Privacy-Enhanced System Design Modeling Based on Privacy Features

Amir Shayan Ahmadian
University of Koblenz Landau, Germany
ahmadian@uni-koblenz.de

Daniel Strüber
University of Koblenz Landau, Germany
strueber@uni-koblenz.de

Jan Jürjens
University of Koblenz Landau, Germany
Fraunhofer ISST, Germany
http://jan.jurjens.de

ABSTRACT

To ensure that their stakeholders' privacy concerns are addressed systematically from the early development phases, organizations can perform a privacy enhancement of the system design, in which appropriate technical and organizational controls are established. Such a privacy enhancement needs to account for three crucial types of input: First, risks to the rights of natural persons, as determined in a dedicated privacy impact assessment. Second, potential interrelations and dependencies among the privacy controls. Third, potential trade-offs regarding the costs of the controls. Despite numerous existing privacy enhancing technologies and catalogs of privacy controls, there has been no systematic methodology to support privacy enhancement based on these types of input.

In this paper, we propose a methodology to support the coherent privacy enhancement of a system design model. We consider an extensive variety of privacy controls, including privacy-design strategies, patterns, and privacy enhancing technologies. Representing these controls as privacy features, we explicitly maintain their interrelations and dependencies in a feature model. In order to identify an adequate selection of controls, we leverage a model-based cost estimation approach that analyzes the associated costs and benefits. We further demonstrate how the selected features can be integrated into the system model, by applying reusable aspect models to encapsulate the required changes to the system design. We evaluated our methodology based on three practical case studies.

CCS CONCEPTS

• **Security and privacy** → *Software security engineering*; • **Software and its engineering** → *Software design engineering*;

ACM Reference Format:
Amir Shayan Ahmadian, Daniel Strüber, and Jan Jürjens. 2019. Privacy-Enhanced System Design Modeling Based on Privacy Features. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3297280.3297431

ACM ISBN 978-1-4503-5933-7/19/04…$15.00
https://doi.org/10.1145/3297280.3297431

1 INTRODUCTION

Article 25 of the *General Data Protection Regulation* (*GDPR*) [30] prescribes *Privacy by Design* (PbD), requiring service providers to implement appropriate technical and organizational controls from the early development phases for ensuring that the privacy concerns of their service customers and the privacy principles related to the processing personal data are addressed by design.

A privacy enhancement begins with a *Privacy Impact Assessment* (*PIA*), which determines privacy threats by performing a systematic risk assessment, and suggests potential technical and organizational controls to mitigate the privacy risks arising from those threats. The principle of privacy by design mandates that the system design needs to be revised (enhanced) to incorporate the suggested controls, thus mitigating the risks. For instance, to mitigate the risk of processing a piece of sensitive data for an unauthorized purpose, a control such as *data minimization* has to be integrated into the system. However, such privacy controls are too abstract to be integrated directly into the system design; instead, they may be established using one or multiple Privacy-Enhancing Technologies (PETs), such as *Role-Based Access Control* [11].

Integrating an appropriate set of PETs into the system design is an intricate task that involves a number of sensitive aspects: (I) Some privacy risks are more pressing than others. Data owners have varying concern about particular kinds of risks. For example, the leakage of email addresses may not be as problematic as that of national identification number or biometric data. According to the GDPR, the latter belong to special categories of personal data. (II) PETs can be related via various dependencies or conflicts. For example, the *authorization* to perform a particular task on data requires an *authentication.* (III) The implementation of PETs may come with various costs; implementing certain desirable PETs can be prohibitively expensive. Despite earlier work on security and privacy enhancement (discussed in Sect 6), there is no methodology for improving an existing system design while simultaneously addressing these aspects.

In this paper, we thus propose a systematic model-based methodology to coherently support the privacy enhancement of IT systems, addressing risks, interrelations, and costs in the abovementioned sense. We use a set of privacy features that realize the privacy controls to conduct the enhancement. Our methodology is based on system models expressed in UML (Unified Modeling Language) [22], the standard modeling language in numerous software domains [29]. Using models, the complexity of systems are handled through abstraction. Moreover, the models enable us to perform the privacy enhancement during the early stages of the system design.

As further input, our methodology takes the risks and controls identified while performing a PIA. We use the PIA methodology introduced in [3]. The result of the enhancement can be evaluated iteratively by experts by performing the PIA on enhanced system models. Specifically, we make the following contributions:

(I) We map the *NIST* privacy controls [21] to a set of privacy features, including privacy design strategies [13], patterns [10, 24, 27], and privacy enhancing technologies [6, 9, 31]. Furthermore, we identify conflicts and dependencies among these features, and specify their interrelations using a feature model [14] (Sect. 4.1).

(II) To perform a cost-benefit analysis in our model-based privacy enhancement, we extend the cost estimation approach provided in [5] to make it applicable to reusable dataflow models (Sect. 4.2).

(III) To enable the integration of the features in the system design, first, we introduce a UML profile to establish traceability between privacy controls and model elements, and second, we propose to express the privacy enhancement by using and extending the concept of *Reusable Aspect Models* (*RAMs*) [16]. We extend RAMs with activity diagrams to specify *data flow* views, which are particularly important in our privacy setting (Sect. 4.3).

The remainder of this paper is organized as follows. In Sect. 2, the necessary background is provided. In Sect. 3, we introduce an example and the research questions. In Sect. 4, we describe our methodology. In Sect. 5, we present our case studies. In Sect. 6, we discuss related work. Finally, in Section 7, we conclude.

