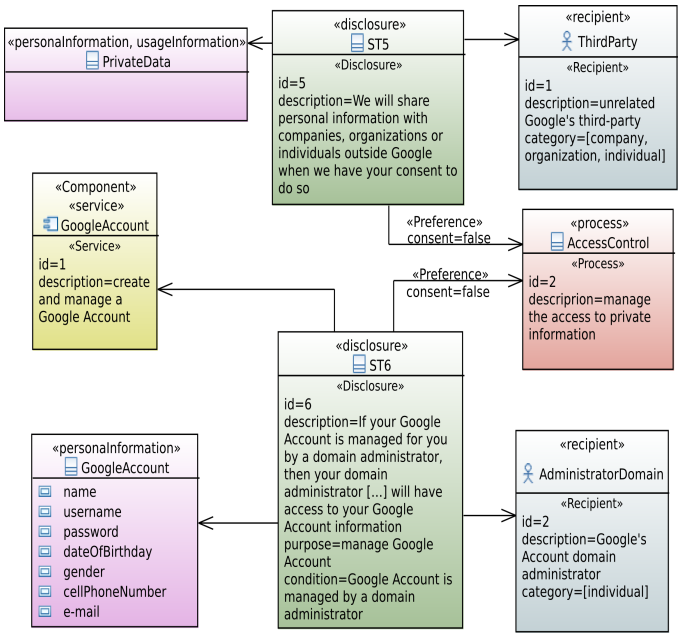

Fig. 2. Representation of statements ST5 and ST6 using the privacy profile.

In Figure 2, both the statements are represented by <<Disclosure>> elements: ST5 describes a generic disclosure of data from Google to other third parties, while ST6 describes the possibility that some specific data of the Google Account will be available to the domain administrator (see Table 3). In this case, the *purpose* for which data to be disclosed is the management of the Google account.

The two different types of recipients to which the statements refer are represented by <<Recipient>> elements. The *ThirdParty* recipient is connected with ST5, and represents a generic third-party unrelated to Google; the categories assigned to this recipient are derived from the statement itself: company, organization, individual. The *AdministratorDomain* recipient represents the domain administrator, and it is categorized as an individual. The *AdministratorDomain* is associated with ST6.

In case of *opt-out* preference by the data subject, both the statements are enforced through access control. This is represented by the *AccessControl* <<Process>> element, since realizing access control requires the implementation of software and organizational measures. Both ST5 and ST6 are then connected with the *AccessControl* element through a relation with the <<Preference>> stereotype. The *consent* attribute set to *false* reflects that access control is applied in case when the user disagree with the statement.

The acceptance of statement *ST6* is mandatory for using the Google Account service; this is represented by the association between ST6 and the GoogleAccount <<Service>> element.

B. Statement ST9

Figure 3 details the model for statement ST9 of Table 3. The statement describes that the access to personal information is restricted to Google employees who need to know that information for processing it; in case they fail to respect privacy obligations, disciplinary measures may be undertaken.

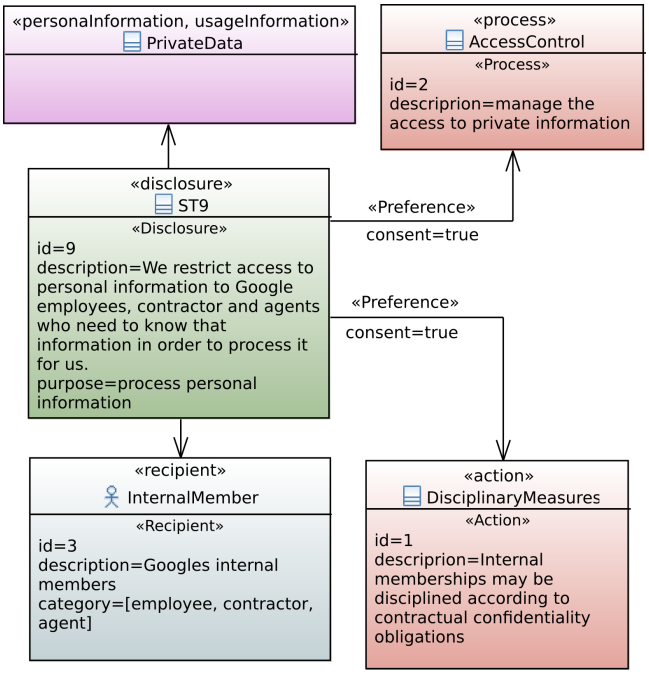

Fig. 3. Representation of statement ST9 using the privacy profile.

