Model-based Privacy Analysis in Industrial Ecosystems

Amir Shayan Ahmadian[1], Daniel Strüber[1], Volker Riediger[1], Jan Jürjens[1,2]

[1] Institute for Software Technology, University of Koblenz-Landau, Germany
[2] Fraunhofer-Institute for Software and Systems Engineering ISST, Germany
ahmadian@uni-koblenz.de, strueber@uni-koblenz.de
riediger@uni-koblenz.de, http://jan.jurjens.de

Abstract. Article 25 of Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing and the free movement of personal data, refers to data protection by design and by default. Privacy and data protection by design implies that IT systems need to be adapted or focused to technically support privacy and data protection. To this end, we need to verify whether security and privacy are supported by a system, or any change in the design of the system is required. In this paper, we provide a model-based privacy analysis approach to analyze IT systems that provide IT services to service customers. An IT service may rely on different enterprises to process the data that is provided by service customers. Therefore, our approach is modular in the sense that it analyzes the system design of each enterprise individually. The approach is based on the four privacy fundamental elements, namely purpose, visibility, granularity, and retention. We present an implementation of the approach based on the CARiSMA tool. To evaluate our approach, we apply it to an industrial case study.

1 Introduction

A main problem for IT service providers is to avoid data breaches and provide data protection. According to a global survey [1], 88% of people are concerned about who can access their private data. In Germany, 72% of people expect the government to keep out of their personal data.

Article 25 of Regulation (EU) 2016/679 refers to data protection by design and by default [3]. This requires that service providers verify if the required privacy levels are fulfilled according to legal requirements and customers' privacy preferences. Furthermore, they must implement appropriate technical and organizational measures in an effective manner, and integrate proper safeguards into the processing to support such requirements.

There exist a range of privacy enhancing technologies (PETs) [6,23,16,15,14], which provide strong privacy guarantees in different domains. However, according to Spiekermann [29,28], *privacy and data protection by design and by default* are powerful terms, and include more than the process of uptaking a few PETs. Cavoukian [10], who first introduced the term *privacy by design (PbD)*, defines

PbD as the idea to integrate privacy and data protection principles in a system's design, and to recognize privacy in an enterprise's management processes.

Based on these considerations, PbD implies the design of a system must be analyzed with regard to privacy preferences and, where necessary, be improved to technically support privacy and data protection. Article 5 of Regulation (EU) 2016/679 stipulates six principles for the processing of personal data: Personal data must be (a) processed lawfully, (b) collected for specified and legitimate *purposes*, (c) adequate and limited to what is necessary regarding the purposes (*granularity*), (d) accurate and kept up to date, (e) kept no longer than necessary (*retention*), and (f) protected against unauthorized processing (*visibility*). These principles correspond to the *key elements of privacy* introduced in Barker et al.'s seminal taxonomy [7]: *purpose*, *visibility*, *granularity*, and *retention*.

System-level privacy analysis is particularly challenging in today's digital society, where industrial ecosystems play a key role. Specifically, an enterprise may depend on or cooperate with other enterprises to provide an IT service to a service customer. For instance, an enterprise as a service customer of an insurance enterprise may send personal data of its employees to the insurance enterprise to issue health insurance contracts for them. The insurance enterprise must assess the solvency of the employees before issuing an insurance contract. Therefore, the personal data of each employee will be transmitted to a financial institute for the relevant assessments. Performing a privacy analysis on such a system's design requires analyzing the relevant components of the insurance enterprise, and the financial institute; and the respective interfaces between the components. To support cases where the system design of the relevant enterprises and components are not entirely available, each enterprise must be analyzed individually.

In this paper, we investigate the following research questions: **RQ1:** How can a modular privacy analysis be performed on the system's design of the IT services in industrial ecosystems, where an IT service is the result of cooperation of different service providers? **RQ2:** How can be analyzed if the key elements of privacy are supported by the systems' design of services that process personal data?

To address these questions, we present a model-based approach to support the analysis of the system's design concerning privacy. Using the system models, the privacy requirements are considered from early stages of the system's design and the development process. Our approach is modular in the sense that it analyzes the system design of each enterprise separately. The approach is based on the fundamental taxonomy of the four privacy key elements [7]. We integrated it into the CARiSMA tool [2], which was originally designed to make a security analysis based on the UMLsec profile available to developers [21] and is now extended to address privacy.

The paper is organized as follows. In Section 2, the necessary background is provided. In Section 3, we describe our approach on model-based privacy analysis. In Section 4, we evaluate our approach using a case study. In Section 5, we discuss related work. Finally, in Section 6, we conclude.

2 Background

Below, we present the necessary background for this paper.

2.1 The four key elements of privacy

In what follows, we briefly describe the four fundamental privacy elements presented in [7]: purpose, visibility, granularity, and retention.

- **Purpose** is the basic element of data privacy. It indicates the authorized reasons to access data [13]. Service providers must record and track the purposes for which the data is collected and is processed.
- **Visibility** indicates who is allowed to access or use the data provided for an authorized purpose. In other words, visibility controls the number and kind of users who can access the data.
- **Granularity** refers to characteristics of data that could be used to facilitate proper use of the data, where there exists different valid accesses for various purposes. In other words, data granularity specifies how much precision is provided in response a query. This is important when the service customer requires the service provider to provide personal data to a third party [13].
- **Retention** refers to the need to restrict access or remove the data after they have been used for the intended purposes.

