Systematic Identification of Information Flows from Requirements to support Privacy Impact Assessments

Rene Meis and Maritta Heisel
paluno - The Ruhr Institute for Software Technology – University of Duisburg-Essen, Germany
{firstname.lastname}@uni-due.de

Keywords: Privacy Impact Assessment, Privacy Analysis, Problem Frames, Requirements Engineering

Abstract: Several countries prescribe or advise government departments and organizations to perform a privacy impact assessment (PIA) if these prepare new projects or change existing ones that involve personal information. A PIA shall summarize what personal information is collected, processed, stored, and distributed in the context of the project. But there is only little support for undertaking a PIA and to create a PIA report, most countries only provide vague guidelines and simple templates. We present in this paper an extension of the problem-based privacy analysis (ProPAn) method that derives information needed to conduct a PIA from a requirements model in problem frame notation. We provide a formally specified method with well-defined steps and tool support to reduce the effort to be spent for eliciting the needed information and to ensure that the needed information is as complete and coherent as possible to form an adequate basis for the creation of a PIA report.

1 INTRODUCTION

To provide privacy-aware software systems, it is crucial to consider privacy from the very beginning of the development. Ann Cavoukian was one of the first who promoted this idea with her concept of privacy by design (Cavoukian, 2011). Several countries prescribe or advise government departments and organizations to perform a so called privacy impact assessment (PIA). Wright et al. (Wright et al., 2011) define a PIA as follows: *"A privacy impact assessment is a methodology for assessing the impacts on privacy of a project, policy, programme, service, product or other initiative which involves the processing of personal information and, in consultation with stakeholders, for taking remedial actions as necessary in order to avoid or minimise negative impacts."* In the same document the authors review the PIA methods of seven countries, namely Australia, Canada, Hong Kong, Ireland, New Zealand, the United Kingdom, and the United States for the EU project PIAF[1]. This project had the goal to provide recommendations on how a regulation for a PIA in the EU should look like. In the draft of the EU data protection regulation (European Commission, 2012) in article 33, the EU describes a procedure similar to a PIA called data protection impact assessment.

In this paper, we extend the problem-based privacy analysis (ProPAn) method (Beckers et al., 2014) and show how this extension helps requirements engineers to elicit the information they have to provide for the conduction of a PIA. Wright et al. distilled from their above mentioned analysis of the PIA practice 36 points that they *"recommend for a European PIA policy and methodology"*. These points consist of 15 recommendations on how a PIA guideline document should look like, 9 points address how PIA should be integrated into policy, for the PIA report they give 6 recommendations and also 6 for the PIA process. Requirements engineers can provide valuable input for some of those points on the basis of a requirements model of the software project for which the PIA shall be conducted. Our proposed method addresses the following points which are central for the success of a PIA:

1. *"A PIA should be started early, so that it can evolve with and help shape the project, so that privacy is* built in *rather than* bolted on." Our method starts at the very beginning of the software development process, namely in the analysis phase, and only needs the initial system description consisting of the functional requirements on the system.

2. *"The PIA should identify information flows, i.e., who collects information, what information do*

[1] http://www.piaf.eu

they collect, why do they collect it, how is the information processed and by whom and where, how is the information stored and secured, who has access to it, with whom is the information shared, under what conditions and safeguards, etc.,"

3. *"The focus of a PIA report should be on the needs and rights of individuals whose personal information is collected, used or disclosed. The proponent of the proposal is responsible for privacy The proponent must "own" problems and devise appropriate responses in the design and planning phases."* With the proposed extension of ProPAn, we provide a systematic approach to identify the individuals whose personal information is collected, how it is used by the software system, and to whom it is disclosed on the basis of a given requirements model.

The rest of the paper is structured as follows. Section 2 introduces an eHealth scenario that we use to illustrate our method. The problem frames approach and ProPAn are presented in Section 3 as background of this paper. Our method is then described in Section 4. Section 5 discusses related work, and Section 6 concludes the paper.

2 RUNNING EXAMPLE

We use a subsystem of an electronic health system (EHS) scenario provided by the industrial partners of the EU project *Network of Excellence (NoE) on Engineering Secure Future Internet Software Services and Systems (NESSoS)*[2] to illustrate our method. This scenario is based on the German health care system which uses health insurance schemes for the accounting of treatments.

The EHS is the software to be built. It has to manage electronic health records (EHR) which are created and modified by doctors (functional requirement R1). Additionally, the EHS shall support doctors to perform the accounting of treatments patients received. The accounting is based on the treatments stored in the health records. Using an insurance application it is possible to perform the accounting with the respective insurance company of the patient. If the insurance company only partially covers the treatment a patient received, the EHS shall create an invoice (R2). The billing is then handled by a financial application (R3). Furthermore, mobile devices shall be supported by the EHS to send instructions and alarms to patients (R4) and to record vital signs of patients (R5). Finally, the EHS shall provide anonymized medical data to researchers for clinical research (R6).

