A UML Profile for Privacy Enforcement

Javier Luis Cánovas Izquierdo(✉) and Julián Salas

Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), Barcelona, Spain
{jcanovasi,jsalaspi}@uoc.edu

Abstract. Nowadays most software applications have to deal with personal data, specially with the emergence of Web-based applications, where user profile information has become one of their main assets. Due to regulation laws and to protect the privacy of users, customers and companies; most of this information is considered private, and therefore convenient ways to gather, process and store them have to be proposed. A common problem when modeling software systems is the lack of support to specify how to enforce privacy concerns in data models. Current approaches for modeling privacy cover high-level privacy aspects to describe what should be done with the data (e.g., elements to be private) instead of how to do it (e.g., which privacy enhancing technology to use); or propose access control policies, which may cover privacy only partially. In this paper we propose a profile to define and enforce privacy concerns in UML class diagrams. Models annotated with our profile can be used in model-driven methodologies to generate privacy-aware applications.

Keywords: UML · UML-profile · Privacy

1 Introduction

In the last years, specially with the emergence of the Web, personal information has become one of the main assets of software applications. This kind of data usually includes information about users (e.g., email addresses or passport identifiers), personal information (e.g., geolocations, pictures or videos) or even composite information that can be discovered by mining the previous information (e.g., route to go to work or places to pass the night). Most of this information may be considered private, and therefore convenient ways to gather, process and store it have to be proposed to comply with existing regulations and to promote participation by providing accountability and transparency to data subjects.

Model-Driven Engineering (MDE) is a methodology focusing on using models to raise the level of abstraction and automation in software development. MDE relies on models and model transformations for the specification and generation of software applications, thus hiding the complexity of the target technology.

A common problem when modeling software systems is the lack of support to specify how to enforce privacy concerns in data models, that is, the mechanisms

M. Mazzara et al. (Eds.): STAF 2018 Workshops, LNCS 11176, pp. 609–616, 2018.
https://doi.org/10.1007/978-3-030-04771-9_46

(e.g., hashing or ciphering) that have to be applied to meet privacy requirements. Current approaches cover high-level privacy aspects [3,6,10] which address privacy concerns regarding to what elements are private but neglecting how to enforce privacy. The work by Basso et al. [5] proposes a UML profile for privacy-aware applications, however, it is mainly focused on defining privacy and user preferences. Other works (e.g., [1,2,4]) propose methodological approaches to address privacy but they do not focus on enforcement mechanisms. There are also approaches like XACML [12], PRBAC [11], UMLsec [9] or Ponder [7] proposing languages adapted to the definition of access control policies, which can be used to partially manage privacy concerns but they do not target enforcement.

In this paper we propose a profile to model privacy concerns in UML class diagrams with the aim of enabling privacy enforcement. Models are annotated by privacy experts, thus enabling developers (and model-driven tools) to understand how privacy has to be applied to the artifacts involved in model-based methodologies. We believe that our proposal promotes a better documentation of the models and could be easily adapted to existing methodologies to enable the generation of privacy-aware software applications.

The rest of the paper is organized as follows. Section 2 motivates the work and presents a running example. Section 3 describes the profile and Sect. 4 concludes the paper and presents the further work.

2 Motivation

Sharing and processing data has many benefits, but it also has risks to individual privacy: it can reveal information about individuals that would otherwise not be public knowledge. Privacy is a fundamental human right and it is commonly agreed it should be enforced by law. Moreover, developing privacy-aware software systems will also bring the benefits of increasing public engagement by promoting the participation and dissemination, and providing transparency and accountability on the data processing methodologies.

As suggested by the *privacy by design* concept [8], privacy should be protected throughout the whole process of any technological development, from the conception of a technology to its realization. Dealing with privacy at each stage of the data lifecycle (i.e., collection, maintenance, release, and deletion) will be enhanced by specific support when modeling software artifacts, thus enabling developers to easily define how data privacy has to be treated.

Along this paper we will use a running example to illustrate our approach. Let's imagine a public organization willing to publish some data regarding its employees (e.g., for statistical purposes). Figure 1 shows a UML class diagram model to represent companies, employees and positions. A company, which has a name and a tax number, is composed of employees, which have names, ages and passport numbers; and offers a set of positions, with a name and a salary.

