Supporting Privacy Impact Assessments using Problem-based Privacy Analysis

Rene Meis and Maritta Heisel

paluno - The Ruhr Institute for Software Technology,
University of Duisburg-Essen, Duisburg, Germany
{rene.meis,maritta.heisel}@uni-due.de

Abstract. Privacy-aware software development is gaining more and more importance for nearly all information systems that are developed nowadays. As a tool to force organizations and companies to consider privacy properly during the planning and the execution of their projects, some governments advise to perform privacy impact assessments (PIAs). During a PIA, a report has to be created that summarizes the consequence on privacy the project may have and how the organization or company addresses these consequences. As basis for a PIA, it has to be documented which personal data is collected, processed, stored, and shared with others in the context of the project. Obtaining this information is a difficult task that is not yet well supported by existing methods. In this paper, we present a method based on the problem-based privacy analysis (ProPAn) that helps to elicit the needed information for a PIA systematically from a given set of functional requirements. Our tool-supported method shall reduce the effort that has to be spent to elicit the information needed to conduct a PIA in a way that the information is as complete and consistent as possible.

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

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 [2]. Several countries prescribe or advise government departments and organizations to perform a so called privacy impact assessment (PIA). Wright et al. [16] 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 of America 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 [5] 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 [1] and show how this extension helps requirements engineers to elicit the information they have to provide to conduct 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 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*

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

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 [8]. 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 [3] 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 R6 is shown in Fig. 1. The problem diagram is about the problem to build the submachine *Research* that provides medical data extracted from the *EHR*s to the *ResearchDatabaseApplication* based on the requests made by *Researchers* to perform clinical research. The functional requirement *R6* refers to the researcher that requests the medical data and to the health records from which this data

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

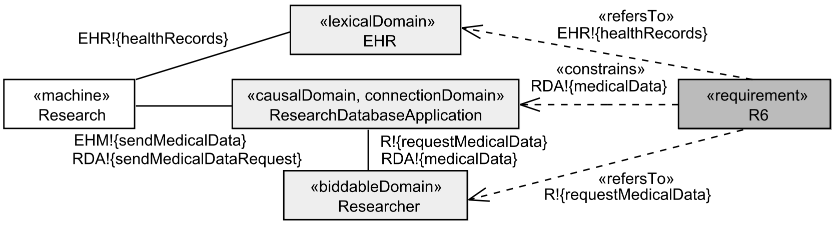

Fig. 1. Problem diagram for functional requirement R6

is extracted. Furthermore, R6 constrains the research database application to provide the requested medical data.

ProPAn [1] 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 are domains of the requirements model. A privacy requirement states that the privacy of the stakeholder shall be preserved against the counterstakeholder in 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 Fig. 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

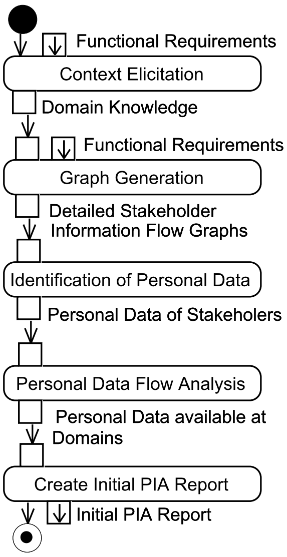
Functional Requirements
Context Elicitation
Domain Knowledge
Functional Requirements
Graph Generation
Detailed Stakeholder Information Flow Graphs
Identification of Personal Data
Personal Data of Stakeholers
Personal Data Flow Analysis
Personal Data available at Domains
Create Initial PIA Report
Initial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