2.2 Model-based security analysis using UMLsec

UMLsec [21] provides a model-based approach to develop and analyze security critical software systems, in which security requirements such as confidentiality, integrity, and availability are expressed within UML diagrams [24]. UMLsec is provided as an UML profile, using the standard UML extension mechanism. In UMLsec, different stereotypes and tags are used to annotate UML diagrams with security properties. The CARiSMA tool [2] performs the corresponding security analysis for security properties such as secure information flow [18] and has been applied to various industrial applications (e.g. [19]). Some of the analysis techniques have also been ported to the code analysis level [12]. UMLsec does not support the analysis of a system concerning privacy.

3 Model-based Privacy Analysis

The terms that are used in this paper are based on the terms and definitions of Regulation (EU) 2016/679. According to this Regulation, a data controller determines the purposes and the means for the processing of personal data (privacy preferences). In our work, a service customer is a data controller, who provides personal data and specifies the privacy preferences of these data. A data processor processes personal data on behalf of the controller. When we talk of service providers, we mean either a data processor, who directly processes the provided

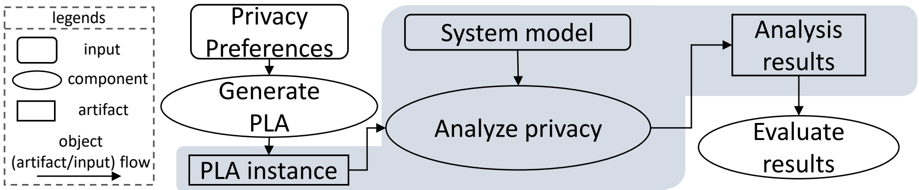


Fig. 1: Model-based privacy analysis by exploiting PLAs.

data, or a data controller, who transfers to other data processors the data and their privacy preferences specified by the service customer.

The work presented in this paper is a part of our ongoing research [5] to provide a method on privacy analysis of IT systems by exploiting privacy level agreements (PLAs). PLAs are appendixes to service level agreements, and offer a structured way to communicate the level of personal data protection provided by a service providers to service customers. PLAs are based on EU personal data protection and privacy legal requirements [11]. Figure 1 presents the overview of this privacy analysis method. In [5], we provided a meta-model formalizing PLAs to specify privacy preferences. In this paper (as it is highlighted in Figure 1), given privacy preferences (contained in a PLA) and system model, we introduce an approach to model-based privacy analysis based on the four key privacy elements. Following the privacy analysis, the analysis results are evaluated. This evaluation is out of scope of this paper. The results of this evaluation are: (I) Privacy-centric questions, which are used to define privacy questionnaires to collect additional feedback from data controllers on privacy preferences. (II) Privacy measures to protect personal data (adheres to PLA outline, Section 4). (III) Conflicts between the system model and privacy preferences.

3.1 The modular privacy analysis

The scenario introduced in Section 1 is illustrated in more detail in Figure 2. Car manufacturing *enterprise A* wants to issue a health insurance for its employees and, therefore, sends personal data of the employees to insurance *enterprise B*. Together with the personal data, *enterprise A* specifies the privacy preferences. For instance, it may specify that *enterprise B* is not authorized to use the credit card number and the birthdate of an employee for marketing purposes, and that it must delete these information five years after the termination of health insurance contracts. Such privacy preferences are specified in *PLA-x*, which is concluded between the car manufacturing enterprise and the insurance enterprise. To issue a health insurance for an employee, the insurance enterprise needs to assess the financial status of the employee and therefore, sends the personal data of the employee to *enterprise C* (a financial institute). *Enterprise A* is not aware of this data transmission. A PLA must be concluded between the insurance enterprise and the financial institute (*PLA-y*), in which the privacy preferences of *enterprise A* are included. Similarly, the financial institute may send the personal data of the employee to a tax institute (*enterprise D*) to collect some information about the employee.

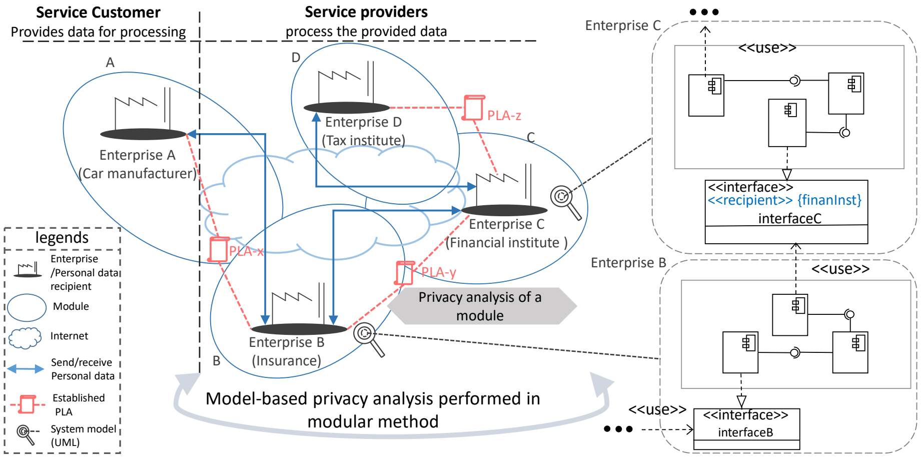


Fig. 2: An illustration of an industrial ecosystem containing four modules.

To perform a privacy analysis on such a system design, where several enterprises process personal data, we need to perform a modular analysis, in which each enterprise and its corresponding interfaces are analyzed individually. The reasons to perform modular privacy analysis are: (I) PLAs are needed as input to privacy analysis. Since *PLA-y* might differ from *PLA-x* and contain additional privacy preferences, the financial institute (*enterprise C*) must be analyzed individually. (II) System models of all data processors and data controllers may not be available. In case that the system model for one of the involved enterprises is not available, a privacy analysis is still desirable. In the scenario, if the system model of the financial institute is not available, a privacy analysis on the insurance enterprise is still possible. Since the privacy preferences of *enterprise A* are contained in *PLA-y*, at some point a privacy analysis on the financial institute may be performed (when the system model is available).