Even with this small model, several concerns can be identified when publishing data conforming to this model. For instance, name and passport information uniquely identifies an employee and should be removed, encrypted or replaced;

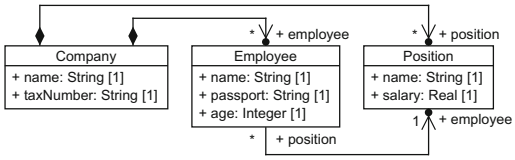

Fig. 1. Running example.

age information can be leveraged to uniquely re-identify an employee and should be treated (e.g., removing outliers to decrease its uniqueness); and salary information is generally considered sensitive information and could be masked by applying generalization (i.e., released using ranges of salaries).

It should be noted that there is no one-fits-all solution for providing privacy. Along with the possible benefits of releasing data there are some risks to individual privacy, this trade-off between the utility vs. privacy should be considered (i.e., performing the minimal number of privacy enforcement modifications to the data to preserve privacy). There are several methods for data protection, each one with its own strengths and weaknesses, and different trade-offs. An extensive analysis should be done to choose a method over others, however, by knowing the characteristics of each of them, a developer may provide certain guarantees of privacy by design to end-users.

In this paper we propose a UML profile to annotate class models with information regarding privacy concerns in order to enable their enforcement.

3 A Profile for Privacy Enforcement

Privacy enforcement covers the set of mechanisms deployed to protect private data [14]. To enforce privacy in UML we defined a profile following the standard recommendations [13]. The profile annotates UML classes and their properties. Class associations require special treatment, as we will show. Next we describe the main elements of the profile[1].

UML Property Privacy Type. UML properties can be classified according to a specific privacy type. This information is required for every property in the class model and classifies its sensitiveness, which is later used by the privacy type applied to the owning class, as we describe below. We identify four privacy types: *non-sensitive*, for non-confidential properties; *sensitive*, for confidential properties; *identifier*, for those properties that can unambiguously identify the owner of the property; and *quasi-identifier*, for properties that uniquely combined can be used to re-identify the owner of the property.

In the profile, the privacy type of a property is specified by the `PrivacyType` stereotype, which extends the `Property` metaclass. The actual values of privacy types are defined in the `PropertyPrivacyType`.

[1] The profile implementation and example are available at http://hdl.handle.net/20.500.12004/1/A/UMLPP/001.

UML Property Anonymization. UML properties can optionally be anonymized following a specific method. The anonymization of a property protects its values and can be used to configure how to store them. These methods are based on reducing the amount or precision of the data and follow two main principles: (1) masking the data and (2) using synthetic values instead of real ones. Masking the data can be divided in two categories: *non-perturbative* and *perturbative*.

Non-perturbative masking reduces the level of details without distorting it. Some well-known non-perturbative masking methods are: (1) *generalization*, which coarses a property by combining several (or a range) of values to a more general one; (2) *top/bottom coding*, which sets values above/below a given threshold into a single category; and (3) *suppression*, which removes outliers values of individual property values in order to decrease the uniqueness of the elements.

Perturbative masking includes (1) *noise addition*, which is applied to numerical properties and consists of adding a noise vector (most commonly) drawn from a $N(0, \alpha\Sigma)$, where Σ is the covariance matrix of the original data values; (2) *data/rank swapping*, which exchanges categorical property values in such a way that marginals are maintained; (3) *post-randomization*, where property values are changed according to a Markov matrix; and (4) *microaggregation*, which partitions the property values into groups containing each at least a specific amount of records and publishing the average record of each group.

In the profile, the anonymization method of a property is specified by the `PrivateMethod` stereotype, which extends the `Property` metaclass. The actual methods are defined in the `AnonymizationMethod`.

UML Class Privacy Type. UML classes can be annotated to indicate the privacy protection mechanism that has to be enforced. Annotating a class with this kind of information protects the way class instances are queried. Thus any instance of a class including this annotation will not provide information regarding its *identifier* properties and will protect *nonsensitive*, *sensitive* and *quasi-identifier* properties. The two main models for privacy protection, from which many others have been developed, are k-anonymity and ϵ-differential privacy (see `KAnonymity` and `DifferentialPrivacy` stereotypes in our profile).

The concept of k-anonymity was defined to release personal data while safeguarding the identities of the individuals to whom the data refer [15]. A dataset is k-anonymous if each record is indistinguishable from at least other $k-1$ records within the dataset, when considering the values of its quasi-identifiers. This model therefore aims to protect from attacks to obtain sensitive property values relying on quasi-identifiers. Applied to a UML class, this mechanism guarantees that individual instances of a UML class are indistinguishable from at least other $k-1$ instances.

