Privacy Analysis using Ontologies

Martin Kost
kost@informatik.hu-berlin.de

Johann-Christoph Freytag
freytag@informatik.hu-berlin.de

DBIS Group
Humboldt-Universität zu Berlin
Berlin, Germany

ABSTRACT

As information systems extensively exchange information between participants, privacy concerns may arise from potential misuse. Existing design approaches consider non-technical privacy requirements of different stakeholders during the design and the implementation of a system. However, a technical approach for privacy analysis is largely missing.

This paper introduces a formal approach for technically evaluating an information system with respect to its designed or implemented privacy protection. In particular, we introduce a system model that describes various system aspects such as its information flow. We define the semantics of this system model by using ontologies. Based on the system model together with a given privacy ontology, and given privacy requirements we analyze the modeled system to detect privacy leakages and to calculate privacy indicators. The proposed method provides a *technical* approach to check whether a system conforms to the privacy requirements of the stakeholders or not.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications; I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis

General Terms

Design, Security, Verification

1. INTRODUCTION

Pervasive or ubiquitous computing is rapidly developing. Several services are already available by devices such as locators, routing systems, intelligent travel guides, or personal devices. Concurrently, the resulting pervasive environments offer new opportunities of abuse. Very often, components exchange data, which might raise privacy concerns. Adversaries might combine observed data to extract personal information and to identify the corresponding individuals.

In consequence, data protection authorities [4] call for the use of a Privacy-by-Design (PbD) approach to integrate privacy requirements into the overall design process. A PbD approach must ensure that privacy criteria are considered during all phases of the design and the implementation of a system. Several researchers already contributed towards a better technical support for PbD. Spiekermann and Cranor identify and contrast two approaches: privacy-by-architecture and privacy-by-policy [26]. Gürses et al. single out data minimization as the fundamental principle for PbD [10]. Kargl et al. describe a privacy policy enforcement system based on a protected distributed perimeter [15]. In [18], Kung et al. generally describe a PbD process applied to ICT (Information and Communication Technologies) applications based on the three principles of minimization, enforcement, and transparency. Despite all of these efforts, a comprehensive support for privacy requirements engineering, implementation, and verification is largely missing.

The challenge is therefore to come up with a technical privacy analysis approach which checks if a system conforms to privacy requirements defined by different stakeholders. Up to now, privacy analysis is mostly performed by hand in a process driven manner; e. g., manual review of non-technical specifications such as business processes, or code inspections. It is only applied to systems and applications in an isolated manner instead of performing a global analysis over all interactive systems together. The analysis results are subjective and therefore not comparable. Existing privacy preserving solutions, e. g., pay-as-you-drive insurance [27], are application and domain specific; the applied privacy analysis approaches cannot be reused easily. Today's privacy analysis approaches are not applicable to complex systems because they do not reflect technical details resulting from system design and implementation.

In summary, this paper makes the following contributions: 1.) Introducing a formal/technical approach for analyzing the information flow of a system in order to evaluate how the user's privacy is protected; 2.) Defining and calculating technical indicators for privacy evaluation; 3.) Analyzing privacy risk using technical privacy indicators; 4.) Automatically detecting violations of given (individual) privacy requirements; 5.) Performing privacy analysis in an application independent manner using domain specific semantics in form of ontologies. Using our approach potentially results in ICT systems that better implement the privacy requirements of different stakeholders.

CODASPY'12, February 7–9, 2012, San Antonio, Texas, USA.

The remainder of this paper is structured as follows: Section 2 presents an example scenario of the ITS (Intelligent Transportation Systems) domain to elaborate and describe open issues of existing approaches for privacy analysis. Section 3 introduces our technical approach of privacy analysis. Therefore, this section describes technologies and concepts, which may be used to create privacy requirements and to describe system models that define information necessary for performing the proposed analysis. We discuss expected results of the analysis in form of privacy violations, indicators, and given guarantees. Further, we integrate necessary process steps into the development process of applications. Section 4 explains how to express the required system models and privacy concepts using ontologies. Section 5 discusses how an implementation of our approach may use privacy ontologies for performing the technical privacy analysis of a system. Section 6 discusses related work.