3 BACKGROUND

Problem frames are a requirements engineering approach proposed by Jackson (Jackson, 2001). The problem of developing the software-to-be-built (called *machine*) is decomposed until subproblems are reached which fit to problem frames. Problem frames are patterns for frequently occurring problems. An instantiated problem frame is represented as a problem diagram. A problem diagram visualizes the relation of a requirement to the environment of the machine and how the machine can influence these domains. The environment of the machine is structured into domains. Jackson distinguishes the domain types causal domains that comply with some physical laws, lexical domains that are data representations, and biddable domains that are usually people. A requirement can refer to and constrain phenomena of domains. Phenomena are events, commands, states, information, and the like. Both relations are expressed by dependencies from the requirement to the respective domain annotated with the referred to or constrained phenomena. Connections (associations) between domains describe the phenomena they share. Both domains can observe the shared phenomena, but only one domain has the control over a phenomenon (denoted by a "!").

We use the UML4PF-framework (Côté et al., 2011) to create problem frame models as UML class diagrams. All diagrams are stored in *one* global UML model. Hence, we can perform analyses and consistency checks over multiple diagrams and artifacts. The problem diagram (in UML notation) for the functional requirements R5 is shown in Figure 1. The problem diagram is about the problem to build the submachine *Record* that records the vital signs of *Patient*s sent to it via *MobileDevice*s in the corresponding *EHR*s. The functional requirement *R5* refers to the patient from whom the vital signs are recorded and to the mobile device which forwards the vital signs, and the requirement constrains the EHR to store the recorded vital signs in the corresponding health record of the patient.

ProPAn (Beckers et al., 2014) extends the UML4PF-framework with a UML profile for privacy requirements and a reasoning technique. A privacy requirement in ProPAn consists of a *stakeholder* and a *counterstakeholder*, both of which are domains of the requirements model. It states that the counterstakeholder shall not be able to obtain personal in-

[2] `http://www.nessos-project.eu/`

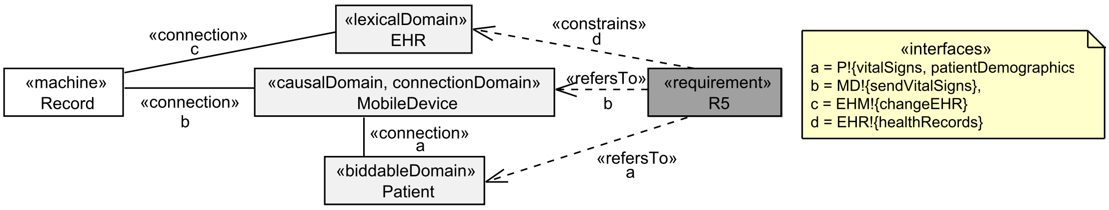

Figure 1: Problem diagram for functional requirement R5

formation of the stakeholder using the system-to-be. Note that *stakeholder* and *counterstakeholder* can be the same biddable domain because biddable domains in the problem frame model do not necessarily represent individuals, but in most cases user roles. Hence, the privacy of an individual can be threatened by another individual of the same user role.The reasoning technique identifies to which domains personal information of the *stakeholder* can potentially flow and to which domains the *counterstakeholder* may have access. For each privacy requirement, the information flows starting from the stakeholder and the access capabilities of the counterstakeholder is visualized in a privacy threat graph. This directed graph has domains as nodes and contains two kinds of edges annotated with statements (requirements, facts and assumptions) describing the origin of the edge. Information flow edges indicate a possible flow of information between the domains and access edges indicate that a domain is able to access information of the other domain. In this paper, we refine these graphs and investigate which personal information really flows between the domains due to the given requirements model.

4 METHOD

Our proposed method is visualized in Figure 2 as UML2 activity diagram. The starting point of our method is a set of functional requirements in form of a UML-based problem frame model. Using this model, we first elicit further context information in the step *Context Elicitation*. The result of this step is *Domain Knowledge* that is integrated into the UML model. Then we can automatically generate *Detailed Stakeholder Information Flow Graphs* from the model and use these in the following step to identify the personal data that is put into the system by stakeholders. The result of this step is the *Personal Data of Stakeholders* and the relations between this data. In the following step, we iteratively analyze the flow of the previously identified personal data through the system using the graphs. During this step, we obtain information about the availability and linkability of personal data at the domains of the system. Our method shall be carried out by requirements engineers in collaboration with experts in the application domain of the system to be built. We will refer to them using term *user*. The final output of our method summarizes due to which requirements, facts, or assumptions personal data flows through the system and can be used as input to create a PIA report. Our method is formally specified and tool supported[3]. The formal specification is not part of this paper due to space limitations, but available as technical report[4]. We extended the UML4PF profile to provide the basis for our tool support as shown in Figure 3. The stereotypes introduced by the profile are discussed in the description of the method steps where they are firstly used.