2 BACKGROUND

2.1 Model-Based Privacy Impact Assessment

In [3], a privacy impact assessment (PIA) methodology to assess the impact of privacy violations, based on a list of privacy targets [23] that are derived from privacy principles, is introduced. This methodology is supported by a model-based privacy analysis [1, 2] to identify the privacy violations. The privacy targets that are at risk are identified by assessing the impact of each violation. The assessment takes into account the potential embarrassment to the owner (*DC*) of the sensitive data, and the reputation damage to the organization (*DP*). The former is determined by obtaining feedback from the owner. The latter is specified by the organization. The feedback of the owner of the sensitive data, and the organization are expressed with different categories, namely *negligible, limited, significant* and *maximum*. Each of these categories is assigned to a value between 1 and 4, called *impact value* (*IV*). Moreover, the assessment accounts for three different categories of sensitive data, being assigned to values between 0 and 1 as well to obtain a *personal data category value* (*PDCV*). The final *impact score* ($IA = PDCV \times DC\text{-}IV \times DP\text{-}IV$) is expressed in an ordinal scale with the values *low*, *medium*, *high*, and *very high*. Finally, a list of controls for mitigating the risks are suggested. The privacy enhancement methodology proposed in this paper leverages this PIA methodology to identify the privacy risks.

2.2 Privacy Design Strategies, Patterns and Privacy-Enhancing Technologies

A strategy describes a fundamental approach to achieve a certain goal. Hoepman [13] introduces eight privacy design strategies, which are derived from existing privacy principles and data protection laws, thus bridging the gap between the legal and the technical domain. To make the definition of these strategies more concrete, Colesky et al. [8] refine these eight strategies by defining a set of sub-strategies for each strategy, and mapping each sub-strategy to a set of privacy patterns [24]. Originally, a pattern is more concrete than a strategy and describe a common recurring structure to solve a general design problem. The term *privacy enhancing technology* (PET) was originally introduced for a category of technologies with embedded privacy features that minimize the processing of personal data, and decrease the privacy risks for the user's data [12]. PETs realize and implement privacy design patterns.

Privacy design strategies, patterns, and privacy-enhancing technologies may be affected by certain relationships and dependencies. For example, the *hide* strategy may not be applied together with the *inform* strategy. Such relationship represent necessary configuration knowledge for ensuring a valid use of the selected strategies. However, these relationships were not considered in the original systematization of privacy design strategies. In the present work, we analyze interactions between the considered strategies, and formally specify the identified relationships using a feature model.

2.3 Function Point Analysis (FPA)

Most cost estimation models require to measure the functional size of a software to be developed [5]. The aim is to quantify the amount of functionality released to a user concerning the data that the software has to use to provide the functions, and the transactions through which the functionality is delivered. *Function Point Analysis* (*FPA*) [4] is one of the most commonly used functional size measurement methods. FPA identifies and weights data and transactional function types. Data functions represent data, and transactional functions represent operations that are relevant to the user. Data functions are classified into *internal logical files* (*ILF*)—the data that is maintained within the boundary of an application—and *external interface files* (*EIF*)—the data that is maintained outside the boundary of the application being measured. Transactional functions are classified into *external inputs* (*EI*), *external outputs* (*EO*), *external inquiries* (*EQ*). An EI processes an ILF. An EO presents data to a user. An EQ retrieves data from ILFs and EIFs. For every data or transactional function different weights are defined.

In [5, 17], to estimate the cost of modeling, the authors apply FPA to UML models by defining a precise mapping between UML elements, and FPA's data and transactional functions. They focus on use-case, class, and sequence diagrams. In this paper, we propose a mapping between activity diagram elements, and FPA's data and transactional functions.

2.4 Reusable Aspect Models (RAMs)

*Reusable Aspect Model*s (*RAM*s) [16] is an aspect-oriented multi-view modeling approach for software design modeling. The paradigm of aspect orientation generally aims to identify, separate, and represent crosscutting concerns. In RAM, the reusable concerns are modeled using UML *class* (structure view), *sequence* (message view), and *state* (state view) diagrams. A RAM may be (re)used within other models via its *usage*, and *customization* interfaces. The former specifies the design structure and the behavior of the reusable

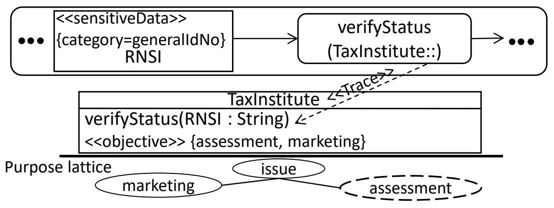


Figure 1: Design model excerpt together with an excerpt of a purpose lattice.

model. The latter specifies how to adapt the reusable model using *parameterized* model elements (marked with a vertical bar, |). A RAM model can be (re)used by composing the parameterized model elements with the elements of other models and RAMs. A RAM *weaver* is used to create a composed design model.

In [20], RAMs are used to model security patterns. We benefit from this work to perform the enhancement of a system model. However, in a privacy context, specifying and analyzing data flows in a system is crucial, which cannot be captured by the classical RAM diagram types. In this paper, we propose an extension of RAM based on activity diagrams to express the behavior of a system.

3 RUNNING EXAMPLE

In this section, we provide an example that will be used through the paper to explain our methodology. This example is taken from a practical scenario introduced in the VisiOn EU project [1]. Figure 1 shows a design model excerpt including a class and an excerpt of an activity diagram. The complete activity diagram specifies the process of issuing a birth certificate for a client in the administration system of the Municipality of the Athens (*MoA*). The lower side of the figure shows an excerpt of a *purpose lattice*. The full lattice determines a set of available purposes and their relationships; the part in dashed lines specifies the authorized purpose for which a particular object might be processed. To issue a birth certificate the *Registry Number of Social Insurance* (*RNSI*) is required. According to Figure 1, the RNSI has to be sent to a *Tax Institute* for verifying the tax status of the customer. To this end, a *CallOperationAction* named *verifyStatus* induces a call to the *verifyStatus* operation in the *taxInstitute* class. For this operation two processing purposes are defined using the stereotype ≪objective≫, namely *assessment*, and *marketing*. According to the purpose lattice (specifying the authorized purposes), the RNSI should not be processed for the purpose of *marketing*. Processing of a piece data object for unauthorized purposes is a privacy violation. This analysis is based on the model-based privacy analysis introduced in [1, 2].