Definition 1. *(Module) A module is a data processor or a data controller together with its all interfaces to other recipients, where according to Regulation (EU) 2016/679 a recipient is a controller, processor, or data subject (personal data owner), to which personal data are disclosed.*

Per Definition 1, Figure 2 contains four modules. *Module A* is a data controller, i.e. *module A* acts as a service customer that provides personal data and specifies the privacy preferences. *Modules B, C, and D* are recipients and cooperate in the processing of the data provided by *module A*. Therefore, these three modules are analyzed separately to verify if they support the privacy preferences.

According to Definition 1, this analysis needs to address the interfaces between the involved modules. In the right part of Figure 2, the structures of *modules B and C* are specified using two UML component diagrams, together with the respective provided interfaces. The interface provided by *module C* and used by *module B* is annotated with ≪recipient≫ specifying that the personal data is transferred from *module B* to *module C*. Later, we will introduce the privacy profile that is used to annotate UML diagrams with annotations such as ≪recipient≫.

3.2 Model-based privacy analysis based on the four fundamental privacy elements

Different UML diagrams may be used to specify the structure and behavior of a system. Figure 3 shows an activity describing the processing of credit card number within the insurance scenario. The activity is annotated with ≪dataPrivacy≫ {creditCardNo}, specifying that a piece of sensitive data (creditCardNo) is processed in this activity. According to this activity, the sensitive data will be stored in a database (*storeNumber*), and will be sent to a financial institute (*sendToBank*) to check the solvency of the data subject (*verifySolvency*).

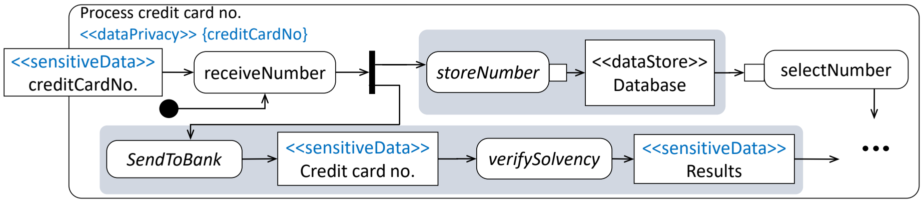


Fig. 3: Design model excerpt (*process credit card no.* Activity) which highlights the need to perform a privacy analysis.

Based on the four privacy elements, we need to analyze the activity to verify if (I) the *creditCardNo* is only processed for the purposes that are mentioned in the privacy preferences, (II) access to the sensitive data is restricted to authorized persons, (III) the granularity level is respected when sensitive data are sent to the bank, and (IV) deletion or restriction mechanisms are in place to ensure that sensitive data stored in a database, such as *creditCardNo*, are eventually deleted or restricted. To this end, we propose four corresponding privacy checks. In what follows, first we give a high-level description of these checks:

- **Purpose-check**: First, this check analyses the system's behavior and structure to identify the system operations that process personal data, and their objectives (purposes). Moreover, it determines if operations that process personal data belong to different systems. Finally, the objectives of the operations are compared with the purposes specified in the privacy preferences by the data controller.
- **Visibility-check**: The check identifies all the data recipients, and it verifies if they are authorized to process personal data. Moreover, concerning one module, this check verifies who is allowed to access personal data. This could be verified by identifying all owners of system operations that process personal data.

- **Granularity-check**: According to [13], granularity is important when personal data are disclosed to other recipients. Based on the interfaces to other recipients, this check verifies if the granularity level is respected by data transmissions.
- **Retention-check**: In the first step, the check verifies if appropriate operations exist to restrict or delete the personal data. Moreover, the check analyzes the system behavior to determine if such an operation will be eventually applied.

The annotations used in Figure 3 enable these privacy checks. In the following section, we introduce a complete list of these annotations. Since the systems are modeled using UML, we use a UML extension mechanism to specify them.

3.3 UML privacy extension

As a basis to implement the privacy checks described in Section 3.2, we introduce two UML extensions. First, the *privacy* profile, allowing elements in UML models to be annotated with privacy-specific information. Second, the *rabac* profile, allowing the generation and enforcement of access control policies for such elements, using the *role- and attribute based access control* model (RABAC, [17]). The *rabac* profile is an extension of UMLsec's *rbac* profile [21]. On top of *rbac*'s *role* and *right* tags, *rabac* allows a refined control management using an *attributeFilter* tag. In nutshell, by using attributes, there is no need to increase the number of roles in a system in many cases, and the problem of role explosion will be prevented. We use *rabac* to define the visibility check, introducing it as a separate profile, since it is not specific to privacy. The complete list of stereotypes together with their tags is provided in Figure 4.

Stereotype	Tags	UML Element	Description
Privacy profile			
≪dataPrivacy≫	data	Behavior	enfoces privacy analysis
≪sensitiveData≫		NamedElement	personal data [3]
≪recipient≫	enterprise	NamedElement	data recipient [3]
≪granularity≫	level	Parameter	the granularity level
≪objective≫	purpose	BehavioralFeature	purposes of operations
rabac profile			
≪abac≫	roles, rights, attributeFilter	Package	enforces role-attribute-based access control
≪abacAttribute≫	name	Operation	rabac for an attribute
≪abacRequire≫	accessRight, filter	Operation	rabac for an operation

Fig. 4: Privacy and *rabac* profiles.