To protect from inferences due to the low variability of sensitive properties in a k-group, ℓ-diversity and t-closeness models were proposed. A k-anonymous set of instances is said to be ℓ-diverse if, for each group of instances sharing quasi-identifier values, there are at least ℓ well-represented values for the sensitive property. A k-anonymous set of instances is said to have t-closeness if, for

each group of instances sharing quasi-identifier values, the distance between the distribution of each sensitive property within the group and the distribution of the property in the whole set is no more than a threshold t.

The ϵ-differential privacy applied to UML classes establishes that the removal or addition of a single element to the set of class instances does not (considerably) change the results on an analysis. Therefore, the presence or absence of any individual element is not revealed by the computation (up to $exp(\epsilon)$) (Fig. 2).

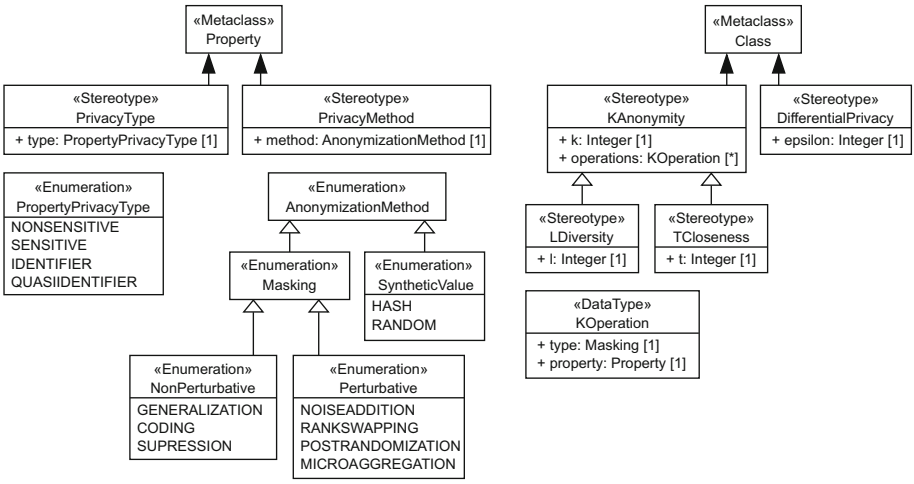

Fig. 2. Proposed UML profile to model privacy enforcement.

Privacy for UML Associations. In our approach, class associations obtain the privacy enforcement declared for the association endpoint. Although this solution could cover the privacy enforcement at UML class model level, it may become a challenging task when these models are transformed to low-level ones used to generate a software system. For instance, UML class models annotated with our profile can be used to generate a database schema, where resolving associations could involve the composition of different database tables. This composition is not trivial, specially if source/target tables corresponds to UML classes annotated with different privacy types. While it would be feasible to compose information of tables coming from UML classes annotated with ϵ-differential privacy, such composition would be challenging for k-anonymity (composability has been mentioned as open research question for Big Data privacy [16,17]).

Example. Figure 3 shows the running example described before annotated with our profile. As can be seen, the `name` properties of the `Company` and `Position` classes have been annotated as `NONSENSITIVE` as they not involve any privacy risk. The `taxNumber` property of the `Company` class, and the `name` and `passport` properties of the `Employee` class have been annotated as `IDENTIFIER`, as they can

be used to uniquely identify the company and the employee, respectively (i.e., they will be removed in any query to the instances of such classes). The `age` and `salary` properties of the `Employee` and `Position` classes have been annotated as `QUASIIDENTIFIER` and `SENSITIVE`, as they store data that has to be protected. Additionally, for illustration purposes we use different anonymization methods for these properties, for instance, employees' names and passport information are protected using `HASH` and `RANDOM` mechanisms, respectively.

In the example we also indicate privacy protection mechanisms for `Employee` and `Position` classes, which apply k-anonymity. The k-anonymity for `Employee` class indicates a k value of 4 and applies the `SUPRESSION` method when retrieving the `age` property, thus decreasing the uniqueness of the class instances. On the other hand, the k-anonymity for `Position` class also indicates a k value of 4 and applies the `GENERALIZATION` method when retrieving the `Salary` property, thus the values of such property are expressed as ranges of values.