Fig. 2. Overview of the proposed method

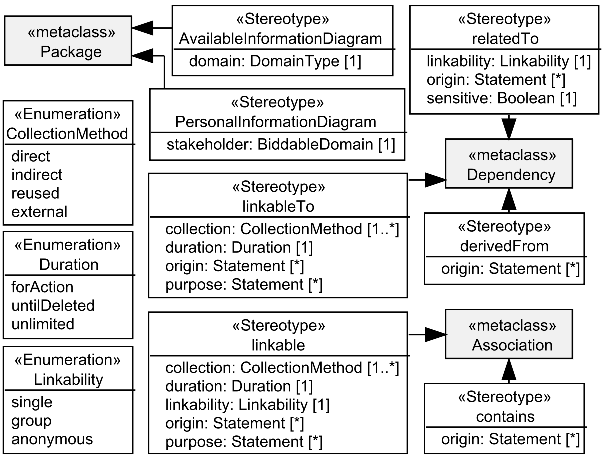

Fig. 3. UML profile extension of UML4PF

previously generated graphs. During this step, we obtain information about the availability and linkability of personal data at the domains of the system. In the last step, we create artifacts for an initial PIA report based on the previously elicited information. Our method shall be carried out by requirements engineers in collaboration with privacy experts and experts in the application domain of the system to be built. We will refer to all of them using the term *user* in the rest of the paper. Our method is supported by the ProPAn-tool[3] that extends the UML4PF-framework [3]. We extended the UML4PF profile to provide the basis for our tool support as shown in Fig. 3. We will explain the stereotypes introduced by the profile where we use them the first time in our method.

4.1 Context Elicitation

Information systems often store and process data of persons who not directly interact with these systems and that hence may not be represented in the requirements model. Furthermore, there are often information 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 [11]. The implicit information flows are captured as domain knowledge diagrams that are generated by

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

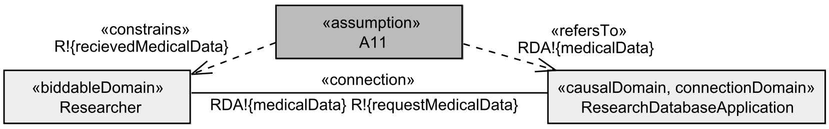

Fig. 4. Researchers receive medical data from the research database application

the ProPAn-tool[4] 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 people (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. The functional requirements usually only refer to the biddable domain involved in it and hence, do not constrain that the biddable domain gains knowledge due to the functional requirement, but this is often the case. Thus, we have to add the missing domain knowledge to the model to document these implicit information flows.

Application to EHS scenario For the sake of simplicity, we only introduce one example for an implicit information flow. For other domain knowledge that we identified for the EHS scenario see [11]. The implicit information flow that we consider in this paper is that researchers get knowledge about the medical data they receive from the research database application based on the requests they make. This information flow is only implicit in the problem diagram for requirement R6 (cf. Fig. 1), because R6 only constrains the research database application to presents the medical data to researchers, but it does not constrain that researchers really receive this information. Figure 4 shows the domain knowledge diagram for assumption A11. It makes explicit that researchers receive the medical data (constrained phenomenon) presented to them by the research database application (referred to phenomenon).

4.2 Graph Generation

A large set of functional requirements and domain knowledge 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

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

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 to domains there is a change at the domain. Hence, there is a potential information flow from the referred to domains to the constrained domains. Our tool uses this information available in the problem frame model to automatically generate the DSIFG for each biddable domain without any user interaction. In contrast to the graphs that are already used in the ProPAn-method (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 together with input edges annotated with the referred-to phenomena starting from the domain to the added statement. And for each statement in the graph, the constrained domains are added to the DSIFG together with corresponding output edges annotated with the constrained phenomena starting from the statement to the added domain.

Application to EHS scenario In this paper, we perform the information flow analysis for the stakeholder doctor. For the analysis of the stakeholder patient, we refer to [12]. An excerpt of the doctor's DSIFG is shown in Fig. 5. The doctor's DSIFG shows how information of the doctor possibly flows through the system based on the functional requirements R1, R2, R3, R4, R5, R6, and the assumption A11. E.g., assumption A11 (cf. Fig. 4) implies an information flow from the research database application (referred to/input domain) to the doctor (constrained/output domain) and requirement R6 (cf. Fig. 1) implies information flows from the health records (EHR) and researchers (referred to/input domains) to the research database application (constrained/output 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 [7] 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 [5] 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 an end-user with a specific sexual orientation will in most cases not allow one to identify or narrow down the set of end-users with that specific sexual orientation. But nevertheless, the sexual orientation of an end-user represents a sensitive personal