2. OPEN PRIVACY ISSUES IN ITS

In this section, we elaborate and describe open issues of existing design approaches that only integrate a non-technical privacy analysis process. Within the project FP7 PRECIOSA project [3], we investigated several ITS applications such as emergency call, electronic tolling, pay as you drive insurance, online navigation, intersection collision detection, and more to derive common privacy requirements. Thereby, we applied a non-technical and a technical privacy analysis. Exemplary, we apply a non-technical privacy analysis process to a selected ITS scenario and highlight the shortcomings of the analysis result. We perform the following steps to apply a non-technical privacy analysis: First, we describe the overall scenario and its required functionalities. Second, we develop an abstract process description. Finally, we evaluate privacy principles to setup non-technical privacy requirements. Based on the analysis result, we identify open issues that result from the technical realization of the designed ITS scenario.

2.1 ITS Scenario and Process Description

The selected scenario supports ITS safety: Vehicles send beacon messages containing the current vehicle position to Roadside Units (RSUs) enabling services such as traffic monitoring and some safety applications like intersection collision warnings. Data processors, i. e., the RSU and the Traffic Control Center (TCC), process personal information including the current location of every car and therefore of every driver. Thus, (on behalf of the data controller) the application designer or the data protection supervisor has to identify general privacy requirements that are application or domain independent as well as domain specific ones.

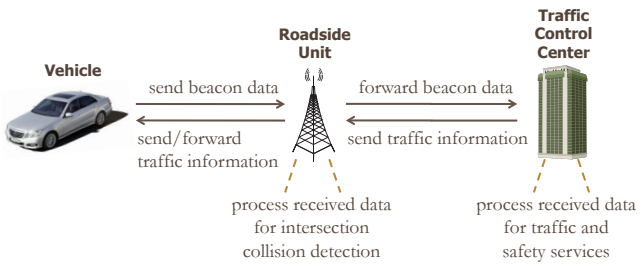

Figure 1: Business Process of ITS Scenario.

Privacy Principle	Resulting Privacy Requirements
1. *Purpose Specification*	**Beacon data** will only be used with the **purpose** of intersection collision detection, traffic monitoring, traffic control, and to **inform about critical events**;
2. *Consent*	The vehicle driver has to **agree** with the use of his/her beacon data for the **specified purposes**; if the driver revokes his agreement the data will no longer be used;
3. *Limited Collection*	**Beacon data** will be **limited** for communication, storage, and collection **to the minimum** necessary for accomplishing the specified purposes;
4. *Limited Use*	RSU and TCC execute **only** those **operations** on the beacon data that are **consistent** with the **purposes** for which the data was collected, stored, and communicated;
5. *Limited Disclosure*	**Beacon data** will **not** be **communicated** outside the system for purposes other than those for which the driver gave his/her consent;
6. *Limited Retention*	Beacon data will be **retained** by the system only **as long as necessary**;
7. *Accuracy & Context Preservation*	Beacon data stored by the system will always be **accurate**, **up-to-date**, and never be decoupled from its **context** and **purpose**;
8. *Security*	Beacon data will be protected by appropriate **security measures** against unauthorized use;
9. *Openness*	The driver will be able to **access all information** that is stored in the system which is **related** to him/her;
10. *Compliance*	The driver will—directly or indirectly—be able to **verify** the **compliance** of the system with the above principles;

Table 1: Application of Privacy Principles.

Given the above-described scenario, the application designer identifies the application requirements and develops an abstract process description (see Figure 1). In our example, the ITS scenario consists of three actors which communicate with each other; these are vehicles, RSUs, and a TCC. The RSU acts as a mediator between vehicles and the TCC. Furthermore, the RSU calculates events of impacting collisions. It requires every vehicle to send its location and a time stamp periodically using a short time interval. The TCC requires additional information such as road condition; e. g., slippery road segments, to monitor the traffic. Monitoring the traffic includes detecting and communicating critical events, or predicting and controlling the traffic flow; e. g., detect emerging traffic jams in order to avoid them.