Privacy profile The terms and names used for the stereotypes and the tags comply with the terms and the definitions of *Regulation (EU) 2016/679* [3].

≪**dataPrivacy**≫: Behavioral specification mechanism may be annotated with this stereotype, specifying the existence of personal data in the behavior which is modeled using the corresponding diagram. Tag *data* specifies a set of personal data.

≪**sensitiveData**≫: A NamedElement may be annotated with this stereotype specifying the element is or contains sensitive data. The definition of sensitive data complies with the definition of personal data provided in Article 4 (1) of Regulation (EU) 2016/679 and particularly adheres to the definition of special categories of personal data provided in Article 9. Additionally, and regarding the controller's preferences, a piece of data that must not be revealed or disclosed could be also annotated with this stereotype.

≪**recipient**≫: A NamedElement may be annotated with this stereotype *recipient* stating that the element is a controller, or a processor, to which the sensitive data are disclosed.

≪**granularity**≫: A Parameter may be annotated with this stereotype together with its tag *level* specifying the level of the precision of data provided in response a query. In other words, granularity is assumed as a new attribute for a parameter, where a parameter ([24]) is an argument used to pass information into or out of an invocation of a behavior.

≪**objective**≫: A BehavioralFeature such as an Operation may be annotated with this stereotype together with its tag {purpose} specifying the purposes of the operation (BehavioralFeature). Tag {purpose} specifies a set of purposes for an operation.

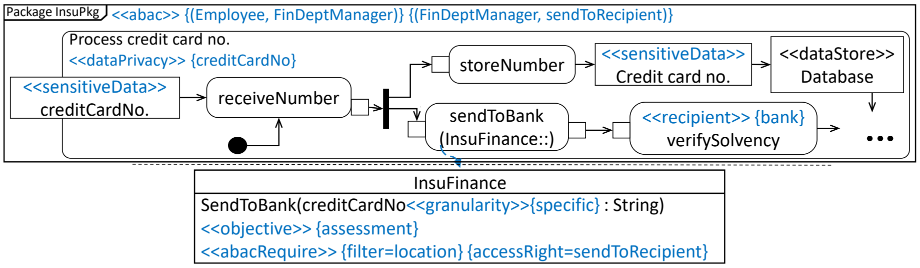

Package InsuPkg
<<abac>> {(Employee, FinDeptManager)} {(FinDeptManager, sendToRecipient)}
Process credit card no.
<<dataPrivacy>> {creditCardNo}
<<sensitiveData>>
creditCardNo.
receiveNumber
storeNumber
<<sensitiveData>>
Credit card no.
<<dataStore>>
Database
sendToBank
(InsuFinance::)
<<recipient>> {bank}
verifySolvency
•••
InsuFinance
SendToBank(creditCardNo<<granularity>>{specific} : String)
<<objective>> {assessment}
<<abacRequire>> {filter=location} {accessRight=sendToRecipient}

Fig. 5: Design model excerpt annotated with the privacy and *rabac* profiles.

Figure 5 shows an excerpt from the activity provided in Figure 3, and a class from a class diagram of the insurance system. For space reasons, we only show the relevant actions and classes. The annotation ≪`dataPrivacy`≫ {`creditCardNo`} specifies that a piece of personal data (*creditCardNo)* is processed in this activity. ≪`recipient`≫ {`bank`} specifies that the *verifySolvency* Action is performed not in the *insurance*, but in the recipient *bank*. Based on the annotation ≪`granularity`≫ {`specific`}, the granularity-check must analyze the parameters that are annotated with ≪sensitiveData≫ concerning the granularity level *specific*. The *sendToBank* Action is a CallOperationAction, i.e., it invokes an

Operation as specified in the *InsuFinance* Class. This operation is annotated with ≪`objective`≫ {`assessment`} specifying *assessment* as the purpose of the operation and, transitively, the purpose of the *sendToBank* Action.

rabac profile *rabac* enables the verification of visibility requirements on personal data. For each operation of a system, a set of data subjects with different roles, who are authorized to process personal data, is defined. Throughout the analysis, this information is compared to the provided privacy preferences. In what follows, the stereotypes of *rabac* together with their tags are explained:

≪**abac**≫: A Package is annotated with this stereotype and its tags, namely *roles*, *rights*, and *attributeFilter* to specify *role-attribute-based access control* is enforced in the system model. The values of *roles* and *rights* are tuples of the following form: (*dataSubject*, *associatedRole*), and (*associatedRole*, *accessRight*) respectively. The former one links a role to a data subject, while the later one associate a right to a role (similar to *rbac* [21]). Tag *attributeFilter* specifies a set of attributes (defined in classes). Based on these attributes, it is possible to define access rights.
≪**abacAttribute**≫: An Operation may be annotated with this stereotype, with tag *name* to specify a specific attribute with a corresponding value to invoke the operation.
≪**abacRequire**≫: An Operation or a Transition may be annotated with *abacRequire* with tags *filter*, and *accessRight* to specify the respective attribute and the access right to invoke the operation. Tag *accessRight* enables on to identify the associated role and data subject that are authorized to perform the operation.

In Figure 5, Operation *sendToBank* is annotated as follows: ≪`abacRequire`≫ {`filter = location`} {`accessRight = sendToRecipient`}. This means that the *accessRight* for this operation is *sendToRecipient.* Considering ≪abac≫, the associated role for this *accessRight* is *FinDeptManager*, who is allowed to invoke the *sendToBank* Operation.

In the following section, we explain how theses stereotypes are used to perform a privacy analysis.

3.4 Privacy checks

Generally, we explain the privacy checks using UML Activities (for detailed information on Activities see [24]). Before we explain the privacy checks, we need to define the privacy preferences to be specified by a data controller who provides the personal data.