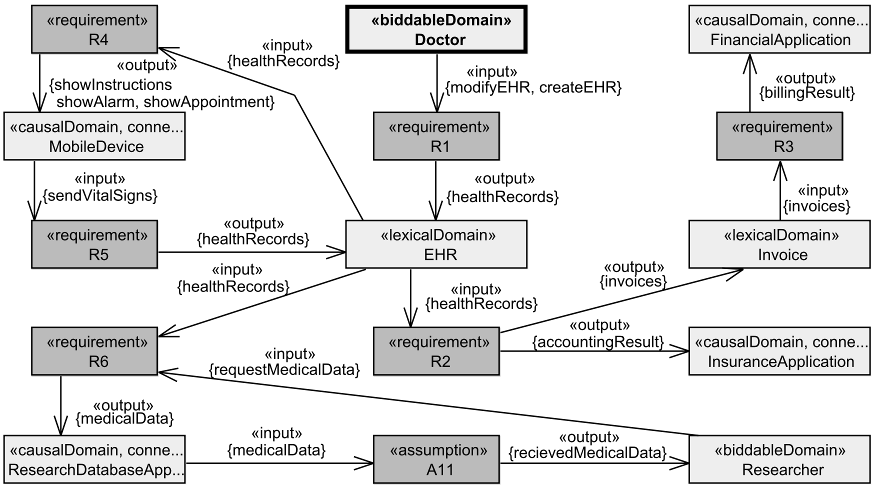

Fig. 5. Excerpt of the doctor's detailed stakeholder information flow graph

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 at least one statement refers to. 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. Fig. 3 and Fig. 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. Fig. 3 and Fig. 6) 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. Fig. 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. Fig. 3) and based on the values the user selects. The property origin is again automatically set by the tool by setting it to the set of statements that refer to the respective phenomenon.

Application to EHS scenario From the DSIFG shown in Fig. 5, we derive that *modifyEHR*, and *createEHR* are the phenomena that have to be considered to identify the personal data of doctors that is processed by the EHS. These phenomena are causal and hence, we have to decide which personal information is contained in them or transmitted by them. We identified that both *modifyEHR* and *createEHR* contain contact information (including name, address, and phone number) of the doctor represented by the symbolic phenomenon *doctorContactInformation*), details about the doctor (e.g. specialization and identification number) represented by *doctorDetails*, the *treatment*s performed by doctors, the *diagnosis* doctors make, and the *notes* doctors make about the progress of the treatment. All these symbolic phenomena represent sensitive personal information related to a doctor. The contact information and the details of the doctor identify a single doctor, whereas the performed treatments, diagnosis, and notes a group of possible doctors. The initially identified relations for the doctor are highlighted using bold connections and gray shapes in Fig. 6. The other relations visible in Fig. 6 are identified during the later iterative analysis.

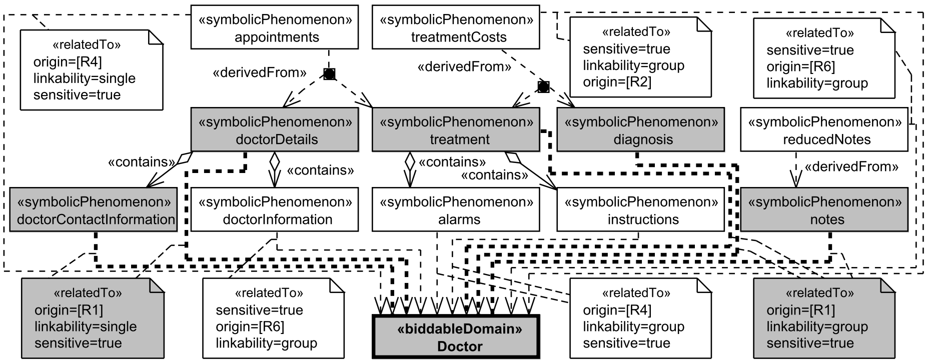
«symbolicPhenomenon» appointments
«symbolicPhenomenon» treatmentCosts
«relatedTo» origin=[R4] linkability=single sensitive=true
«relatedTo» sensitive=true linkability=group origin=[R2]
«relatedTo» sensitive=true origin=[R6] linkability=group
«derivedFrom»
«derivedFrom»
«symbolicPhenomenon» doctorDetails
«symbolicPhenomenon» treatment
«symbolicPhenomenon» diagnosis
«symbolicPhenomenon» reducedNotes
«contains»
«contains»
«contains»
«contains»
«derivedFrom»
«symbolicPhenomenon» doctorContactInformation
«symbolicPhenomenon» doctorInformation
«symbolicPhenomenon» alarms
«symbolicPhenomenon» instructions
«symbolicPhenomenon» notes
«relatedTo» origin=[R1] linkability=single sensitive=true
«relatedTo» sensitive=true origin=[R6] linkability=group
«biddableDomain» Doctor
«relatedTo» origin=[R4] linkability=group sensitive=true
«relatedTo» origin=[R1] linkability=group sensitive=true

Fig. 6. Identified personal information for the doctor

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 enhanced with some additional information.