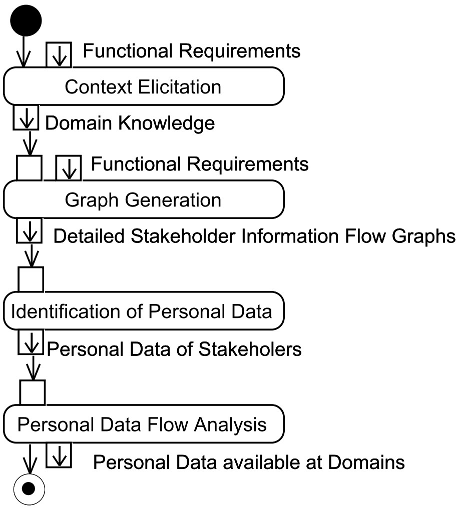
Functional Requirements
Context Elicitation
Domain Knowledge
Functional Requirements
Graph Generation
Detailed Stakeholder Information Flow Graphs
Identification of Personal Data
Personal Data of Stakeholers
Personal Data Flow Analysis
Personal Data available at Domains

Figure 2: Problem-Based Method for the Identification of Privacy-Relevant Information Flows

4.1 Context Elicitation

Information systems often store and process data of persons who not directly interact with systems and that hence may not be represented in the requirements model. Furthermore, there are often informa-

[3] https://www.uni-due.de/swe/propan.shtml

[4] https://www.uni-due.de/imperia/md/content/swe/pia-formal.pdf

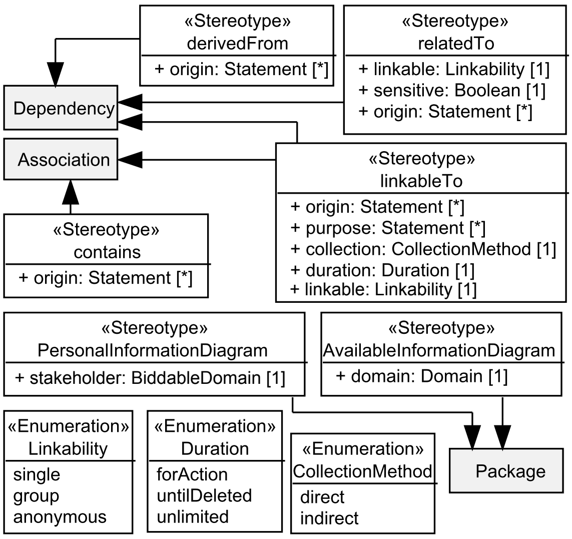

Figure 3: UML profile extension of UML4PF

tion flows between domains in a system that are out of the scope of the functional requirements of the system to be built. E.g., doctors and patients may exchange information without using the system to be built. To elicit these *indirect* stakeholders and *implicit* information flows between domains and stakeholders that are not covered by the requirements, we developed elicitation questionnaires (Meis, 2014). The implicit information flows are captured as domain knowledge diagrams that are generated by the ProPAn-tool based on the user's answers. A domain knowledge diagram is similar to a problem diagram, but it does not contain a machine and instead of a requirement it contains a *fact* (an indicative statement that is always true) or an *assumption* (an indicative statement that is may not true under some circumstances). For our proposed method, it is especially important that during the context elicitation the user elicits the domain knowledge from which domains biddable domains probably gain information. Domains that are part of the same problem diagram as a biddable domain are candidates for domains from which that biddable domain may gain information.

Application to EHS scenario For the sake of simplicity, we only introduce three examples for implicit information flows that we identified for the EHS scenario in (Meis, 2014). First, doctors often act on behalf of patients and enter information into the EHS that they previously got from patients during the treatment (A2). Second, it is possible that the EHS is launched with already existing EHRs (F1). Third, employees using the financial application are able to access the available data necessary for the billing process (A6). The domain knowledge diagram for A2 is shown in Figure 4.

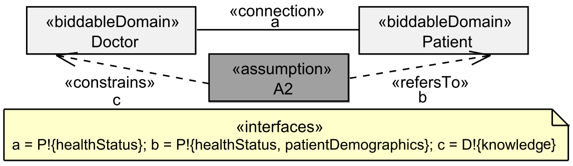

Figure 4: Doctors act on behalf of patients

4.2 Graph Generation

A large set of requirements often implies complex flows of information through the system that are only visible if all requirements are considered simultaneously. Hence, it is a difficult task to analyze these information flows. To assist users to analyze the information flows implied by the given set of requirements, we generate graphs from the problem frame model. In this paper, we introduce so-called detailed stakeholder information flow graphs (DSIFGs) to identify the personal data of the stakeholder and at which domains that information is available due to the functional requirements and the elicited domain knowledge. In a problem frame model, *statements* (requirements, assumptions, and facts) refer to and constrain domains of the machine's environment. If a domain is referred to by a statement, then this implies that it is potentially an information source, and if a domain is constrained, then this implies that based on the information from the referred domains there is a change at the domain. Hence, there is a potential information flow from the referred to domains to the constrained once. Our tool uses this information available in the problem frame model to automatically generate the DSIFG for each biddable domain without user interaction. In contrast to the previously defined graphs (cf. Section 3), a DSIFG has a petri-net like structure with domains as places and statements as transitions. The DSIFG starts with the stakeholder under consideration. Iteratively, all statements that refer to a domain in the DSIFG are added to the DSIFG with input edges annotated with the referred-to phenomena starting from the domain. And for each statement in the graph, the constrained domains are added to the DSIFG with corresponding output edges annotated with the constrained phenomena.