2.2 Non-technical Privacy Analysis

Application designers evaluate the developed abstract process description of their application in order to identify privacy issues. Non-technical privacy analyses such as *Privacy Impact Assessment* [20] take privacy principles and regulations as input to create a set of questions. The answers form privacy impact assessments that assist managers and decision-makers to avoid or mitigate privacy risks and to determine the best design choice. In Table 1, we exemplary

Technical Aspects	Resulting Privacy Issues
1. **Complexity**: processing and combining plenty of non personal information;	1. May **produce personal identifiable information**;
2. **Adding** (the processing of) technical **information** such as IDs, time, and location;	2. May lead to **tracking** of individuals; e. g., location tracking profiles;
3. **Optimization** of costs and performance leads to consolidation of infrastructure such as data stores and data tables;	3. **Violation of isolation principle** may arise if components implement multiple applications or communicate with other components;
4. **Refining design** decisions which extends the information flow; gathering current vehicle location via mobile phone instead of using smart devices to raise reliability in regions with sparse infrastructure or because of energy consumption limits;	4. **New personal information is processed** to get current location of vehicle—may require other (configuration of) PETs (Privacy Enhancing Technologies); e. g., anonymization;
5. **Caching** of intermediate results; e. g., the mobile phone data, because of performance issues;	5. Raises issue of **limited retention** which could not be detected beforehand;

Table 2: New Privacy Issues resulting from Implementation.

select and evaluate ten privacy principles (based on [5] and [14]) to derive privacy requirements.

Afterwards, the application designer takes the abstract process description and the identified application requirements to design a formal system model. Thereby, he derives technical privacy requirements from the identified non-technical privacy requirements in Table 1. For instance, we may derive the requirement to exchange vehicle ids, e. g., license plates, with dynamic pseudonyms; or we may limit the data which is sent to vehicles as warning messages; for instance, by obfuscating the event location with an obfuscation range of 20 meters.

2.3 Arising technical Privacy Issues

After designing the system model, designers and developers spend several iterations implementing and redesigning the software model. During these iterations they identify new technical requirements, modify existing ones, and implement the modifications. Up to now, these iterations usually do not consider identifying and refining privacy requirements. In Table 2, we give some examples how an implementation may introduce new privacy issues that are impossible or at least difficult to detect beforehand.

Based on our evaluation, we conclude that a technical analysis should provide the following functionalities in order to address the identified privacy issues of Table 2:

- Identifying personal information—to detect privacy issues 1 and 4 of Table 2;
- Identifying personal identifiers—to detect 1;
- Identifying possible combining non-personal information to personal information and operations which create personal information—to detect 1, 2, and 3;
- Identifying operations which operate on personal information—to detect 5;
- Identifying components which perform operations on personal information (potentially collecting these information)—to detect 3 and 5;
- Identifying privacy threats in form of single operations or operation sequences which violate privacy requirements—to detect 2;
- Calculating additional privacy indicators such as values of privacy metrics for processed data—to detect 4;
- Checking a system model for its conformance with privacy requirements defining constraints on its design, deployment, and behavior such as obligations to use specific (configured) components/PETS—to detect 3.

In general, it is difficult to detect privacy-violating processing of personal information because information has several serialization forms. There exist different data type definitions, attribute names etc. for the same information. Information may be combined in different ways to derive new information that might be sensitive or identifying. We may describe location information using different a.) GPS formats, b.) expansions such as postal code, city name, or street name, c.) sights, d.) event bound locations, e.) persons which may be related to locations by different identifying information; for instance, a.) complete name, b.) nickname, c.) social role such as trainer of a football club, d.) relationships among (groups of) people such as living partners or teacher and their students at some school and more. To address these challenges we derive the following requirements for performing a technical privacy analysis:

1. **consider** the **concepts** behind the processed data items—abstract from data formats in the system model;
2. **classify** the **concepts** and **consider their relationships** such as generalization, part-of, equivalence;
3. **create** and **evaluate rules** to derive indirect and complex privacy aspects from described relationships.