To document that some personal data about a stakeholder is available at a domain, our tool creates for this domain a package in the UML model with stereotype ≪availableInformationDiagram≫ 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 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. Fig. 3). Note that in the first place, we 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 [5] has to be analyzed later. We distinguish four kinds of collection methods. First, direct collection from the stakeholder, e.g., the stakeholder enters the information on its own. Second, indirect collection, e.g., the information is collected by observing the stakeholder's behavior. Third, reused data that was previously collected (for another purpose). Fourth, data collected by external third parties. Note that we allow to assign multiple collection methods to a linkableTo relation. 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.

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. The attribute origin is set to the value of the corresponding relatedTo relation from which the linkableTo relation is created and the attribute purpose is initially an empty collection.

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 statement R1 (cf. Fig. 5). The information initially available at the patient is the gray part with bold connections in Fig. 6.

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. The computation of the transitive closure can reveal, e.g., that a personal information a that is not available at one of the input domains, but can be derived from two pieces of personal information b available at one input domain and c available at another input domain, possibly flows to the output domain(s) due to the statement under consideration.

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 allows 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 and how long it will be available at the output domain (attribute duration). Based on the user's selection, our tool automatically generates the corresponding model elements. The stereotype property origin is automatically set by the tool to the statement under consideration. For the attribute collectionMethod two cases are distinguished. First, if one of the input domains is the stakeholder, then the user can choose how the information is collected due to the statement. Second, if the stakeholder is not one of the input domains, then the collection method is set automatically to the union of the collection methods specified at the input domains. 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 is available at the input domain for the purpose to be made available at an output domain according to the currently considered statement.

Depending on how the information transfer is described by the current statement, it is possible that an output domain is able to link two pieces of data 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 ≪linkable≫ (cf. Fig. 3) that is part of the available information diagram 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 allows to specify for each output domain which personal data available at the output domain is linkable to each other, to which degree the data is linkable to each other, how this link was collected, and how long this link will be available at the domain. The tool then creates on the basis of the user's selection the linkable relations (cf. Fig. 3) and sets the origin of the linkable relation to the statement under consideration. If such a link already existed at a input domain then the current statement is added to the attribute purpose of this linkable relation, similar to the way we set the purpose of personal data that is available at an input domain.

After the above steps, the tool removes the considered statement from the set of statements that still 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 information flows are documented.

Application to EHS scenario We consider the first step of the analysis for the stakeholder doctor and select statement R1. As input domain, we have the doctor and the only output domain is the EHR (cf. Fig. 5). The available phenomena are the identified personal data of the doctor, namely his/her doctor contact information, doctor details, treatment, diagnosis, and notes (cf. gray and bold part of Fig. 6). We do not identify further contained or derived personal data in the first step, but we identify that the doctor's contact information is contained in the doctor's details. R1 requires that doctors are able to create and modify health records. Doing this they enter and update contact information and details about them, and information about the treatments, diagnoses, and notes they make. All this information is entered directly by the respective doctor. Health