Definition 2. *(Privacy Preferences) Let P be a partially ordered set of all defined purposes, V be a partially ordered set of all subjects to whom the data is visible, G be a set of all possible granularity levels, and R be a set of retentions* *conditions. The privacy preferences of a piece of personal data pd is defined as a tuple:*

$$PrP_{pd} = (P', V', g, r)$$

where $P' \subseteq P$, $V' \subseteq V$, $g \in G$, *and* $r \in R$.

According to Definition 2, purpose and data subject sets are defined as partially ordered sets. This enables us to organize these two sets in lattice structures, where each node presents a purpose or a data subject and each edge represents a hierarchical relation between two purposes or data subjects where they subsume each other, i.e. one purpose or data subject is more specific than the other. For instance, concerning visibility, in a lattice which organizes the data subjects, *marketing department* and *sales department* are the descendent (children) of *business department*, and are more specific (for more information, see [13]).

Purpose-check. The upper part of Figure 6 shows an excerpt from a design model. The lower part demonstrates two lattices, namely a purpose-lattice, and a visibility-lattice (simplified for space reasons). The purpose-lattice presents the set of all possible purposes in the system. The parts shown with dashed lines specify the privacy preferences for *creditCardNo*, which are specified by the data controller (car manufacturing enterprise). For instance, in the purpose-lattice, the *creditCardNo* may be processed for purpose *assessment* and its child purposes.

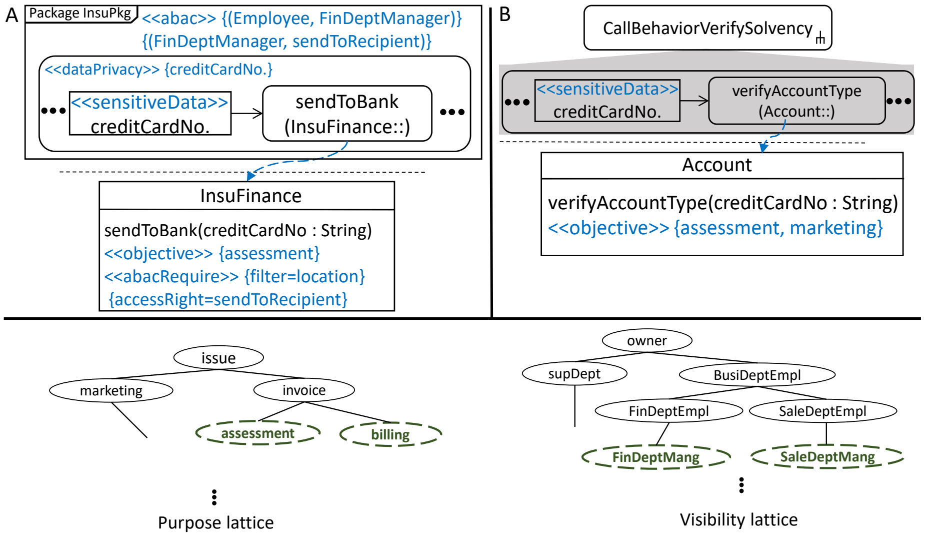

A
Package InsuPkg
<<abac>> {(Employee, FinDeptManager)}
{(FinDeptManager, sendToRecipient)}
<<dataPrivacy>> {creditCardNo.}
<<sensitiveData>>
creditCardNo.
sendToBank
(InsuFinance::)
InsuFinance
sendToBank(creditCardNo : String)
<<objective>> {assessment}
<<abacRequire>> {filter=location}
{accessRight=sendToRecipient}
B
CallBehaviorVerifySolvency
<<sensitiveData>>
creditCardNo.
verifyAccountType
(Account::)
Account
verifyAccountType(creditCardNo : String)
<<objective>> {assessment, marketing}
issue
marketing
invoice
assessment
billing
Purpose lattice
owner
supDept
BusiDeptEmpl
FinDeptEmpl
SaleDeptEmpl
FinDeptMang
SaleDeptMang
Visibility lattice

Fig. 6: Design model excerpt and lattices. Purpose, and visibility-checks use the highlighted stereotypes.

Based on the purpose-lattice and the purposes defined in the provided privacy preferences, the purpose-check for each action of the activity processing a piece of personal data (annotated with ≪sensitiveData≫) identifies the objective. According to the specification of Activities [24], two cases may happen:

(I) For a **CallOperationAction** (Figure 6, part A), the corresponding operation in the class diagram is directly identified. Then the purpose-check compares the objectives of the operation to the privacy preferences (purpose-lattice). For instance, in our example, *assessment* is specified as the purpose for the *sendToBank* Operation. Since *assessment* is included in the privacy preferences (purpose-lattice) as an authorized purpose to process *creditCardNo*, the check is successful.

(II) For a **CallBehaviorAction** (Figure 6, part B), the activity invoked by the CallBehaviorAction is analyzed. Similar to case (I), if the actions of this activity are CallOperationActions, the objectives of the corresponding operations are identified and compared with the privacy preferences. For instance, in Figure 6, a mentioned purpose of the *verifyAccountType* Operation is *marketing*, which is not specified as a valid purpose in the privacy preferences. Therefore, the purpose-check is not successful.

Visibility-check. Similar to purpose-check, the visibility-check assumes a lattice such as the one in Figure 6. The visibility-lattice presents all possible data subjects to whom *creditCardNo* may possibly be disclosed. The dashed parts specify the authorized data subjects according to the privacy preferences. Considering the annotations ≪`abacRequire`≫ {`accessRight = sendToRecipient`} and ≪`abac`≫ {`(FinDeptMang, sendToRecipient)`}, the visibility-check identifies that the model specifies *FinDeptMang* (finance department manager) processes *creditCardNo*. Since *FinDeptMang* is authorized to process *creditCardNo* in the privacy preferences, the check is successful.