Performing the privacy impact assessment (PIA) methodology (Sect. 2.1) yields two privacy targets at risk: **P1.4** *Ensuring limited processing for specified purposes*, and **P5.2** *facilitating the objection to direct marketing activities*. The category of the RNSI is *general ID number* ($PDCV = 1$). Assuming that both the data owner and the organization rate the impact as *maximum* ($IV = 4$), the final impact score for both targets at risk are $16 = 1 \times 4 \times 4$, according to the ordinal scale provided in [3] a *very high* score. To mitigate the risks the following NIST privacy controls [21] are suggested: For **P1.4**: ***AP-2*** *Purpose Specification*, and **DM-1** *Minimization of Personally Identifiable Information*. For **P5.2**: **DM-1**, and **TR-1** *Privacy Notice*.

[1]https://cordis.europa.eu/project/rcn/194888_en.html

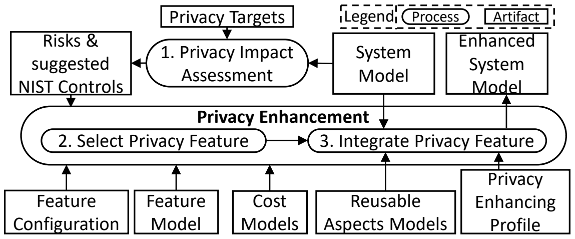


Figure 2: The overview workflow of the methodology.

The produced list of controls must be evaluated for applicability, a challenging task that involves two crucial questions: **Question 1:** *How can an adequate selection of controls to mitigate the identified privacy risks be identified?* We need to answer this question by taking into account the severity of the identified violations (as captured by the impact score), possible interrelation and dependencies between controls, and the costs for deploying the controls to the system. **Question 2:** *How can the selected controls be incorporated into the system model?* Following the privacy-by-design principle, we need to ensure that the system at hand is designed with the selected privacy controls in mind. To this end, the challenge is to enrich and expand the design model to account for the controls.

4 PRIVACY-ENHANCED SYSTEM MODELING

Figure 2 provides an overview of our methodology to support the privacy enhancement considering risks, interrelations between the controls, and trade-offs regarding the costs of the controls. A Privacy Impact Assessment (PIA) is performed upfront to identify privacy risks and to suggest a list of NIST privacy controls [21] to mitigate those risks. Our privacy enhancement of a system model is performed based on the suggested controls, a feature model of privacy design strategies, a cost model, and a set of reusable aspect models (RAMs). The main aim of the enhancement is to select proper privacy features—privacy design strategies and their refinement into patterns and PETs—and integrate them in a system model to mitigate the risks. In this section, we first present the feature model. Secondly, we introduce a model-based cost-estimation approach. Finally, we describe how the enhancement of a system model is performed using a UML profile and RAMs.

4.1 Privacy Design Strategies Feature Model

The purpose of the controls is to minimize, mitigate, or eliminate the identified privacy threats. Controls can be technical or non-technical; technical controls lend themselves to incorporation into the system. Nevertheless, the NIST technical controls are too generic to be directly integrated into a system model. For instance, ***DM-1*** *Minimization of Personally Identifiable Information* is a NIST privacy technical control. When integrating this control into the system model, one can rely on various data-minimization technologies and strategies, for example: exclude data from processing, define specific data processing purposes, or destroy data.

Hence, to apply the controls to system models, we map the NIST privacy controls to a set of *privacy design strategies*, showing an

Table 1: An excerpt of the mapping between privacy design strategies and the NIST privacy controls.

Privacy Control (NIST)	Design Strategy
(AP-2) Purpose Specification	Consent
(DM-1) Minimization of Personally Identifiable Information	Minimize, Hide
(DM-2) Data Retention and Disposal	Minimize, Hide
(IP-1) Consent	Consent
(TR-1) Privacy Notice	Notify

excerpt in Table 1. As introduced in Sect. 2.2, we reuse a selection of eight privacy design strategies from existing work, including their concrete specifications using *sub-strategies*, *privacy design patterns* and *privacy-enhancing technologies* (PETs), thereby simplifying the realization of controls. The privacy design strategies, design patterns, and PETs provide an abstraction layer upon which the enhancement of system models with different levels of abstractions is enabled.

To map the design strategies to privacy design patterns—not included in Table 1—we leverage the correlations of the strategies and patterns provided in [7, 8]. We add a number of design patterns [10, 27] to refine this correlation. Eventually, we map each design pattern to one or more PET(s) [6, 9, 31]. The mapping between the controls and the privacy design strategies and the mapping between the privacy design patterns and the PETs, is achieved using an extensive literature review and argumentation. The complete table with the all mappings is documented in an Excel file[2].

As a contribution of this work, we performed an investigation of interactions between the considered selection of privacy design-strategies, patterns, and PETs. To ensure that we can use them in our automated approach, we formalized the identified interactions using feature modeling. Feature modeling allows to capture variabilities in a system in terms of features and relationships between them. An excerpt of the resulting privacy-design-strategy feature model is presented in Figure 3. A feature model provides a tree-like hierarchy to structure different features. A child feature is either *mandatory* or *optional* for its parent feature. Furthermore, a feature model allows to group a set of feature together with *or*-groups and *alternative*-groups. Where *or*-groups require at least one feature (from that group) be present if its parent feature is present, whereas *alternative*-groups require exactly one feature from that group to be present if its parent is present. Furthermore, a feature model allows us to define *require* and *exclude* relations between different features.

To investigate the interactions between the strategies, patterns, and PETs, we performed an extensive literature review. A few of this interactions are demonstrated in Figure 3 using the *require* and *exclude* relations. For instance, principally the anonymization is in conflict with transparency, therefore the strategies and patterns which use anonymization excludes the strategy related to transparency (for instance, *Inform* strategy) [13]. *hierarchical attribute-based access control* [32] (a sub pattern of *Authorization*) requires encryption to provide the authorization mechanisms, therefore, it requires the strategy *Obfuscate* (not shown in Figure 3).