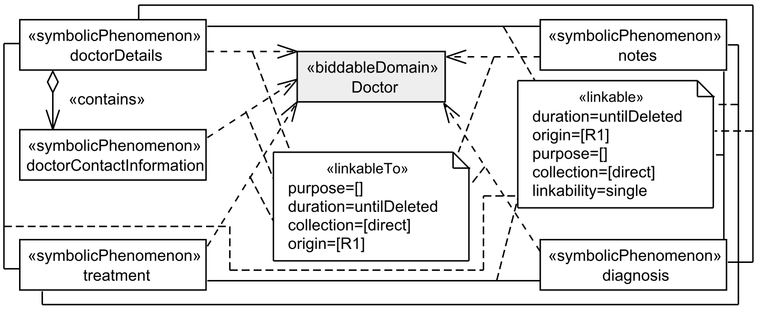

Fig. 7. Available information diagram for the EHR after the first analysis step

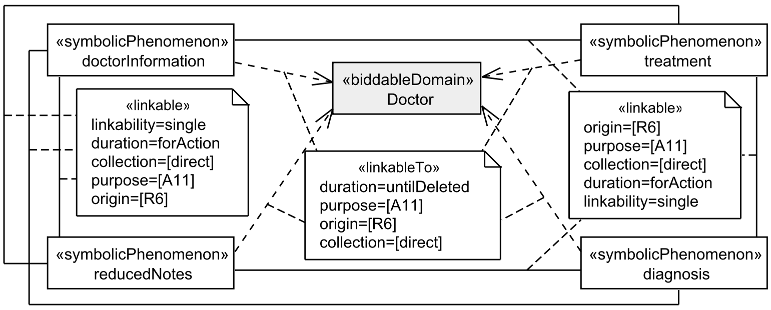

Fig. 8. Available information diagram for the research database application

records do not have to be deleted after the treatment is done, but there can be situations where they have to be deleted after some period. This is because some regulations prescribe to ensure that the health records are kept up to date. If this cannot be assured, e.g., because a patient does not show up for a longer time period, the respective personal data has to be deleted. Hence, we set the duration of availability to untilDeleted. Furthermore, the tool adds R1 to the property purpose of the stereotype instances ≪linkableTo≫ in the available information diagram of the doctor (input domain) for all the personal data that flows to the EHR (output domain). The personal data that doctors enter due to R1 are (and have to be) linkable to each other at the EHR for further processing. E.g., it has to be known to which doctor (determined by doctorDetails) the notes, treatments and diagnoses belong, and also it has to be known which treatments, diagnosis, and notes are related to each other. These links are recorded based on the direct input of doctors, the links allow a 1-to-1 mapping between the personal data (linkability set to single), and these links are kept until they are deleted, analogously to the personal data itself. The generated available information diagram for the EHR after the first analysis step is shown in Fig. 7.

During the further analysis, we identify additional personal information of the doctor that is processed by the EHS. This information is shown in addition to the initially identified personal data of the doctor in Fig. 6. Due to requirement

R4, we identified that the alarms and instructions that are shown to patients using their mobile devices can be considered as a part of the treatment the doctor specifies for a patient and hence this information has also to be considered as personal information. Additionally, we identify from R4 that the appointments of patients are derived from the doctor's details and the specified treatment by the doctors themselves. For the accounting of patients (R2) the costs of the treatments are derived based on the treatments performed and the diagnosis of the doctor. To allow researchers to perform clinical research based on the kept health records (R6), an anonymization of the personal data to be sent to the research database application has to be performed. Only information about the doctor (phenomenon doctorInformation) which is contained in the doctor's details and which does not allow to uniquely identify a single doctor is provided to the research data base application. Additionally, the notes of doctors are reduced to ensure that these notes do not contain any information that reveals the identity of doctors. Due to limitations of space, we do not show all available information diagrams. Figure 8 shows the personal data of the doctor available at the research database application. For clinical research, the diagnoses, treatments, and anonymized information about the doctor and the reduced notes are available at the research database application. All this information is linkable to each other to be of value for clinical research. Furthermore, it was automatically documented by the tool that the purpose for which the personal data and the links between the personal data is available at the research database application is assumption A11 that we identified in the context elicitation step.

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. At this point of the method, 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 counterstakeholders that possibly have access to personal data available at the domain that they should not be able to access (cf. [1]).

Wright et al. [16] propose eleven criteria that indicate the effectiveness of a PIA report. The artifacts on which our method is based and which it produces can be used to address two of them. First, Wright et al. stress to *"include a description of the project to be assessed, its purpose and any relevant contextual information"*. By relying on the problem frames approach, we already have a description of the project to be assessed in the form of a context diagram [8] and the functional requirements. Furthermore, we extend this description in the step *context elicitation* with additional information about the environment of