Granularity-check. The supported granularity levels are *none, existential, partial*, and *specific*. Obviously, *none* means that nothing about a piece of personal data may be revealed in response to a query. *Existential* means that a query may only be answered by specifying if a piece of personal data exists or not. *Partial* means that a piece of data is revealed only partially. For instance, for numeric data, a range of numbers is specified. *Specific* means that a piece of data is precisely provided in response to a query.

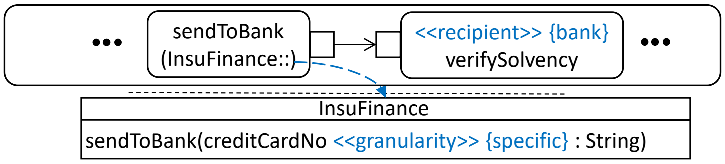


Fig. 7: Design model excerpt. The granularity-check uses the highlighted stereotypes.

Granularity is important when a piece of personal data is transferred to a recipient, that is, another enterprise. In to Figure 7, the granularity-check first identifies if a piece of personal data is transferred to another enterprise for processing, using the ≪sensitiveData≫ and ≪recipient≫ annotations. Afterwards,

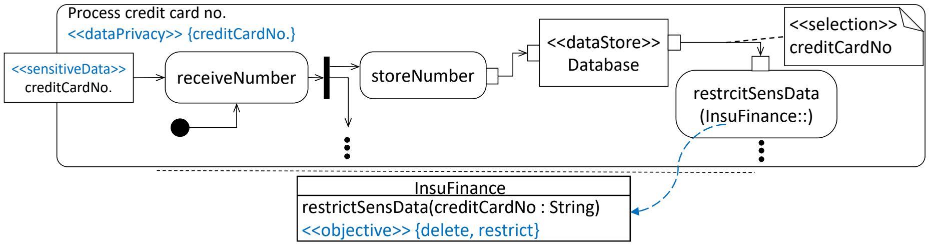


Fig. 8: Design model excerpt. Retention-check can be executed using the highlighted stereotypes.

similar to the purpose-check, the corresponding Operation is identified by verifying the action and the respective class. Using the ≪granularity≫ specified for the parameters, the level of granularity used by the operation will be ascertained. The specified level will be compared to the granularity level given in the privacy preferences. For instance, in Figure 7, if we assume the granularity level specified by the data controller in the privacy preferences is partial, the check fails.

Retention-check. The retention-check verifies that whenever a piece of personal data is stored in a database, an action exists to eventually restrict access to or delete this data. According to the specification of Activites [24], a node annotated with ≪dataStore≫ acts as a database holding object tokens.

The retention-check verifies if in an activity a piece of personal data (an object annotated with ≪sensitiveData≫) is stored in a node annotated with ≪dataStore≫. Afterwards, the retention-check verifies if a selection on the ≪dataStore≫ node exists, which retrieves the piece of personal data, and subsequently an action with *restrict* or *delete* purpose exists that restricts or deletes the piece of personal data.

For instance, in Figure 8, *creditCardNo* is stored in a database. This implies that a selection shall retrieve *creditCardNo* and an action of purpose *deletes* or *restricts* shall process *creditCardNo* before the activity terminates. If such an action does not exist, then the retention-check is not successful. Similar to the purpose-check, by mapping the action to a respective operation in a class, its purposes are identified.

4 Case study

To evaluate our approach, we applied it to an industrial case study, namely *birth certificate registration* scenario in Municipality of Athens (MoA). MoA is a public administration (PA) in Athens. This case study is one of the case studies of *VisiOn* research project [4]. In this project a privacy platform to evaluate and analyze privacy levels of a PA system, and generate a privacy level agreement between a citizen and a PA to enforce privacy policies, is developed.

In Figure 9, an excerpt from the architecture of *VisiOn Privacy Platform (VPP)* is presented. Three components of VPP architecture are represented.

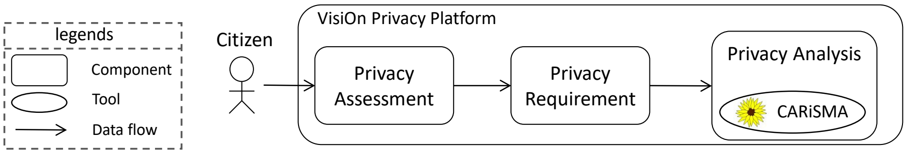
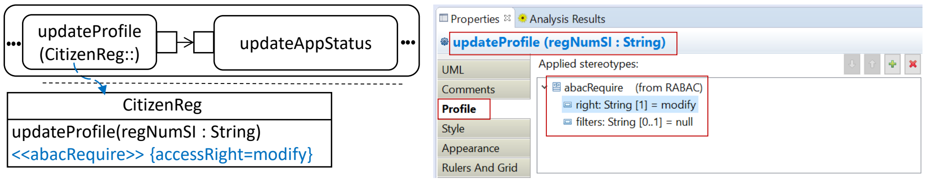


Fig. 9: An excerpt from the architecture of *VisiOn Privacy Platform (VPP)*

(I) Privacy assessment, providing a questionnaire to obtain the privacy preferences of a citizen, (II) Privacy requirement, determining privacy requirements based on the preferences of a citizen. (III) Privacy analysis, analyzing the system model of a PA considering privacy requirements. Our model-based privacy analysis approach is implemented and integrated into the privacy analysis component. The implementation of our approach is based on the CARiSMA tool [2]. CARiSMA enables the developers and IT system designers to annotate UML diagrams with security-specific information, using *UMLsec*, and privacy-specific information, using the profiles introduced in this work. The annotated diagrams can be analyzed using privacy and security checks.