Statement ST9 is a <<Disclosure>> statement, since it describes the disclosure of information to internal Google members. The purpose for which data is disclosed is for data processing. The statement is associated to the *PrivateData* elements, which represents the generic data that will be disclosed. Since the statement does not detail which kind of data will be disclosed, we assume that both personal information and usage information are involved; for this reason both the <<PersonalInformation>> and <<UsageInformation>> stereotypes are applied to the same *PrivateData* element.

The *InternalMember* element is a <<Recipient>>, and it represents the recipient of data disclosure. The categories associated with this recipient are *employee*, *contractor*, and *agent*. Among the enforcement solutions, we find the *AccessControl* element and the *DisciplinaryMeasures* element. The *AccessControl* is a <<Process>> element, and it has been already described in the previous section; however, in this case it is applied to enforce an *opt-in* preference of the user. This is modeled by the <<Preference>> relation with the *consent* attribute having the *true* value.

The other enforcement solution, *DisciplinaryMeasures*, is modeled with an <<Action>> element and describes the application of disciplinary measures, e.g., revocation of contracts in case of privacy obligations are not met by the recipient. Also in this case, the enforcement solution is applied in case of an *opt-in* preference of the user.

It is important to mention that a complete model should also include a <<PrivacyPolicy>> element, having a containment relation with all the statement elements included in the model. For this case study, <<PrivacyPolicy>> contains the three statements shown in the figures 2 and 3 (ST5, ST6, ST9), as well as the other six mentioned in Table 3 (ST1-4, ST7-8). To simplify the presentation of the above diagrams, the <<PrivacyPolicy>> element was not shown. A complete diagram representing all the statements aggregated to the <<PrivacyPolicy>> is presented in Figure 4.

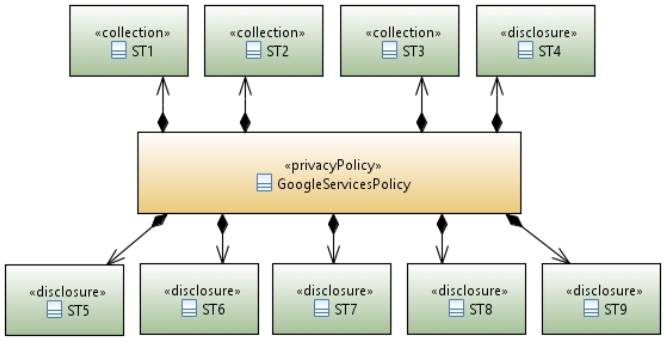

Fig. 4. Privacy Policy element and respective Statements.

V. Conclusions and Future Works

The work in this paper presented a UML profile for privacy-aware applications. The profile can be used to describe the privacy policy that is applied by an application, and keep track of which elements are in charge of enforcing it, e.g., for tracking of privacy requirements or for documentation purposes. As privacy protection of personal information is a big challenge nowadays, modeling solutions addressing privacy elements – especially the enforcing of privacy policies – are a relevant contribution. We found that the visual support of UML diagrams, specialized with stereotypes to indicate the roles of components, was really useful for documentation purposes.

Using the proposed UML profile, we created a set of class diagrams describing a part of the Google privacy policy. Through such diagrams it is possible to identify the interaction between the elements and model the convenient solutions which can be adopted for protecting privacy according to the respective privacy policy. Overall, the outcomes of the case study were promising in respect of the capacity to model realistic privacy aspects.