Application to EHS scenario An excerpt of the patient's DSIFG is shown in Figure 5. The patient's DSIFG shows e.g., that assumption A2 (cf. Figure 4) implies an information flow from the patient (referred-to domain) to the doctor (constrained domain) and that requirement R5 (cf. Figure 1) implies information flows from the patient and the mobile device (referred-to domains) to the health records (con-

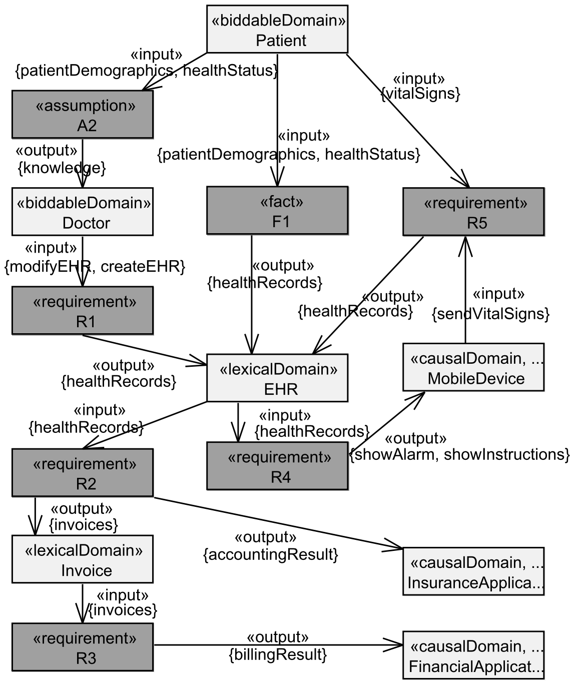

Figure 5: Excerpt of the detailed stakeholder information flow graph for the stakeholder patient

strained domain).

4.3 Identification of Personal Data

For the analysis of the information flow graph, the user has to identify the *personal data* of the stakeholder that is processed in the system under consideration. In the literature, often the term *personally identifiable information (PII)* is used. The International Organization for Standardization (ISO/IEC, 2011) defines PII as *"any information that (a) can be used to identify the PII principal to whom such information relates, or (b) is or might be directly or indirectly linked to a PII principal"*. The European Commission (European Commission, 2012) uses the term *personal data* in the draft of the EU data protection regulation and defines *"'personal data' means any information relating to a data subject*. In this paper, we use the terms *personal data* and *personal information* synonymously as more general terms than PII. Personal data is not only data that can be used to identify an individual or that is linkable to an individual, but also data related to an individual without providing any link to the related individual. E.g., knowing that there is a user with a specific sexual orientation will in most cases not allow one to identify or narrow down the set of users with that specific sexual orientation. But nevertheless, the sexual orientation of a user represents a sensitive personal information that needs special protection if it is processed by the system under consideration. Note that the user of the method can decide to use a more specific definition of personal data, but we decided to use the general term to capture all possibly critical processing of personal data in the system under consideration.

As starting point for the identification of personal data from the requirements model, the user has to look at the data that the stakeholder directly or indirectly provides to the system. This personal data is contained in the phenomena of the stakeholder that are referred to by some statement. Hence, the user has to consider the phenomena annotated at the edges starting from the stakeholder in his/her DSIFG. We distinguish two cases for the identification of personal data in our requirements model. A phenomenon can either be a causal or a symbolic phenomenon. Causal phenomena represent events or commands a domain issues and symbolic phenomenon represent a state, value, or information. If the phenomenon is symbolic, then the user has to check whether this phenomenon represents personal data. If the phenomenon is causal, then the user has to check whether it contains/transmits personal data.

To document the contains/transmits relationship between phenomena, we use aggregations with stereotype ≪contains≫ connecting the phenomena in the UML model (cf. Figures 3 and 6). Besides the property that information is contained in other information, it is often the case that information is not directly contained but derived from other information. This relation is documented as dependency with stereotype ≪derivedFrom≫ (cf. Figure 3) starting from the derived phenomenon and pointing to the phenomena which are necessary to derive it. It is possible that a personal information can be derived from different sources, e.g., the actual position of a person can be derived from the GPS coordinates of the person's smart phone or using the currently available wireless networks also provided by the person's smart phone. In such cases, we add multiple dependencies to the model.

Note that a contains relationship is naturally transitive and that if a phenomenon is derived from a set of phenomena, then each phenomenon of the set can be replaced by a phenomenon that contains it and the phenomenon can also be derived by each superset of the documented set. At the points where we need these properties, our tool computes the transitive closure of these properties. Furthermore, our tool automatically documents for traceability of decisions made, the *origin* of our decision for introducing a contains or derivedFrom relationship. The tool sets the property origin of contains and derivedFrom relations (cf. Figure 3) automatically to the statements from which we identified the relations.