MoA is in the process of developing a new system called MACS. MACS shall provide different online services to citizens, such as issuing a birth certificate. To provide such services, MoA requires citizen's personal data such as their *Registry Number of Social Insurance (RNSI)*. Moreover, MoA may cooperate with other public administrations such as central tax institute and financial institutes.

In our case study, we model the *birth certificate registration* using a selection of UML diagrams. Our privacy and *rabac* profiles are used to annotate these diagrams with privacy-specific stereotypes. Figure 10 illustrates the annotation of an operation with the ≪abacRequire≫ stereotype from the *rabac* profile, as facilitated by the Eclipse CARiSMA perspective.



updateProfile(regNumSI : String)
<<abacRequire>> {accessRight=modify}

updateProfile (regNumSI : String)
abacRequire (from RABAC)
right: String [1] = modify
filters: String [0..1] = null

Fig. 10: Excerpts from class diagram and activity diagram of *birth certificate registration* scenario, together with the property view of the CARiSMA perspective in Eclipse.

The four privacy checks, as implemented in CARiSMA, analyze the system model concerning the privacy preferences of a citizen. If a check is not successful, it will generate an analysis report. For instance, in Figure 10, if accessRight *modify* is defined as an accessRight for *MACSadministrator*, but a citizen specified in the questionnaire that only department manager (*MACSDeptManager*) is allowed to process *RegNumSI*, the visibility-check is not successful, and a report specifying the violation and its reasons is generated. Such reports later will be evaluated and may result in the application of an appropriate privacy-preserving measure, or the generation of privacy-centric questions.

In this case study, all the relevant classes of the MACS system together with corresponding activities to describe the behavior of the system are modeled. Moreover, the interfaces and relevant classes and related activities of a bank, which is responsible for approving an invoice issued by MACS, are modeled.

Results of this case study include that our approach is successfully applied to a software system in industrial ecosystem with complex structure and behavior. More specifically, concerning the research questions investigated in this work, results include the following: **RQ1:** We defined the term module in industrial ecosystems concerning the system's design of IT services, and we introduced modular privacy analysis in such ecosystems, in the sense that enterprises that cooperates with each other to process personal data are analyzed separately. **RQ2:** We introduced a UML privacy extension (the privacy profile, and *rabac* profile) to enable four privacy checks to analyze a system model based on the key four privacy elements.

Since generally, CARiSMA is based on the analysis of the system models that are modeled using UML diagrams, to perform the privacy analysis using the four checks, the MoA system and the cooperating systems must be modeled using UML. Based on these considerations, in this paper the privacy profile is defined for UML elements. However, concerning the description of each stereotype (Section 3.3), the privacy concepts may be adapted for other modeling languages. Furthermore, concerning the fact that IT systems may be modeled using different modeling languages, a transformation may be defined to perform the privacy checks on such models. Moreover, by evaluating the results of the case study, we observed that by performing privacy analysis on system models, not all privacy issues of the systems may be handled. However, as we previously mentioned (Section 1), the system models enable us to consider key privacy elements from early stages in the system's design, and verify if privacy preferences are supported.

5 Related Work

Generally, model-based privacy analysis has attracted little attention in the scientific literature so far. A possible explanation is the earlier lack of legal incentives driving its adoption process. Our work is motivated particularly by Article 25 of Regulation (EU) 2016/679, which is the current state of European legislation.

In [30], an extension of privacy agreement levels by implementing access purposes for individual personal information in a lattice structure is introduced. This approach enables service customers to control the use of individual data. However, in this approach no privacy analysis regarding customer preferences is performed.

In [13], a lattice-based privacy-aware access control model is introduced. In their approach, they provide a concrete privacy enhancing technique to control the access to a system concerning the four key privacy elements. However, using this approach, one is not able to perform a privacy analysis to verify if privacy preferences are supported by a system's design, so that the design can be improved when necessary.

UMLsec [21,20] provides an approach to develop and analyze security critical software, in which security requirements such as integrity, availability, and confidentiality are specified in system models. Moreover, the security analysis techniques have been integrated with the requirements elicitation phase [9,27]. However, UMLsec analysis does not consider privacy.

In [8], the authors provide a UML profile for privacy-aware applications. This profile enables one to describe a privacy policy that is applied by an application and keep track of which elements are in charge of enforcing it. This profile does not enable one to analyze a system's design.

In [22], the authors propose a method (PriS) for incorporating privacy user requirements into the system design process. PriS provides a methodological framework to analyze the effect of privacy requirements on organizational processes. The authors focus on the integration between high-level organizational needs and IT systems. A privacy analysis is not conducted on a system's design.

In [25,26], the authors provide a model-based privacy 'best practice', and a variety of guidelines and techniques to assist experts and software engineers to consider privacy when the systems are designed. However, they only focus on top-level security and privacy goals, and they do not perform a privacy analysis.

6 Conclusion

We have introduced a modular model-based privacy analysis approach for industrial ecosystems. The approach is based on four key privacy elements, namely purpose, visibility, granularity and retention. A set of stereotypes are introduced to express key privacy elements within the diagrams in a UML system specification. This annotations enable four privacy checks, which adhere to the four key privacy elements. The approach is integrated into VisiOn project, in which a platform for privacy analysis of public administration systems is provided.