While we have not yet evaluated it in a real development scenario, we believe that the proposed profile could help the development of web applications and services including privacy protection. Privacy-related components are immediately identified within the software architecture, and their role highlighted with domain-specific stereotypes. Thus, the profile described in this paper is a step forward in integrating privacy-related information in the development process of a web application. We intend to investigate if the solution can also be a step forward for tests activities related to privacy. The idea is to use UML models annotated with our profile as a support to evaluate if applications and services enforce the privacy policies correctly and protect user's privacy. Also, it can be useful to check if the terms of a privacy law within a country were being met.

As future work we intend to perform an extensive study applying the proposed profile in real development scenarios, in order to better evaluate its applicability and possible necessity of extensions. Besides this, extensions of this work are planned in two main directions. The first direction aims at integrating the UML profile proposed in this paper with a reference architecture for the privacy domain [24]. The second direction will investigate the possibility to combine the profile with established techniques for privacy enforcement, e.g., access control, thus integrating the profile into a more comprehensive framework for addressing privacy in the development process.

Acknowledgment

The study reported herein was partially supported by the DEVASSES (*DEsign, Verification and VAlidation of large-scale, dynamic Service SystEmS*) project, funded by European Union's Seventh Framework Programme under grant agreement PIRSES-GA-2013-612569.

Also, we thank the support of São Paulo Research Foundation (FAPESP), through the grant #2013/17823-0.

REFERENCES

[1] E. Bertino, D. Lin, and W. Jiang, "A Survey of Quantification of Privacy Preserving Data Mining Algorithms", in Privacy-Preserving Data Mining, vol. 34, C. C. Aggarwal, P. S. Yu, and A. K. Elmagarmid, Orgs. Springer US, pp. 183–205 (2008).

[2] Hoepman, J-H. "Privacy Design Strategies". In Proceedings of 29th IFIP TC 11 International Conference, SEC 2014, Marrakech, Morocco, June 2014, pp 446-459.

[3] Object Management Group. OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1. OMG Document {formal/2011-08-06}. Aug. 2011.

[4] UML Profile Diagrams. [Online]. Available: http://www.uml-diagrams.org/profile-diagrams.html [Accessed: 16-sep-2014].

[5] Object Management Group. UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms (OMG QoS&FT), Version 1.1. OMG Document {formal/2008-04-05}. Apr. 2008

[6] Object Management Group. A UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded systems, Version 1.1. OMG Document {formal/2011-06-02}. June 2011.

[7] Object Management Group. OMG Systems Modeling Language (OMG SysML), Version 1.3. OMG Document {formal/2012-06-01}. June 2012.

[8] M. Casassa Mont, S. Pearson, S. Creese, M. Goldsmith, N. Papanikolaou, "A Conceptual Model for Privacy Policies with Consent and Revocation Requirements", Privacy and Identity Management for Life. IFIP Advances in Information and Communication Technology Volume 352, 2011, pp 258-270.

[9] OASIS, eXtensible Access Control Markup Language (XACML), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

[10] Q. Ni, A. Trombetta, E. Bertino, J. Lobo . "Privacy-aware role based access control". Proceedings of the 12th ACM Symposium on Access Control Models and Technologies (Sophia Antipolis, France, June 20-22, 2007). ACM, New York, pp. 41-50, 2007.

[11] N. Damianou, N. Dulay, E. Lupu, M. Sloman, "The Ponder Policy Specification Language", Policies for Distributed Systems and Networks Lecture Notes in Computer Science Volume 1995, 2001, pp 18-38.

[12] L. Cranor, B. Dobbs, S. Egelman, G. Hogben, J. Humphrey, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. M. Reagle, M. Schunter, D. A. Stampley, and R. Wenning (2006). "The Platform for Privacy Preferences 1.1 (P3P1.1) Specification". World Wide Web Consortium NOTEP3P11-20061113.