Our tool assists users to identify personal data. The tool presents for a selected stakeholder the phenomena (derived from the DSIFG) that are candidates for personal data of the stakeholder. For each symbolic phenomenon that the user identifies to be personal data, the tool documents the relation to the stakeholder by creating a dependency with stereotype ≪relatedTo≫ starting from the phenomenon and pointing to the stakeholder. To document the relation's quality, the user has to answer two questions:

1. Does the phenomenon represent sensitive personal data for the stakeholder?
2. Does the personal data identify the single individual it belongs to, does it narrow down the set of possible individuals it is related to to a subgroup, or does the information not provide any link to the corresponding individual and is hence anonymous?

The answers to the above questions are stored as properties of ≪relatedTo≫ (cf. Figure 3) and are set manually by the user.

Application to EHS scenario From the DSIFG shown in Figure 5, we derive that *patientDemographics*, *healthStatus*, and *vitalSigns* are the phenomena that have to be considered for patients. All these symbolic phenomena represent sensitive personal information related to a patient. The demographics identify a single individual, whereas the health status and vital signs a group of possible patients. The initially identified relations for the patient are highlighted using bold connections and gray shapes in Figure 6. The other relations visible in Figure 6 are identified during the later iterative analysis.

4.4 Personal Data Flow Analysis

In this step, we analyze how the identified personal data of each stakeholder is propagated through the system based on the given requirements and domain knowledge. As a result of this process, we obtain for each domain and stakeholder of the system a projection of the identified personal data of the stakeholder that is available at the domain.

To document that some personal data about a stakeholder is available at a domain, our tool creates for this domain a package with stereotype ≪availableInformationDiagram≫ in the UML model and adds into this package a dependency with stereotype ≪linkableTo≫ starting from the personal data to the stakeholder when the user identifies this relation during the process. We document as quality attributes of the relation linkableTo to which degree the data available at the domain is linkable to the stakeholder, from which statements of the requirements this relation was derived (origin), for which purpose the information is available at the domain, how the collection of information took place, and how long the information will be available at the domain (duration) using the stereotype properties (cf. Figure 3). Note that we in the first place document for which purpose some personal information is available at a domain due to the requirements model. Whether the stakeholder gave consent to process the data for this purpose and whether the purpose is legitimate as required by some data protection regulations (European Commission, 2012) has to be analyzed later. We distinguish between direct collection from the stakeholder, e.g., the stakeholder enters the information on its own, and indirect collection, e.g., the information is collected by observing the stakeholder's behavior. We distinguish three kinds of duration. If the duration is forAction, then the information will only be available at the domain as long as the information is needed for the action to be performed. If the duration is untilDeleted, then the information will be deleted at some point in time when it is no longer needed, but not directly after it is no longer needed. The duration unlimited expresses that once the information is available at that domain, it will stay available there.

4.4.1 Initialization of Personal Data Flow Analysis

At each domain, the initially available information is the information that the user identified in the previous step for this domain. I.e., the personal data related to the domain itself. The initial available information diagrams are created automatically by our tool. The tool sets the collection method for the initial available information to direct and the duration of availability to unlimited.

During a step of the later iterative personal data flow analysis, the user selects a statement of the DSIFG for which he/she wants to investigate which personal data available at the input domains of the statement flows to which output domain of the statement and in which quality. The tool guides through the process and presents the statements that still have to be considered to the user. Initially, these are the statements for which the stakeholder under consideration is an input domain.

Application to EHS scenario For the stakeholder patient, we have initially to consider the statements $A2$, $F1$, and $R5$ (cf. Figure 5). The information initially available at the patient is the gray part with bold connections in Figure 6.