[2]https://www.dropbox.com/sh/ct08b2fb78t2t9n/AAAafk-t9FIEeFp0XUR9Juava?dl=0

Furthermore, to enable an adequate selection of features, similar to the work presented in [3, 23] the privacy features are classified based on the *levels of rigour*. The levels of rigour are defined as the attributes of the features and express the strength of the features to mitigate privacy risks with different severity levels. The features are categorized to four levels of rigour, namely *sufficient*, *medium*, *strong*, and *very strong*. The levels of rigour have to be assigned to the features before the privacy enhancement.

According to Figure 3, for the privacy enhancement of a system design model, initially any of the 8 privacy design strategies may be selected. For instance, if the strategy *Hide* is selected, optionally one or more sub-strategies might be chosen. If the sub-strategy *Restrict* is selected, one or more design pattern(s) may be selected. The design pattern *Authorization* may be realized by only one of the given technologies (*U-Prove*, *Idemix*, or *RBAC* (Role-Based Access Control)). According to the feature model, the technology *U-Prove* is further realized by the technology *Blind Signature Protocol*.

We created the feature model using the *FeatureIDE*[3] framework [15]. This framework also supports the configuration of features (that is, their assignment to *active* or *inactive*) based on a dedicated editor; configurations can be saved as configuration files. At the beginning, the configuration of features is an empty configuration of the feature model. In our work, we use a configuration to identify which features already exist or are modeled in a system model. The already existing features in a feature model are specified as active. For space reasons, we only show a representative excerpt of the feature model, the full feature model can be found online[4].

4.2 Model-Based Cost Estimation

To estimate the costs of privacy design strategies, patterns, and PETs, we propose a model-based cost estimation approach. The approach assumes that the design strategies, patterns, and PETs are modeled using activity diagrams, one of the main diagram types for specifying behavioral modeling in UML. *Functional point analysis* (*FPA*) has been used in [5, 17] to estimate costs in system models (use case, class, and sequence diagrams); we now describe how we customized FPA for application to activity diagrams.

In an activity diagram, a piece of data is specified as an *ObjectNode*. An ObjectNode is fed into an action of an activity as a parameter. Concerning FPA, we identify an ObjectNode as a *data function*. We further need to classify an ObjectNode (*internal logical file* (*ILF*) or *external interface file EIF*). In [2], the stereotype ≪recipient≫ is introduced to annotate the actions (of an activity) that belongs to a process outside the boundary of the process being analyzed. An ObjectNode that is fed into an action that is annotated with ≪recipient≫ is an EIF. All other ObjectNodes are ILFs.

An action in an activity diagram is a *transactional function*. An action which processes an ILF is an *external input* (*EI*). An action which processes an EIF is an *external output* (*EO*). An action which retrieves an object from a *DataStoreNode*—models a database in an activity diagram—with the UML selection behavior (specified within a note symbol with the keyword ≪selection≫), is an *EQ*.

Having mapped the FPA elements to UML activity's elements, to every function (either data or transaction) a weight must be

[3]https://featureide.github.io/

[4]https://www.dropbox.com/sh/ct08b2fb78t2t9n/AAAafk-t9FIEeFp0XUR9Juava?dl=0

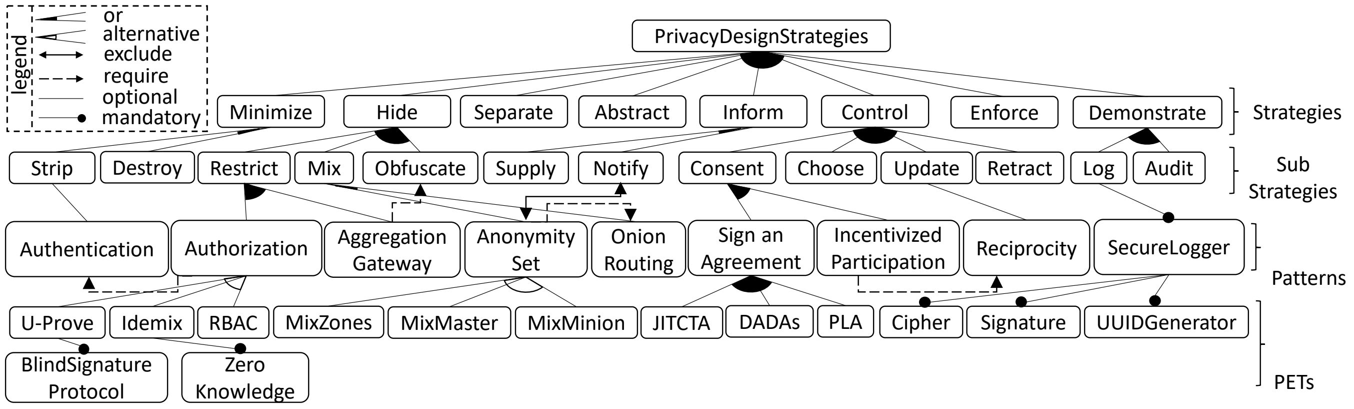


Figure 3: An excerpt of the feature model including privacy design strategies, sub-strategies ,privacy patterns, and PETs.

Table 2: Function types and their weights.

Function Type	ILF	EIF	EI	EO	EQ
Weight	7	5	3	4	3

assigned. The weighting of function types may be obtained on the basis of their complexities. We use the approach presented in [5], to assign complexity scores and weights to the functions. The number of each function type in the model is multiplied by a predefined weight (Table 2) for the function type to calculate the final FP.

For instance, if in an activity diagram a which models a feature, two ILFs, one EIF, three EIs, one EO and one EQ are identified, then the total function point $FP(a)$ for the activity diagram a is: $FP(a) = (2 \times 7) + (1 \times 5) + (3 \times 3) + (1 \times 4) + (1 \times 3) = 35$.