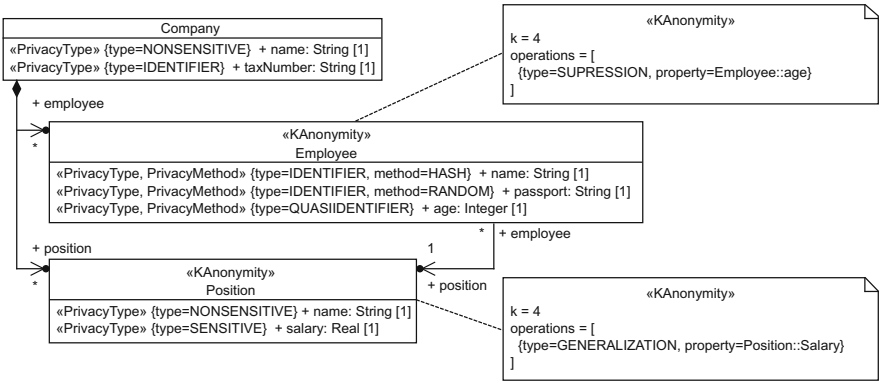

Fig. 3. Privacy enforcement profile applied to the running example.

This UML Class model annotated with our profile provides detailed information to enforce privacy when dealing with its instances. This information can later be used in model-driven methodologies to generate the needed artifacts in a privacy-aware software application. For instance, profile information can be used to customize the generation and configuration of the database schema, and to tune the behavior of queries in the data.

4 Conclusion and Further Work

In this paper we have presented a UML profile to model and enforce privacy concerns in UML class diagrams. We believe our approach paves the way to use models annotated with privacy enforcement information in model-based approaches to enable the validation and generation of privacy-aware applications.

As further work, we are interested in applying our approach to specific fields, such as Big Data and Web Engineering, where it is common to deal with sensitive information. We also plan to explore how privacy information could promote the Open Data movement, currently mainly lead by public organizations. We believe that offering better mechanisms to enforce privacy in Open Data datasets could encourage more organizations (even private companies) to join the movement.

References

1. Ahmadian, A.S., Peldszus, S., Ramadan, Q., Jürjens, J.: Model-based privacy and security analysis with carisma. In: Foundations of Software Engineering, pp. 989–993 (2017)
2. Ahmadian, A.S., Strüber, D., Riediger, V., Jürjens, J.: Model-based privacy analysis in industrial ecosystems. In: European Conference on Modelling Foundations and Applications, pp. 215–231 (2017)
3. Allison, D.S., Yamany, H.F.E., Capretz, M.A.M.: Metamodel for privacy policies within SOA. In: Workshop on Software Engineering for Secure Systems, pp. 40–46 (2009)
4. Alshammari, M., Simpson, A.: A UML profile for privacy-aware data lifecycle models. In: International Workshop on Computer Security, pp. 189–209 (2017)
5. Basso, T., Montecchi, L., Moraes, R., Jino, M., Bondavalli, A.: Towards a UML profile for privacy-aware applications. In: International Conference on Computer and Information Technology, pp. 371–378 (2015)
6. Busch, M.: Evaluating & engineering: an approach for the development of secure web applications (2016)
7. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The ponder policy specification language. In: International Workshop on Policies for Distributed Systems and Networks, pp. 18–38 (2001)
8. Hoepman, J.: Privacy design strategies - (extended abstract). In: International Conference on Systems Security and Privacy Protection, pp. 446–459 (2014)
9. Jürjens, J.: UMLsec: extending UML for secure systems development. In: 5th International Conference on the Unified Modeling Language, pp. 412–425 (2002)
10. Mont, M.C., Pearson, S., Creese, S., Goldsmith, M., Papanikolaou, N.: A conceptual model for privacy policies with consent and revocation requirements. In: International Summer School on Privacy and Identity Management for Life, pp. 258–270 (2010)
11. Ni, Q., et al.: Privacy-aware role-based access control. ACM Trans. Inf. Syst. Secur. **13(3), 24:1–24**, 31 (2010)
12. OASIS: Extensible Access Control Markup Language (XACML). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml. Accessed April 2018
13. OMG: Unified Modeling Language. https://www.omg.org/spec/UML/2.5/. Accessed April 2018
14. Salas, J., Domingo-Ferrer, J.: Some basics on privacy techniques, anonymization and their big data challenges. Mathematics in Computer Science (2018, in press)
15. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report (1998)

16. Soria-Comas, J., Domingo-Ferrer, J.: Big data privacy: challenges to privacy principles and models. Data Sci. Eng. **1**(1), 21–28 (2016)
17. Torra, V., Navarro-Arribas, G.: Big data privacy and anonymization. In: Lehmann, A., Whitehouse, D., Fischer-Hübner, S., Fritsch, L., Raab, C. (eds.) Privacy and Identity 2016. IAICT, vol. 498, pp. 15–26. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55783-0_2