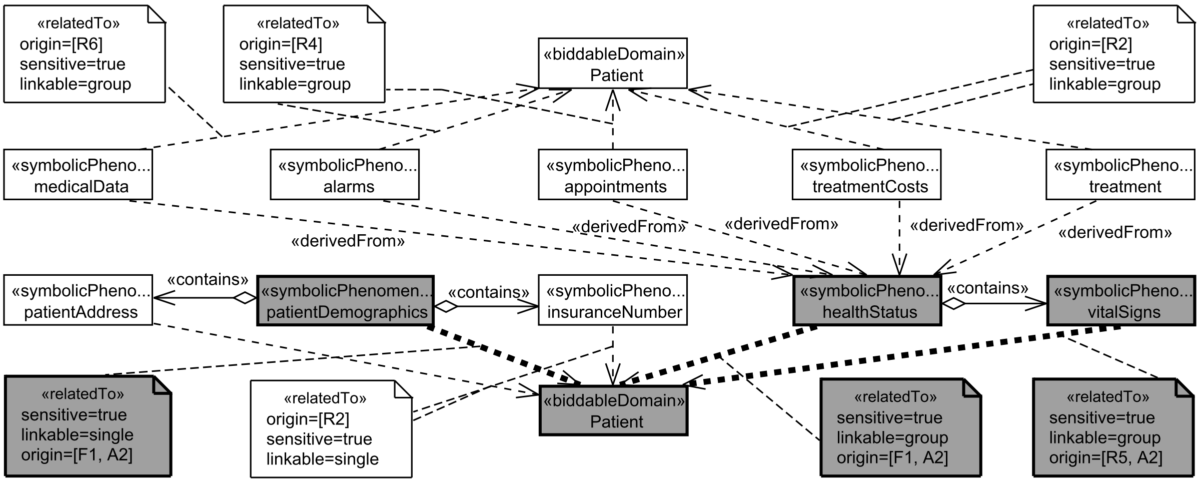

Figure 6: Identified personal information for the patient

4.4.2 Iterative Analysis of the Flow of Personal Data

Now, the user iteratively chooses a statement to be considered for the stakeholder under consideration. Our tool then collects the personal information of the stakeholder that is available at the input domains and computes the transitive closure using the contains and derivedFrom relations.

As mentioned before, the user may identify that only a part of or information derived from the available information is transmitted to output domains. Because of that, the tool asks the user to select available information from which only parts or derived information is transmitted. The user has only to select the available information and to enter the name of the new information. The tool then creates the newly identified phenomenon and the corresponding contains, derivedFrom, and relatedTo relations with the current statement as origin.

Then the user has to decide for each output domain which of the available information is transmitted to it. Based on the user's selection, our tool automatically generates the corresponding model elements. The stereotype properties of ≪linkableTo≫ (besides origin and purpose) have to be adjusted by the user manually. For each transmitted phenomenon, the tool adds the current statement to the property purpose of the ≪linkableTo≫ dependency between the phenomenon and the stakeholder under consideration in an input domain's available information diagram if such a dependency exists. I.e., we document that the information has to be available at the input domain to be transferred to an output domain.

Depending on how the information transfer is described by the current statement, it is possible that an output domain is able to link two data sets related to a stakeholder to each other. I.e., there is information available at the domain that allows everyone who has access to this information to know that different personal data is related to the same individual, but not necessarily to which individual. E.g., the doctor is able to link the health status of a patient to his/her demographics and hence, knows to which patient a health status is related. To document at which domain which information about the stakeholder is linkable, we use an association with stereotype ≪linkableTo≫ (cf. Figure 3) that is part of the package of the domain at which this link is known and connects the phenomena which can be linked. After the user specified the information transmitted to the output domains, the tool asks for each output domain which personal data available at the output domain is linkable to each other and creates on the basis of the user's selection the linkableTo relations. The stereotype properties have to be set by the user manually.

After the above steps, the tool removes the current statement from the set of statements that have to be considered and adds all statements that have one of the current output domains for which the user identified a new information flow as input domain. In this way, the user iteratively traverses the DSIFG supported by the tool until all statements have been considered.

Application to EHS scenario We consider the first step of the analysis with stakeholder patient and statement A2. As input domain, we have the patient and the only output domain is the doctor (cf. Figure 5). The available phenomena are the identified personal data of the patient, namely his/her demographics, vital signs and health status (cf. gray and bold part of Figure 6). We do not identify further contained or

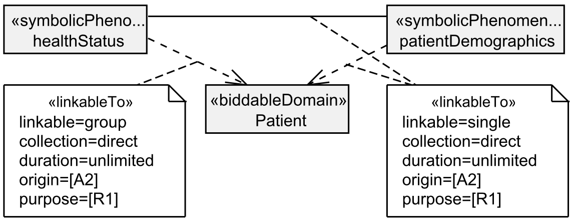

Figure 7: Available information diagram for the doctor

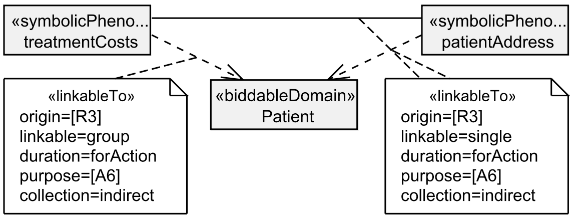

Figure 8: Available information diagram for the financial application

derived personal data in the first step, but we identify that a health status also contains vital signs of the patient. From assumption A2, we see that the doctor gets from his/her patients information about their health status and their demographics. This information is collected directly from the patient. As doctors do not have to delete their records after some time, we set the duration of availability to unlimited. The health status alone is linkable to a group of patients and the demographics to a single patient. Furthermore, the tool adds A2 to the property purpose of the stereotype instances ≪linkableTo≫ in the available information diagram of the patient that start from *patientDemographics* and *healthStatus*. As doctors directly collect the demographics and health status from patients, they are able (and have to be able) to link a health status to a patient's demographics. This information is directly collected from the patient and the duration of availability is unlimited. The linkableTo relations are generated by the tool based on our selections and we adjust the properties of ≪linkableTo≫ manually to the above described values. The resulting available information diagram for the doctor is shown in Figure 7.