Table 2 only provides a baseline to calculate the FPs in an activity diagram. However, such weights can and should be modified (or classified)—for instance by a deeper analogical analysis based on the experiences of the privacy experts and the feedback of the stakeholders—to obtain more precise weights.

4.3 Model-Based Privacy Enhancement

In our methodology, a design model specifies the structure and behavior of a system using class and activity diagrams. In a privacy enhancement, the selected privacy features are applied to design models, in particular, to their contained class and activity diagrams. The privacy enhancement is performed on two abstraction levels: (I) Establishing traceability between privacy features and affected design model elements via a dedicated profile. (II) Extending the structure and behavior with privacy features by applying *reusable aspect models* (*RAMs*).

In a nutshell, the privacy enhancement starts by identifying proper strategies concerning the suggested controls obtained by performing a PIA upfront, and the mapping (Table 1) between the controls and the strategies. The selection of the proper sub-strategies, patterns and PETs (privacy features) considers the *impact score*s of the privacy targets at risks, and the *levels of rigour* of the features. Furthermore, the interrelations between the features specified by *require* and *exclude* relations in the feature model have to be considered. If a selection between two or more features from the same level of rigour is necessary, a cost analysis concerning the behavior of the features is performed to select a feature.

UML profile for privacy enhancement. To support the system developers in understanding which elements are affected by privacy concerns, we introduce a UML profile named *privacy enhancing profile*. This profile can be used to automatically establish traceability between privacy features and model elements, by annotating the elements. Our profile includes one stereotype called ≪enhance≫, whose details we show in Table 3. In the following explanations, we focus on the enhancement of activity diagrams; class diagrams are handled similarly. A *Behavior* (an action in an activity diagram) may be annotated with ≪enhance≫ and its tags, namely $\{strategy\}$, $\{pattern\}$, and $\{PET\}$ specifying the respective feature to be integrated into the action. The action has to be a *CallBehaviorAction* (indicated by placing a rake-style symbol), which calls a behavior including the behavior of the integrated feature.

The metamodel shows in Figure 4 demonstrates the underlying concept of the enhancement. A privacy control is mapped to a set of (privacy) features which enhance a behavior in a system model. A behavior may have precondition and postcondition constraints. A constraint is an assertion that specifies a restriction that must be satisfied by any valid realization of the behavior containing the constraint [22]. We benefit from these constraints to regulate the control flow constraints specified by require and exclude relations. A feature f may require or exclude other feature(s). To verify if a feature is required or excluded, using the preconditions' constraints, we investigate if the feature is *active* or *inactive* in the enhanced model element. The preconditions indicate the constraints that must be hold before invoking the feature f. In other words, the preconditions specify the features that have to be integrated into the system model (*active—require* relation) before integrating the feature f, and the features that must not exist in the system model (*inactive—exclude* relation) by integrating the feature f. Preconditions are evaluated on the given configuration of the feature model. Furthermore, the postcondition establishes a constraint that holds in the resulting system state, indicating that for instance, a feature (contained in the tag of ≪enhance≫) is integrated into the system model. The postconditions modify the configuration of a feature model and provide a basis to check the preconditions.

Table 3: The privacy enhancing profile with the ≪enhance≫ stereotype to express the privacy enhancement of a system model.

Stereotype	Tags	UML Element	Description
≪enhance≫	strategy, pattern, PET	Behavior	Expressing the privacy enhancement of a Behavior.

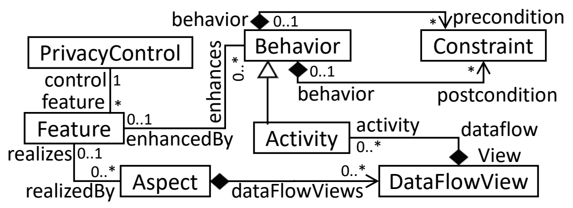


Figure 4: The metamodel demonstrating the underlying concepts of the privacy enhancement.

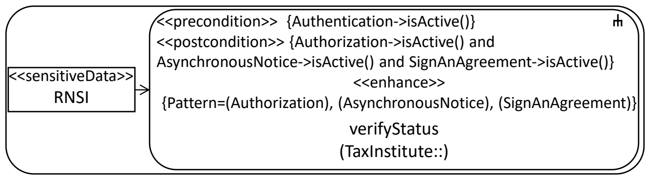

<<precondition>> {Authentication->isActive()}
<<postcondition>> {Authorization->isActive() and AsynchronousNotice->isActive() and SignAnAgreement->isActive()}
<<enhance>>
{Pattern=(Authorization), (AsynchronousNotice), (SignAnAgreement)}
verifyStatus
(TaxInstitute::)
<<sensitiveData>>
RNSI

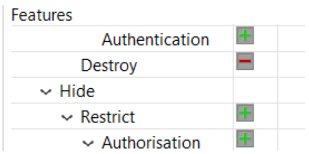
Figure 5: An excerpt of the privacy-enhanced system model.

In our example (see Sect. 3), the *RNSI* is processed for the unauthorized purpose *marketing*. A PIA identified 2 privacy targets at risks (**P1.4** and **P5.2**), and suggested three controls (**AP-2**, **DM-1**, **TR-1**). Concerning Table 1, these controls are mapped to four strategies, *Minimize, Hide, Notify* and *Consent*.

We show how a system model may be enhanced with the strategy *Hide*. Considering the feature model (Figure 3), *Hide* has three sub-strategies: *Restrict, Mix* and *Obfuscate*; the fourth one *Dissociate* is omitted for space reasons. We already mentioned that the features in Figure 3 are categorized based on four levels of rigour. Since, the *impact score*s calculated for **P1.4** and **P5.2** are *very high*, only sub-strategies with the rigour level *very strong* are considered, namely *Restrict* and *Mix*. With a similar argumentation, for sub-strategy *restrict*, the *Authorization* pattern may be considered and for sub-strategy *mix*, the *Anonymity Set*. Since the *Anonymity Set* excludes the *Notify* strategy and according to the initial set of mapped strategies, the *Notify* strategy has to be integrated into the system design, the sub-strategy *restrict*, and the *Authorization* pattern are used to enhance the model.