In this section, we identified open privacy issues that result from realizing technical aspects of an information system. To address these issues we derived necessary functionalities and requirements that a technical privacy analysis must realize. We need to integrate different technologies in order to implement/support the derived functionalities. In addition, we have to coordinate the use of the functionalities by sophisticated process steps that must be integrated into the development lifecycle.

3. TECHNICAL PRIVACY ANALYSIS

In the previous section, we elaborated essential functionalities and requirements of a technical privacy analysis. Based on these results we propose technologies and concepts for realizing these functionalities and we coordinate necessary process steps. The described approach bases on work carried out within the two research projects FP7 PRECIOSA [3] and DESWAP [2]. First, we describe the general idea of our approach to support the assessment of privacy. Next, we describe methods that we may use to derive privacy requirements. We use these requirements to detect privacy violations by evaluating the system model. In addition we introduce privacy indicators to detect vulnerabilities of the system and to support the transparency of realized privacy

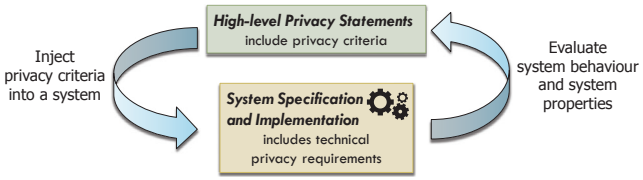

Figure 2: Privacy Assessment Cycle.

protection mechanisms. In order to evaluate a system regarding privacy we model system aspects; e. g., aspects describing the processing of data or the composition of the system. We introduce model elements for describing simple system models that comprise the most relevant aspects. In order to use a technical privacy analysis we must coordinate necessary process steps and integrate them into existing development processes. Further, we discuss the privacy guarantees we may get by applying the formal privacy analysis. At the end of this section, we describe how to define a common privacy vocabulary in order to addresses the requirements identified at end of Section 2.3.

3.1 Supporting Privacy Assessment

Our approach aims to analyze systems formally with respect to the implementation of privacy criteria. Therefore, we realize a *privacy assessment cycle* as shown in Figure 2. This cycle involves two groups of stakeholders: those with an interest in privacy protection, and those responsible for design and implementation. The former stakeholders identify their privacy needs on a non-technical level; e. g., using a PIA process [20]. The resulting privacy requirements include privacy criteria such as user preferences and privacy regulations. The latter stakeholders use these non-technical privacy requirements to translate them into technical privacy statements, to carry out the design and implementation of a system, and to evaluate the resulting system behavior and properties in order to calculate privacy indicators, i. e., evidence that the high-level privacy statements are met.

We suggest using a technical method for evaluating and verifying formally the implemented privacy protection solution with respect to the specified requirements. When applied during system design and system development, this approach could significantly increase the acceptance of the system by users.

3.2 Deriving Privacy Requirements

Our approach is based on the evaluation of specified privacy requirements. In general, it is a challenging task to specify requirements that reflect the interests of the stakeholders appropriately. In existing development processes, the application/system engineers often work together with the stakeholders for setting up and refining non-technical application requirements. Next, the engineers together with domain experts translate the non-technical into technical requirements. Requirements engineering technologies assist engineers for setting up and translating such requirements.

In particular, goal-driven requirements engineering employs goals (enhanced with descriptions of scenarios and purposes) to elicit, specify, analyze, and validate requirements [16]. The authors He and Anton applied this approach to privacy in the area of access control and permissions [12]. While restricted to Role Based Access Control (RBAC), they provide a foundation that can be adapted to other privacy protection mechanisms as well. High-level privacy policies and requirements are expressed in the form of authorization rules. Major concepts to define privacy protection elements are purpose, condition, obligation, and context. Context constraints define restrictions on data purpose and privacy preferences such as the recipient of data or data retention period.

The creation of formal descriptions that define privacy constraints involves further aspects such as failures of a system or vulnerabilities. A goal-oriented approach is proposed by [7] including a risk analysis based on an attack/adversary model. This model is used to identify countermeasures and calculate the probability of the execution of an attack and its success. Attack trees are an established method for modeling security threats [21]. They have already been successfully utilized for the modeling of attacks on inter-vehicle communication systems [6].