During the further analysis, we identify that from the health status of a patient several data is derived (cf. upper part of Figure 6). Alarms and appointments are derived from the health status to be displayed on the mobile devices of patients (R4). We identified that for the accounting the treatments done and the corresponding costs are derived from the health status (R2). For clinical research the health status is anonymized to medical data (R6). Additionally, we identified from R2 that the patient demographics contain the patient's address to be used in the invoices and the insurance number for the accounting (cf. lower left part of Figure 6). Due to limitations of space, we do not show all available information diagrams. Figure 8 shows the personal data of the patient available at the financial application. Only the patient's address and the treatment costs are available at the financial application for the billing process. These two kinds of information are linkable to each other.

4.5 Using the Elicited Knowledge for a PIA Report

The user can now use the collected data to fill parts of a PIA report. The UML model contains:

1. The personal data of stakeholders that is used in the system.
2. The information at which domain of the system which personal data is available and in which quality.
3. Traceability links to identify the requirements, facts, and assumptions that lead to the information flows.
4. For each domain, we can derive the set of counter-stakeholders that possibly have access to personal data available at the domain that they should not be able to access (cf. (Beckers et al., 2014)).

Furthermore, the collected data can be used to start a privacy risk assessment. In the same way, as we elicited the intended information flow implied by the requirements model, we could also document the privacy threats implied by unintended information flows and their probability of occurrence. On the other hand, we could also investigate whether the information available at domains by intention or information that can be derived from that data can lead to privacy threats and how probable these threats are. For each identified personal information, we could additionally elicit the consequences that the disclosure of this information would imply. Based on the probability of privacy threats and the consequences of information disclosure, we could then evaluate the privacy risks implied by the system to be built.

Application to EHS scenario Possible threats to the privacy of patients can be located in the financial application. Employees who are involved in the billing process are able to access patient's addresses and their treatment costs, which are linkable to each other (cf. Figure 8). As the treatment costs are derived from the health status of the patient (cf. Figure 6), employees may gain knowledge about chronic illnesses that patients have if regularly similar treatment costs are recorded.

5 RELATED WORK

Privacy-aware Requirements Engineering

The LINDDUN-framework proposed by Deng et al. (Deng et al., 2011) is an extension of Microsoft's security analysis framework STRIDE (Howard and Lipner, 2006). The basis for the privacy analysis is a data flow diagram (DFD) which is then analyzed on the basis of the high-level threats Linkability, Identifiabilitiy, Non-repudiation, Detectability, information Disclosure, content Unawareness, and policy/consent Noncompliance.

The PriS method introduced by Kalloniatis et al. (Kalloniatis et al., 2008) considers privacy requirements as organizational goals. The impact of the privacy requirements on the other organizational goals and their related business processes is analyzed. The authors use privacy process patterns to suggest a set of privacy enhancing technologies (PETs) to realize the privacy requirements.

Liu et al. (Liu et al., 2003) propose a security and privacy requirements analysis based on the goal and agent-based requirements engineering approach *i** (Yu, 1997). The authors integrate the security and privacy analysis into the elicitation process of *i**. Already elicited actors from *i** are considered as attackers. Additional skills and malicious intent of the attackers are combined with the capabilities and interests of the actors. Then the vulnerabilities implied by the identified attackers and their malicious intentions are investigated in the *i** model.

The above mentioned methods all support the identification of high-level privacy threats or vulnerabilities and the selection of privacy enhancing technologies (PETs) to address the privacy threats or vulnerabilities. These steps are not yet supported by the ProPAn-method. But in contrast to a problem frame model, DFDs, goal models, and business processes, as they are used by the above methods, are too high-level and lack of detailed information that is necessary to identify personal data that is processed by the system and how the personal data flows through the system. Hence, the methods proposed by Deng et al., Kalloniatis et al., and Liu et al. lack of support for the elicitation of the information that is essential for a valuable privacy analysis. Additionally, we provide a tool-supported method to systematically identify the personal data and collect the information at which domains of the system this personal data is available in a way that allows us to use the data to assist PIAs.

Omoronyia et al. (Omoronyia et al., 2013) present an adaptive privacy framework. Formal models are used to describe the behavioral and context models, and user's privacy requirements of the system. The behavioral and context model are then checked against the privacy requirements using model checking techniques. This approach is complementary to ours, because the knowledge collected by our method can be used to set up adequate models, which is crucial to obtain valuable results.

Methodologies supporting PIA

Oetzel and Spiekermann (Oetzel and Spiekermann, 2014) describe a methodology to support the complete PIA process. Their methodology describes which steps have to be performed in which order to perform a PIA. Hence, their methodology covers all necessary steps that have to be performed for a PIA. In contrast to our method, Oetzel and Spiekermann's methodology does not give concrete guidance on how to elicit the relevant information needed for a PIA which is the focus of this work.