In Figure 5, the action *verifyStatus* is enhanced with the patterns that belong to the four strategies mentioned above. Since this action has to include the behavior of the patterns, it is demonstrated as a *CallBehaviorAction* (indicated by placing a rake-style symbol). Moreover, the *Authorization* pattern requires the *Authentication* pattern, this is expressed in the precondition constraint, and therefore, the action is not annotated with *Authentication*. In this activity diagram, the enhancement is performed using patterns, however, the enhancement can be applied by strategies or PETs.

An excerpt of the configuration of the feature model is provided in Figure 6. The postcondition constraints of an action (Figure 5)



Features	
Authentication	🟩 +
Destroy	🟥 −
⌄ Hide	
⌄ Restrict	🟩 +
⌄ Authorisation	🟩 +

Figure 6: An excerpt of the feature model's configuration.

lead to the respective features in the configuration being active (the green + symbol) automatically. On activating the *Authorization* pattern, the *Authentication* pattern is activated as well, due to precondition constraints and *require* relation.

Using and extending RAMs for privacy by design. In [20], the authors model security design patterns with *reusable aspect models* (*RAM*s) [16] to build a unified system of security design patterns that addresses multiple security concerns. While they do not consider privacy concerns and also focus on different diagram types than we do, we benefit from this work, since we can apply RAMs as well in order to encapsulate the required changes to the system model. For our privacy enhancement, we extend RAMs with a new kind of view called *data flow views*, which complement the existing structure and behavior views. As indicated in Figure 4, data flow views are modeled with activity diagrams, which are geared to capture privacy-relevant flows using object flows. Data flow views allow us to model the features as RAMs.

The activity diagram in Figure 5 is annotated with the *Authorization* pattern. This annotation specifies that the system model has to be revised by weaving the *Authorization* aspect into the system model. The *Authorization* aspect is demonstrated in Figure 7. The proposed *data flow view* in the provided RAM specifies that whenever a method is invoked on a protected class (pointcut), an authorization has to be performed before invoking the method (advice). If the access is granted (upon a successful evaluation of the request), the method will be invoked, otherwise an exception will be thrown (△ symbol). Since this aspect requires the *Authentication* aspect, first the *Authentication* RAM (not shown in this paper) has to be woven into the *Authorization* RAM. Moreover, handling the exception may be demonstrated in the *Authorization* aspect, or another RAM may be defined to handle such exceptions.

Similar to [16, 20], for weaving the aspects we use the generic weaver (GeKo) [18], a generic aspect-oriented model composition and weaving approach with available tool support. Furthermore, in [19], a formal specification for aspect weaving into activity diagrams is presented. We may use this approach to semantically apply a RAM weaver to activity diagrams.

As mentioned before, the process of applying a feature to a system model is based on the interrelations from the feature model and the level of rigour, identified by the final privacy impact assessment score. If two or more features from the same level of rigour are applicable to a system model, our model-based cost estimation approach from Sect. 4.2 identifies the appropriate feature. This approach is applicable to extended RAMs with *data flow views* as well.

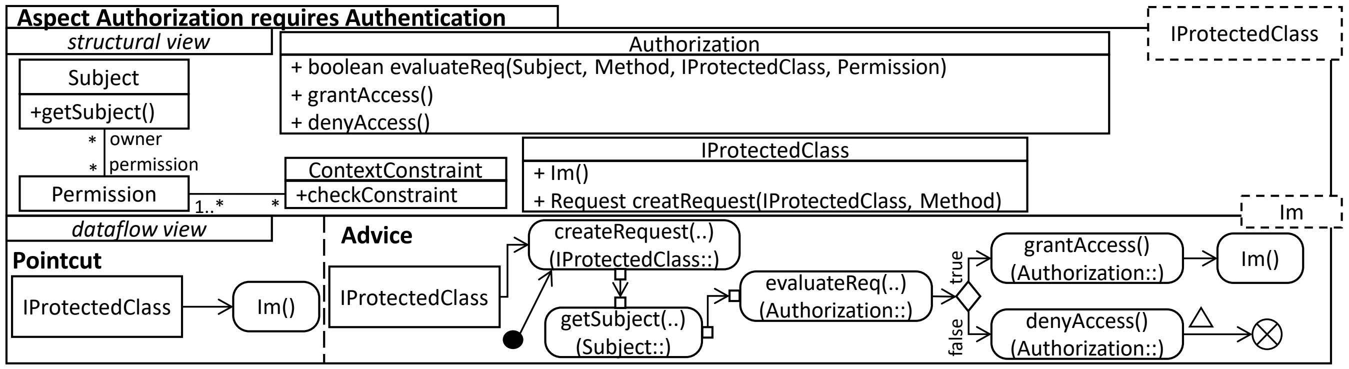


Figure 7: The *Authorization* aspect including a dataflow view.

Table 4: Information on the three case studies.

Case study	UML elements
City administration of Athens (Greece)	141
A hospital in Rome (Italy)	167
A ministry in Italy	309

The activity diagram in the *Authorization* aspect has one ILF and four EIs. Since after the *DecisionNode* (◇), only one action is chosen, the number of EIs is four and not five. Based on Table 2, the final FP number for the *Authorization* aspect is $19 = (1 \times 7) + (4 \times 3)$.

5 CASE STUDIES

We evaluate our methodology using three industrial case studies from the VisiOn EU project. These case studies represent three *Public Administration* (*PA*) systems, in which protecting the privacy of sensitive data is obligatory. The goal of the project was to develop a platform to analyze privacy-critical systems, and to enforce privacy agreements on the use of sensitive data. The PA administrators, after a training by a variety of webinars and training sessions, created three system models based on UML. Table 4 provides information on the case studies and their models. Each model had a class, a component, an activity, and a state diagrams. The first case study provides online administrative service to the citizens of Athens, Greece. The running example (Sect. 3) is based on the first case study. The second case study models the structure and the processes of a hospital in Rome, Italy. The third case study models the structure and the processes of a ministry in Italy.