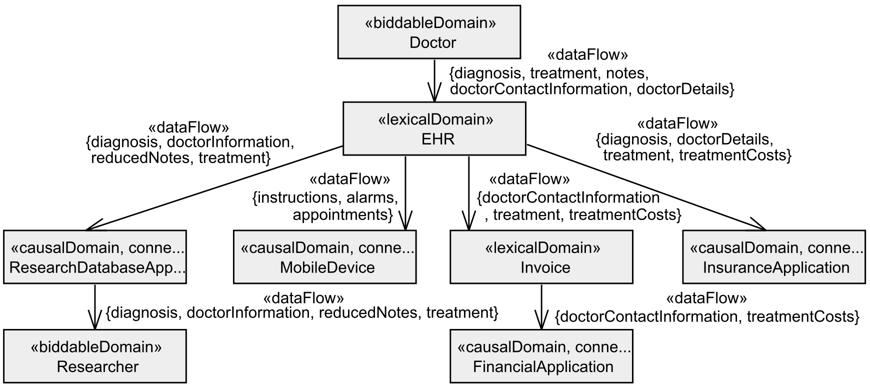
«biddableDomain»
Doctor
«dataFlow»
{diagnosis, treatment, notes, doctorContactInformation, doctorDetails}
«lexicalDomain»
EHR
«dataFlow»
{diagnosis, doctorInformation, reducedNotes, treatment}
«dataFlow»
{instructions, alarms, appointments}
«dataFlow»
{doctorContactInformation, treatment, treatmentCosts}
«dataFlow»
{diagnosis, doctorDetails, treatment, treatmentCosts}
«causalDomain, conne...
ResearchDatabaseApp...
«causalDomain, conne...
MobileDevice
«lexicalDomain»
Invoice
«causalDomain, conne...
InsuranceApplication
«dataFlow»
{diagnosis, doctorInformation, reducedNotes, treatment}
«dataFlow»
{doctorContactInformation, treatmentCosts}
«biddableDomain»
Researcher
«causalDomain, conne...
FinancialApplication

Fig. 9. Stakeholder data flow graph for the doctor

the software-to-be in the form of facts and assumptions. Second, Wright et al. advise to *"map the information flows (i.e., how information is to be collected, used, stored, secured and distributed and to whom and how long the data is to be retained)"*. The information about how the information is collected, used, to whom it is disclosed, and how long the data is retained is elicited during the information flow analysis and documented in the personal and available information diagrams. This information can be used for PIA reports in different ways. Several PIA guidelines suggest to visualize the information flows in the form of an information flow graph (cf. Fig. 9). Our tool is able to automatically generate such graphs (that we call *stakeholder data flow graphs*) automatically on the basis of the available information diagrams. A symbolic phenomenon p flows from a domain i to an other domain o iff p is available at both i and o, and the intersection of the purposes why p is available at i with the statements from which it was identified that p is available at o (origin) is not empty.

Furthermore, we can develop templates to automatically fill the PIA report that has to be created with the information elicited with our method. E.g., we can use the following four templates to document 1. how the information is to be collected, 2. used, 3. to whom it is disclosed and 4. how long the data is to be retained.

1. <*personalInformation*> is <*collection*>ly collected from <*stakeholder*> according to <origin>.
2. <*personalInformation*> is used for <*origin*> by/for the <*Domain*>.
3. <*personalInformation*> is disclosed to <*Domain*>s according to <*origin*>.
4. <*personalInformation*> is retained <*duration*> at the <*Domain*> due to <*origin*>.

The templates are instantiated for a fixed stakeholder and a fixed personal information of that stakeholder, which represent the values of the parameters <*stakeholder*> and <*personalInformation*>. The other parameters are instantiated for specific edges of the stakeholder data flow graph (SDFG). The parameter

<*Domain*> is instantiated with the target domain of the considered edge and the parameters <*collection*>, <*origin*>, and <*duration*> are the corresponding attributes of the linkableTo relation between the personal information and the stakeholder in the available information diagram of the target domain. The first template is instantiated for each edge in the stakeholder data flow graph (SDFG) that starts from the stakeholder and at which the personal information is annotated (collection), the second for each edge that does not start at the stakeholder and does not end at a biddable domain (use of collected data), the third for each edge that does not start at the stakeholder and ends at a biddable domain (flow of data to persons), and the fourth for each edge that does not end at a biddable domain (storage of data).

According to Wright et al., a PIA report shall also contain information about a privacy risk assessment and the proposed measures to reduce the identified privacy risks. Our method does not yet support a privacy risk assessment, but we think that the information our method elicits is a good starting point for the performance of a privacy risk assessment.

Application to EHS scenario The stakeholder data flow graph for the doctor is shown in Fig. 9. It visualizes which personal data (annotated at the edges) flows from which domains to which domains (nodes of the graph). The properties of the personal data are not visualized in the data flow graph, but this information is contained in the corresponding personal and available information diagrams.

If we apply the above mentioned templates and instantiation rules for the stakeholder *Doctor* and the personal information *treatment*, we can generate the following text automatically from the model (cf. Figures 7, 8, and 9) in order to be used for a PIA report. The terms in *italics* represent instantiated parameters of the templates.