Tancock et al. (Tancock et al., 2010) propose a PIA tool for cloud computing that provides guidance for carrying out a PIA for this domain. The information about the system has to be entered manually into the tool. The PIA tool by Tancock et al. covers more parts of a PIA then our method,. In contrast, our method can use the information provided by an existing requirements model and provides in this way more guidance for the elicitation of the information essential for a PIA.

6 CONCLUSIONS

To assist the creation of a PIA report for software projects, we developed a tool-supported method that derives necessary inputs for a PIA from a requirements model in a systematic manner. This method is based on a requirements model in problem frame notation and hence, can be started at the very beginning of the software development process, when it is still possible to influence the software project. Our method assists requirements engineers and domain experts to systematically identify the personal data processed by the system to be built and how and in which quality this personal data flows through the system. This information can then be used to create a PIA report and can also serve as starting point for a privacy risk assessment. Our proposed UML profile can easily be extended with further stereotype properties and values to capture additional information that has to be documented for a specific PIA report.

Our method has some limitations. As starting point of the analysis, we rely on a complete model of functional requirements. Hence, changes in the functional requirements generally imply a re-run of our method and all collected information has to be elicited again. To overcome this limitation, we could enhance our method as follows. If a requirement is removed from the mode, then all information flows that originate from this requirement could be automatically removed from the model by the tool. This is possible due to the attribute *origin* (cf. Figure 3). And if a requirement is added then we would have to check whether this requirement introduces new relevant domain knowledge, and whether the requirement together with the new domain knowledge introduce new information flows to the already elicited information flows. In this way, the already collected information from the unchanged requirements could be kept. Another limitation is that our proposed tool is only a prototype implementation that needs to be further analyzed for usability and user acceptance.

As future work, we want to support the generation of PIA reports based on the elicited information. For this, we will extend our tool support with the possibility to define templates that can be filled with the information contained in the UML model and then be used as part of a PIA report. We also want to extend our proposed method with a privacy risk assessment and to integrate a privacy threshold assessment that indicates which level of detail the PIA shall have. Furthermore, we plan to empirically validate our method, the tool support, and the outputs produced by our method.

REFERENCES

Beckers, K., Faßbender, S., Heisel, M., and Meis, R. (2014). A problem-based approach for computer aided privacy threat identification. In *Privacy Technologies and Policy*, LNCS 8319, pages 1–16. Springer.

Cavoukian, A. (2011). Privacy by design – the 7 foundational principles. `https://www.ipc.on.ca/images/resources/7foundationalprinciples.pdf`.

Côté, I., Hatebur, D., Heisel, M., and Schmidt, H. (2011). UML4PF – a tool for problem-oriented requirements analysis. In *Proc. of RE*, pages 349–350. IEEE Computer Society.

Deng, M., Wuyts, K., Scandariato, R., Preneel, B., and Joosen, W. (2011). A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *RE*.

European Commission (2012). Proposal for a regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation). `http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52012PC0011`.

Howard, M. and Lipner, S. (2006). *The Security Development Lifecycle*. Microsoft Press, Redmond, WA, USA.

ISO/IEC (2011). ISO 29100 Information technology – Security techniques – Privacy Framework.

Jackson, M. (2001). *Problem Frames. Analyzing and structuring software development problems*. Addison-Wesley.

Kalloniatis, C., Kavakli, E., and Gritzalis, S. (2008). Addressing privacy requirements in system design: the PriS method. *RE*, 13:241–255.

Liu, L., Yu, E., and Mylopoulos, J. (2003). Security and privacy requirements analysis within a social setting. In *Requirements Engineering Conf., 2003. Proc.. 11th IEEE Int.*, pages 151–161.

Meis, R. (2014). Problem-based consideration of privacy-relevant domain knowledge. In *Privacy and Identity Management for Emerging Services and Technologies 8th IFIP Int. Summer School Revised Selected Papers*, IFIP AICT 421. Springer.

Oetzel, M. and Spiekermann, S. (2014). A systematic methodology for privacy impact assessments: A design science approach. *European Journal of Information Systems*, 23(2):126–150.

Omoronyia, I., Cavallaro, L., Salehie, M., Pasquale, L., and Nuseibeh, B. (2013). Engineering adaptive privacy: On the role of privacy awareness requirements. In *Proc. of the 2013 Int. Conf. on SE*, ICSE '13, pages 632–641, Piscataway, NJ, USA. IEEE Press.

Tancock, D., Pearson, S., and Charlesworth, A. (2010). A privacy impact assessment tool for cloud computing. In *IEEE 2nd Int. Conf. on Cloud Computing Technology and Science (CloudCom)*, pages 667–676.

Wright, D., Wadhwa, K., Hert, P. D., and Kloza, D. (2011). A privacy impact assessment framework for data protection and privacy rights – Deliverable D1. Technical report, PIAF consortium.

Yu, E. (1997). Towards modeling and reasoning support for early-phase requirements engineering. In *Proc. of the 3rd IEEE Int. Symposium on RE*, pages 226–235, Washington, DC, USA. IEEE Computer Society.