After performing a PIA in each case study, we enhanced the system models by applying our methodology. We could successfully apply our methodology to all three case studies. For instance, the complete (*issuing birth certificate*) scenario (Sect. 3), eventually has been enhanced by seven design patterns. In Figure 5, we illustrated an example of the privacy enhancement in this case study. After performing a PIA, two more targets were at risk (besides the targets from the running example), namely **P1.8** *Ensuring limited storage*, and **P4.2** *Facilitating the rectification, erasure or blocking of data.* These privacy risks were the result of storing the RNSI in a database (*DataStoreNode*) without implementing an appropriate mechanism to remove the RNSI after it has been processed for the authorized purpose (*assessment*). To mitigate these risks, the system model has been enhanced by following controls and strategies: the control **DM-2** (the *minimize* and *hide* strategies), and the control **IP-1** (the *consent* sub-strategy). In all three case studies, we identified the processing of sensitive data for unauthorized purposes. Therefore, the system models were enhanced by **DM-1** *Minimization of Personally Identifiable Information*, and **AP-2** *Purpose Specification.*

Limitations. Our approach requires a set of default configurations. For instance the level of rigour for each feature must be specified before applying the methodology. Moreover, to estimate the cost of the features, we used a set of predefined complexities for the data and transactional functions. In fact, using more rigorous complexities refines the estimations.

Our cost estimation approach relies on the assumption that effort can be estimated reliably in terms of element-counting metrics.

In our evaluation, we only consider a limited number of models from three case studies, focusing on activity diagrams. In future we aim to study a larger set of cases with a more selection of diagram types to evaluate our methodology more broadly.

6 RELATED WORK

There exists a wide range of research on the PETs, privacy-patterns, controls, and design strategies. Moreover, there exists numerous methodologies, to support the privacy hardening and the selection of controls. We leverage related work, and aims to reflect different aspects coherently in privacy hardening.

In [20], a model-based approach built on a system of security design patterns (*SoSPa*) to systematically automate the application of multiple security patterns in a system development, is presented. In this approach, the selection of the most appropriate features is only based on the interrelations between the patterns. The risks, their severities, and the privacy enhancement costs are not supported.

In [26], a promising approach to apply runtime reconfigurations to adaptive software systems using the concepts of product lines is provided. In [28] the authors propose a framework to employ a planning technique to automatically select suitable features that satisfy both the stakeholders' functional and non-functional requirements. These approaches neither precisely consider data privacy nor support the privacy enhancement of a system design in the early phases. A research direction for future work is to investigate the integration of the system design and the run-time configurations.

In [9], a classification of the PETs is provided. In [8, 13] a set of privacy design strategies are introduced. These works do not provide any mechanism to enhance a system design and select an appropriate strategy or a PET. However, they provide a conventional foundation to build the feature model introduced in this paper.

In [25], the authors propose a set of privacy-enhanced extensions to the BPMN language for capturing data leakage in a business process. In this approach, the information flow analysis in the early phases of the system design is not supported.

7 CONCLUSION

We have introduced a methodology for enhancing system models with privacy controls to mitigate privacy violations during the design of a software system. Our methodology relies on an existing privacy impact assessment approach that identifies a set of risks and controls for mitigating the risks. Since the controls are rather abstract and cannot be directly integrated into the system design, we map them to more concrete privacy features, including strategies, design patterns and privacy enhancing technologies (PETs). To determine an adequate selection of features, we take into account the severity of the identified violations, possible interrelation and dependencies between the features, and the cost of integrating features into a system design. Furthermore, we performed an investigation of the interactions between the features, and captured the respective interrelations and dependencies in a feature model. To estimate the cost of the strategies, patterns, and PETs, we proposed a model-based cost estimation approach by customizing functional point analysis for applying to activity diagrams. Eventually, we introduced a UML profile to trace the privacy enhancement of a system model, and extended the concept of reusable aspect models to enhance a system behavior with appropriate privacy design strategies. We applied our methodology to three case studies.

ACKNOWLEDGEMENTS

This research was partially supported by: (I) Engineering Responsible Information Systems (University of Koblenz Landau) (II) Qu4lity Prjoect (EU's Horizon 2020 Program, No. 825030).

REFERENCES

[1] Amir Shayan Ahmadian, Jan Jürjens, and Daniel Strüber. 2018. Extending model-based privacy analysis for the industrial data space by exploiting privacy level agreements. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC 2018, Pau, France, April 09-13, 2018.* 1142–1149.

[2] Amir Shayan Ahmadian, Daniel Strüber, Volker Riediger, and Jan Jürjens. 2017. Model-Based Privacy Analysis in Industrial Ecosystems. In *Modelling Foundations and Applications - 13th European Conference, ECMFA 2017, Held as Part of STAF 2017, Marburg, Germany, July 19-20, 2017, Proceedings.* 215–231.

[3] Amir Shayan Ahmadian, Daniel Strüber, Volker Riediger, and Jan Jürjens. 2018. Supporting privacy impact assessment by model-based privacy analysis. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC 2018, Pau, France, April 09-13, 2018.* 1467–1474.

[4] Aj Albrecht. 1979. Measuring Application Development Productivity, In IBM Application Development Symp. *Proc. of IBM Application Development Symp..*

[5] Vieri Del Bianco, Luigi Lavazza, and Sandro Morasca. 2012. A Proposal for Simplified Model-Based Cost Estimation Models. In *PROFES 2012, Madrid, Spain, June 13-15, 2012 Proceedings.* 59–73.

[6] John J. Borking and Charles D. Raab. 2001. Laws, PETs and Other Technologies for Privacy Protection. *Journal of Information, Law and Technology* 2001, 1 (2001).