As we mentioned in Section 1, privacy by design implies that the system's design of IT services must be analyzed to verify if the required privacy levels are fulfilled, and where necessary appropriate technical and organizational measures must be implemented to support privacy and data protection. In the future, we will investigate how the results from our privacy analysis can be evaluated to identify proper technical and organizational measures.

ACKNOWLEDGEMENTS

This research was partially supported by the research project Visual Privacy Management in User Centric Open Environments (supported by the EU’s Horizon 2020 program, Proposal number: 65364).

References

1. Personal data in the cloud: The importance of trust. Tech. rep., Fujitso Global Business Group, Tokyo 105-7123, JAPAN (Sep 2010)
2. CARiSMA (2016), https://rgse.uni-koblenz.de/carisma/
3. 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 (2016)
4. VisiOn Project (2016), http://www.visioneuproject.eu/
5. Ahmadian, S., Jürjens, J.: Supporting model-based privacy analysis by exploiting privacy level agreements. In: 8th IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2016) (2016)
6. Antignac, T., Le Métayer, D.: Privacy by Design: From Technologies to Architectures, pp. 1–17. Springer International Publishing (2014)
7. Barker, K., Askari, M., Banerjee, M., Ghazinour, K., Mackas, B., Majedi, M., Pun, S., Williams, A.: A Data Privacy Taxonomy, pp. 42–54. Springer Berlin Heidelberg (2009)
8. Basso, T., Montecchi, L., Moraes, R., Jino, M., Bondavalli, A.: Towards a uml profile for privacy-aware applications. In: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. pp. 371–378 (Oct 2015)
9. Breu, R., Burger, K., Hafner, M., Jürjens, J., Popp, G., Wimmel, G., Lotz, V.: Key issues of a formally based process model for security engineering. In: Sixteenth International Conference "Software & Systems Engineering & their Applications". Paris (2003)
10. Cavoukian, A., Chibba, M.: Advancing privacy and security in computing, networking and systems innovations through privacy by design. In: Proceedings of the 2009 conference of the Centre for Advanced Studies on Collaborative Research, November 2-5, 2009, Toronto, Ontario, Canada. pp. 358–360 (2009)
11. Cloud Security Alliance: Privacy Level Agreement [V2]: A compliance tool for providing cloud services in the european union (2013)
12. Dupressoir, F., Gordon, A.D., Jürjens, J., Naumann, D.A.: Guiding a general-purpose C verifier to prove cryptographic protocols. Journal of Computer Security 22(5), 823–866 (2014), http://dx.doi.org/10.3233/JCS-140508
13. Ghazinour, K., Majedi, M., Barker, K.: A lattice-based privacy aware access control model. In: Computational Science and Engineering, 2009. CSE '09. International Conference on. vol. 3, pp. 154–159 (Aug 2009)
14. Gürses, S., Gonzalez Troncoso, C., Diaz, C.: Engineering privacy by design. Computers, Privacy & Data Protection (2011)
15. Hafiz, M.: A pattern language for developing privacy enhancing technologies. Softw., Pract. Exper. 43(7), 769–787 (2013)
16. Hoepman, J.H.: Privacy Design Strategies, pp. 446–459. Springer Berlin Heidelberg (2014)
17. Jin, X., Sandhu, R.S., Krishnan, R.: RABAC: role-centric attribute-based access control. In: International Conference on Mathematical Methods, Models and Architectures for Computer Network Security (MMM-ACNS). pp. 84–96 (2012)
18. Jürjens, J.: Secure information flow for concurrent processes. In: Palamidessi, C. (ed.) CONCUR 2000 (11th International Conference on Concurrency Theory). vol. 1877, pp. 395–409 (2000), http://www.jurjens.de/jan

19. Jürjens, J.: Modelling audit security for smart-card payment schemes with UMLsec. In: Dupuy, M., Paradinas, P. (eds.) Trusted Information: The New Decade Challenge. pp. 93–108 (2001), `http://www.jurjens.de/jan`, proceedings of the *16th International Conference on Information Security (SEC 2001)*
20. Jürjens, J.: Model-based security engineering with UML. In: Aldini, A., Gorrieri, R., Martinelli, F. (eds.) Foundations of Security Analysis and Desing III: FOSAD 2004/2005 Tutorial Lectures, Lecture Notes in Computer Science, vol. 3655, pp. 42–77 (2005)
21. Jürjens, J.: Secure systems development with UML. Springer (2005)
22. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Addressing privacy requirements in system design: the pris method. Requirements Engineering 13(3), 241–255 (2008)
23. Kerschbaum, F.: Privacy-Preserving Computation, pp. 41–54. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
24. Object Management Group (OMG): UML 2.5 Superstructure Specification (2011)
25. Pearson, S.: Taking account of privacy when designing cloud computing services. In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. pp. 44–52. CLOUD '09, IEEE Computer Society (2009)
26. Pearson, S., Allison, D.: A model-based privacy compliance checker. IJEBR 5(2), 63–83 (2009)
27. Schneider, K., Knauss, E., Houmb, S., Islam, S., Jürjens, J.: Enhancing security requirements engineering by organisational learning. Requirements Engineering Journal (REJ) 17(1), 35–56 (2012)
28. Spiekermann, S.: The challenges of privacy by design. Commun. ACM 55(7), 38–40 (Jul 2012)
29. Spiekermann, S., Cranor, L.F.: Engineering privacy. IEEE Trans. Softw. Eng. 35(1), 67–82 (Jan 2009)
30. van Staden, W., Olivier, M.S.: Using Purpose Lattices to Facilitate Customisation of Privacy Agreements, pp. 201–209. Springer Berlin Heidelberg (2007)