Still open is the issue of formulating privacy requirements while addressing the requirements identified in Section 2.3 and to evaluate the effectiveness of applied realizations such as enforced access control policies.

3.3 Detecting Violations of Privacy Requirements

Privacy requirements may define constraints on systems and applications at different points in the development life cycle such as design, implementation, deployment, and runtime. In addition different groups of stakeholders define domain independent, domain specific, or application specific requirements. For instance, privacy principles are general (domain independent) requirements that engineers may consider for every application. As well as domain independent requirements, we must support to express application specific privacy requirements that were derived by refining domain independent requirements.

In our approach, we specify privacy requirements in form of privacy constraints. Privacy constraints are logical expressions that define conditions on properties derived from the system model. By evaluating privacy constraints for a given system model we detect privacy violations. We detect different forms of violations; e. g.:

- unrestricted or unpermitted operations on personal information;
- the violation of individual privacy preferences; e. g., out-of-range-values of privacy metrics or the violation of access control constraints;
- the absence of required security mechanisms such as encryption;
- the violation of privacy principles such as limited retention or data minimalisation, i. e., the identification of unnecessary computation of information which exceeds the assigned purposes.

We give a simplified example of the privacy principle *limited retention* formulated as privacy constraint:

Violation of Limited Retention :=
creation of permanent data D *and*
(no obligation defined for removing D *or*
no remove operation is following)

We formulated the following user defined constraint in form of a privacy policy. The policy defines permissions to

execute operations on selected data of the user. Thereby, the permissions are bound to a defined context that has to match with the context of the request. In our example, the context defines that the data can only be processed on a server node by the traffic state application. The policy permits to query the data and to retrieve the result if certain anonymity is guaranteed. We may get the required anonymity either by formulating a query that produces the required anonymity or by applying an anonymization function on the query result.

```
Context(node-type='server' and
     requestor='TrafficStateApp')
{
  Permit process-query On location, trafficstate,
      vehicle-type As query-result
  Permit retrieve On query-result With
      (k-anonymity > 10)
  Permit TableAnonymization(metric='k-anonymity',
      anonymity-value='10') On query-result
      Retrieve=true
}
```

3.4 Defining and Evaluating Indicators

Privacy indicators describe privacy aspects of the system such as identified personal information, combined personal information, operations on personal information, and components that perform such operations. Additionally, privacy indicators may consist of privacy metrics to calculate quantitative values such as degree of anonymity.

Privacy indicators are selected system properties that are domain or application independent. On the detailed/technical level, these indicators may reflect properties directly described by the system model. The selected properties mostly describe aspects of the performed data processing and the composition of the system. We use logic rules and metrics to calculate and to derive new indicators from the system model and other indicators. We further abstract system details to derive indicators that describe general system properties such as the number of performed operations on personal information or the number of components involved in executing a privacy relevant functionality. In this way, we may abstract the indicators up to non-technical indicators describing aspects such as privacy risk or the compliance of privacy principles.

Example of defining an architectural constraint that uses privacy indicators to restrict a distributed storing of personal information:

PI := personal information
NPPI := nodes containing components that process PI
NSPI := nodes containing components that store PI

if sizeof(NPPI) $>$ 1 then
define constraint: NSPI $\leq$ 1

Example of a privacy indicator that describes the compliance of privacy principles:

Violations of Privacy Principles :=
{Violation of Limited Retention,
Violation of Purpose Specification, ...}

Compliance of Privacy Principles :=
forall v *in* Violations of Privacy Principles: v = false;