[7] Michael Colesky, Julio C. Caiza, José M. del Álamo, Jaap-Henk Hoepman, and Yod-Samuel Martín. 2018. A System of Privacy Patterns for User Control. In *Proceedings of ACM SAC Conference (SAC18).*

[8] Michael Colesky, Jaap-Henk Hoepman, and Christiaan Hillen. 2016. A Critical Analysis of Privacy Design Strategies. In *2016 IEEE Security and Privacy Workshops, SP Workshops 2016, San Jose, CA, USA, May 22-26, 2016.* 33–40.

[9] David Eckhoff and Isabel Wagner. 2018. Privacy in the Smart City - Applications, Technologies, Challenges, and Solutions. *IEEE Communications Surveys and Tutorials* 20, 1 (2018), 489–516.

[10] Eduardo Fernandez-Buglioni. 2013. *Security Patterns in Practice: Designing Secure Architectures Using Software Patterns* (1st ed.). Wiley Publishing.

[11] David Ferraiolo and Richard Kuhn. 1992. Role-Based Access Control. In *NIST National Computer Security Conference, NCSC 1992, October 13-16, 1992, Baltimore, Maryland, USA, Proceedings.* 554–563.

[12] Ronald Hes and John Borking (Eds.). 2000. *Privacy-Enhancing Technologies: The Path to Anonymity – Revised Edition.* Registratiekamer.

[13] Jaap-Henk Hoepman. 2014. Privacy Design Strategies - (Extended Abstract). In *ICT Systems Security and Privacy Protection - 29th IFIP TC 11 International Conference, SEC 2014, Marrakech, Morocco, June 2-4, 2014. Proceedings.* 446–459.

[14] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. 1990. *Feature-Oriented Domain Analysis (FODA) Feasibility Study.* Technical Report. Carnegie-Mellon University Software Engineering Institute.

[15] Christian Kästner, Thomas Thüm, Gunter Saake, Janet Feigenspan, Thomas Leich, Fabian Wielgorz, and Sven Apel. [n. d.]. FeatureIDE: A tool framework for feature-oriented software development. In *31st International Conference on Software Engineering, ICSE 2009.*

[16] Jörg Kienzle, Wisam Al Abed, Franck Fleurey, Jean-Marc Jézéquel, and Jacques Klein. 2010. Aspect-Oriented Design with Reusable Aspect Models. *Trans. Aspect-Oriented Software Development* 7 (2010), 272–320.

[17] Luigi Lavazza, Vieri Del Bianco, and Carla Garavaglia. 2008. Model-based functional size measurement. In *Proceedings of the Second International Symposium on Empirical Software Engineering and Measurement, ESEM 2008, October 9-10, 2008, Kaiserslautern, Germany.* 100–109.

[18] Brice Morin, Jacques Klein, Olivier Barais, and Jean-Marc Jézéquel. 2008. A Generic Weaver for Supporting Product Lines. In *Proceedings of the 13th International Workshop on Early Aspects (EA '08).*

[19] Djedjiga Mouheb, Dima Alhadidi, Mariam Nouh, Mourad Debbabi, Lingyu Wang, and Makan Pourzandi. [n. d.]. Aspect Weaving in UML Activity Diagrams: A Semantic and Algorithmic Framework. In *Formal Aspects of Component Software - 7th International Workshop, FACS 2010, October 14-16, Revised Selected Papers.*

[20] Phu Hong Nguyen, Koen Yskout, Thomas Heyman, Jacques Klein, Riccardo Scandariato, and Yves Le Traon. 2015. SoSPa: A system of Security design Patterns for systematically engineering secure systems. In *18th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MoDELS 2015, Ottawa, ON, Canada, September 30 - October 2, 2015.* 246–255.

[21] NIST and Emmanuel Aroms. 2012. *NIST Special Publication 800-53 Revision 4 Recommended Security and Privacy Controls for Federal Information Systems and Organizations.* CreateSpace.

[22] Object Management Group (OMG). 2017. UML 2.5.1 Specification. (2017).

[23] Marie Caroline Oetzel and Sarah Spiekermann. 2014. A systematic methodology for privacy impact assessments: A design science approach. *EJIS* 23, 2 (2014).

[24] Institute of Distributed Systems. 2018. EU Privacy Patterns. (2018). https://privacypatterns.eu/ Accessed: 2018-03-06.

[25] Pille Pullonen, Raimundas Matulevicius, and Dan Bogdanov. 2017. PE-BPMN: Privacy-Enhanced Business Process Model and Notation. In *Business Process Management - 15th International Conference, BPM 2017, September 10-15, 2017.*

[26] Karsten Saller, Malte Lochau, and Ingo Reimund. 2013. Context-aware DSPLs: model-based runtime adaptation for resource-constrained systems. In *17th International Software Product Line Conference co-located workshops, SPLC 2013.*

[27] Markus Schumacher, Eduardo B. Fernández-Buglioni, Duane Hybertson, Frank Buschmann, and Peter Sommerlad. 2005. *Security Patterns - Integrating Security and Systems Engineering.* Wiley.

[28] Samaneh Soltani, Mohsen Asadi, Dragan Gasevic, Marek Hatala, and Ebrahim Bagheri. 2012. Automated planning for feature model configuration based on functional and non-functional requirements. In *16th International Software Product Line Conference, SPLC '12.* 56–65.

[29] Harald Störrle. 2017. How are Conceptual Models used in Industrial Software Development? A Descriptive Survey. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, EASE 2017.*

[30] The European Parliament and the Council of the Europeam Union. 2016. Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data . *Official Journal of the European Union* L 119 (2016).

[31] G. W. van Blarkom, John J. Borking, and J. G. E. Olk. 2003. *Handbook of Privacy and Privacy-Enhancing Technologies: The case of Intelligent Software Agents.* Technical Report. Privacy Incorporated Software Agent Consortium, Den Haag.

[32] Zhiguo Wan, Jun-e Liu, and Robert H. Deng. 2012. HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing. *IEEE Trans. Information Forensics and Security* 7, 2 (2012), 743–754.