1. **Collection**
   Treatment is *direct*ly collected from *Doctor*s according to *R1*.
2. **Use**
   Treatment is used for *R2* by/for the *Invoice*.
   Treatment is used for *R2* by/for the *InsuranceApplication*.
   Treatment is used for *R6* by/for the *ResearchDatabaseApplication*.
3. **To whom**
   Treatment is disclosed to *Researcher*s according to *A11*.
4. **Retention**
   Treatment is retained *untilDeleted* at the *EHR* due to *R1*.
   Treatment is retained *untilDeleted* at the *ResearchDatabaseApplication* due to *R6*.
   Treatment is retained *forAction* at the *Invoice* due to *R2*.
   Treatment is retained *forAction* at the *InsuranceApplication* due to *R2*.

5 Related Work

Privacy-aware Requirements Engineering The LINDDUN-framework proposed by Deng et al. [4] is an extension of Microsoft's security analysis framework STRIDE [6]. 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. [9] 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 implement the privacy requirements.

Liu et al. [10] propose a security and privacy requirements analysis based on the goal and agent-based requirements engineering approach i^* [17]. 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. [14] 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 [13] 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. [15] 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 this personal data flows through the system. We sketched how this information can be used to create parts of a PIA report. Additionally, it can also serve as starting point for a privacy risk assessment. Our proposed UML profile can 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 attributes origin (cf. Fig. 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 further 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

1. Beckers, K., Faßbender, S., Heisel, M., Meis, R.: A problem-based approach for computer aided privacy threat identification. In: Privacy Technologies and Policy.

pp. 1–16. LNCS 8319, Springer (2014)

2. Cavoukian, A.: Privacy by design – the 7 foundational principles (January 2011), https://www.ipc.on.ca/images/resources/7foundationalprinciples.pdf
3. Côté, I., Hatebur, D., Heisel, M., Schmidt, H.: UML4PF – a tool for problem-oriented requirements analysis. In: Proc. of RE. pp. 349–350. IEEE Computer Society (2011)
4. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. RE (2011)
5. European Commission: 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) (January 2012), http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52012PC0011
6. Howard, M., Lipner, S.: The Security Development Lifecycle. Microsoft Press, Redmond, WA, USA (2006)
7. ISO/IEC: ISO 29100 Information technology – Security techniques – Privacy Framework (2011)
8. Jackson, M.: Problem Frames. Analyzing and structuring software development problems. Addison-Wesley (2001)
9. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Addressing privacy requirements in system design: the PriS method. RE 13, 241–255 (August 2008)
10. Liu, L., Yu, E., Mylopoulos, J.: Security and privacy requirements analysis within a social setting. In: Requirements Engineering Conf., 2003. Proc.. 11th IEEE Int. pp. 151–161 (2003)
11. Meis, R.: 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 (2014)
12. Meis, R., Heisel, M.: Systematic identification of information flows from requirements to support privacy impact assessments. In: ICSOFT-PT 2015 - Proc. of the 10th Int. Conf. on Software Paradigm Trends. SciTePress (2015)
13. Oetzel, M., Spiekermann, S.: A systematic methodology for privacy impact assessments: A design science approach. European Journal of Information Systems 23(2), 126–150 (2014)
14. Omoronyia, I., Cavallaro, L., Salehie, M., Pasquale, L., Nuseibeh, B.: Engineering adaptive privacy: On the role of privacy awareness requirements. In: Proc. of the 2013 Int. Conf. on SE. pp. 632–641. ICSE '13, IEEE Press, Piscataway, NJ, USA (2013)
15. Tancock, D., Pearson, S., Charlesworth, A.: A privacy impact assessment tool for cloud computing. In: IEEE 2nd Int. Conf. on Cloud Computing Technology and Science (CloudCom). pp. 667–676 (2010)
16. Wright, D., Wadhwa, K., Hert, P.D., Kloza, D.: A privacy impact assessment framework for data protection and privacy rights – Deliverable D1. Tech. rep., PIAF consortium (2011), http://www.piafproject.eu/ref/PIAF_D1_21_Sept2011Revlogo.pdf
17. Yu, E.: Towards modeling and reasoning support for early-phase requirements engineering. In: Proc. of the 3rd IEEE Int. Symposium on RE. pp. 226–235. IEEE Computer Society, Washington, DC, USA (1997)