[13] P. Ashley, S. Hada, G. Karjoth, C. Powers, M. Schunter. "Enterprise Privacy Authorization Language (EPAL 1.2)", 2003. [Online]. Available: http://www.zurich.ibm.com/security/enterprise-privacy/epal/Specification/index.html. [Accessed: 23-mar-2015].

[14] P. Kumaraguru, J. Lobo, L. F. Cranor , S. B. Calo, "A survey of privacy policy languages", Workshop on Usable IT Security Management (USM 07): Proceedings of the 3rd Symposium on Usable Privacy and Security, ACM, 2007.

[15] J. Jurjens, "UMLsec: Extending UML for Secure Systems Development", Proceedings of the 5th International Conference on The Unified Modeling Language, 2002.

[16] Ç. Cirit, F. Buzluca, "A UML profile for role-based access control", Proceedings of the 2nd International conference on Security of information and networks (SIN'09), 2009.

[17] D.S. Allison, H.F. El Yamany, M.A. M. Capretz. "Metamodel for privacy policies within SOA," Software Engineering for Secure Systems, 2009. SESS '09. ICSE Workshop on , vol., no., pp.40,46, 19-19 May 2009.

[18] Organization for Economic Co-operation and Development. "OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data", 2013. [Online] Available: http://www.oecd.org/sti/ieconomy/2013-oecd-privacy-guidelines.pdf [Accessed: 26-aug-2014].

[19] A. Cavoukian. "Creation of a Global Privacy Standard", 2006. [Online]. Available: https://www.ipc.on.ca/images/resources/gps.pdf. [Accessed: 23-mar-2015].

[20] A. Cavoukian, T. Hamilton. "The Privacy Payoff: How Successful Businesses Build Customer Trust", McGraw-Hill Ryerson Limited, Whitby, Ontario, Canada, 2002.

[21] ISO/IEC 29100. International Standard - Information technology - Security Techniques - Privacy framework. First Edition (2011).

[22] ISO/IEC 29101. International Standard - Information technology - Security Techniques - Privacy architecture framework. First Edition (2013).

[23] OASIS. "Privacy Management Reference Model and Methodology (PMRM) Version 1.0",2012. [Online]. Available: http://docs.oasis-open.org/pmrm/PMRM/v1.0/csd01/PMRM-v1.0-csd01.pdf. [Accessed:23-mar-2015].

[24] T. Basso, R. Moraes, M. Jino, M. Vieira. "Requirements, Design and Evaluation of a Privacy Reference Architecture for Web Applications and Services". Proceedings of the 30th ACM/SIGAPP Symposium on Applied Computing, Salamanca, Spain, 2015. In press.

[25] M. F. Dennedy, J.Fox, T. R. Finneran. "The Privacy Engineer's Manifesto: Getting from Policy to Code to QA to Value". Apress; 1st Edition, January, 2014.

[26] I. Oliver. "Privacy Engineering: A Dataflow and Ontological Approach". CreateSpace Independent Publishing Platform, 1st Edition, July, 2014.

[27] D. Solove. "A Taxonomy of Privacy," University of Pennsylvania Law Review, Vol. 154, No. 3, pp. 477 – 560, 2006.

[28] A. I. Antón and J. B. Earp. A requirements taxonomy for reducing web site privacy vulnerabilities. Requirements Engineering, 9(3) pp.169–185, 2004.

[29] Google Privacy & Terms. "Privacy Policy", 2014. [Online]. Available: http://www.google.com/policies/privacy/ [Accessed: 26-aug-2014].

[30] BBC News. "Google agrees privacy policy changes with data watchdog", 2015. [Online] Available: http://www.bbc.com/news/technology-31059874. [Accessed: 23-mar-2015].

[31] Information Comissioner's Office. "Google to change privacy policy after ICO investigation", 2015. [Online] Available: https://ico.org.uk/about-the-ico/news-and-events/news-and-blogs/2015/01/google-to-change-privacy-policy-after-ico-investigation/. [Accessed: 23-mar-2015].