Example of a simplified privacy indicator that calculates a privacy risk:

$$\text{Privacy Risk} := \frac{1}{\text{Privacy Risk Aspects}} \cdot \Big(\frac{\text{Number of violated Privacy Principles}}{\text{Number of Privacy Principles}} + \frac{\text{NumViolPrcplFunc}}{(\text{NumPerfFunc} * \text{NumPrcpl})} + \frac{\text{Number of Operations on Personal Information}}{\text{Number of Operations}} + \frac{\text{Average of (Anonymity of Processed Data Set}}{\text{Size of Processed Data Set}} + \frac{\text{Average of (Anonymity of Stored Data Set}}{\text{Size of Stored Data Set}} + \frac{\text{Number of Vulnerabilities}}{(\text{NumPerfOpPI} + \text{NumCompExcOpPI})} \Big)$$

where:

NumViolPrcplFunc := Number of violated Privacy Principles of performed Functionalities
NumPerfFunc := Number of performed Functionalities
NumPrcpl := Number of Privacy Principles
NumPerfOpPI := Number of performed Operations on Personal Information
NumCompExcOpPI := Number of Components which execute operations on Personal Information

By evaluating the calculated privacy indicators we may a.) detect and address privacy leakages; e. g., the publication of personalized location information, b.) calculate the privacy risk or privacy implications such as privacy nudges [1], which result from using the system, c.) select and configure appropriate PETs; e. g., an anonymization function to obfuscate location information, and d.) evaluate the effect of integrating PETs regarding given privacy requirements.

3.5 Modeling Data Processing

We model system aspects especially the data processing (as part of the system model) to get the information necessary for performing a technical privacy analysis. Privacy requirements directly or indirectly describe the permitted and unpermitted information flow that we consider as the processing of personal information. In order to evaluate whether a system model violates given privacy requirements the system model has to comprise the data processing performed within the system. Besides detecting privacy violations and privacy threats, we also want to identify vulnerabilities that describe points at which privacy threats may occur. The used data processing model includes the description about a.) *components* which operate on data, b.) *operations* performed by these components, c.) *data* items as input and output of the operations, and d.) the *information flow* of the performed operations. Figure 3 illustrates a graphical representation of the data processing model. We started our investigations with this simple system model that may be extended to consider additional system and privacy aspects.

Besides structural and operational system information we need data classifications to evaluate privacy aspects; e. g., to identify personal information. Ideally, designers together

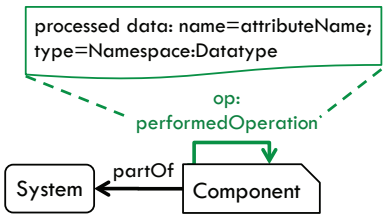

Figure 3: Model of Data Processing—Agenda.

with domain experts annotate the system model by mapping data types with corresponding entries of standardized *data classifications*. Such classifications may be defined using taxonomies or (domain specific) ontologies. Object identification mechanisms as applied to information integration may assist designers to define correct mappings. Alternatively, designers and domain experts may use their own data classification. In consequence they have to prove the suitability of the applied classifications.

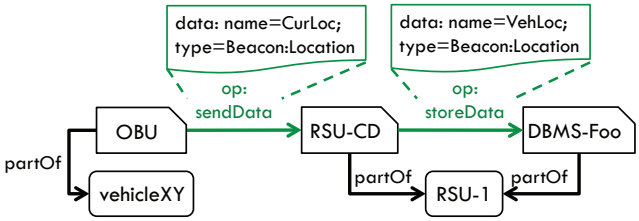

Figure 4: ITS scenario—Model of Data Processing.

The described data processing models may be derived from UML diagrams as part of a system specification, from declarative query, data flow languages, or other system specifications or may be created by hand. With Figure 4, we partially illustrate the data processing model of the ITS scenario described in Section 2. The sample scenario consists of the two systems *vehicleXY* and *RSU-1* which communicate with each other using their embedded components *OBU* (on board unit) and *RSU-CD* (RSU communication device). Additionally, the roadside unit integrates the database management system DBMS-Foo as a component for storing the received beacon data.

3.6 Applying Formal Analysis

Figure 5 illustrates how we support development processes by integrating formal analysis. During the *translation* phase, high level requirements are translated into technical requirements. These requirements are used in the *realization* phase to create a formal system description and identified related *constraints*. The *analysis and verification* phase uses a formal system description to assure that *constraints* are met. In the case of constraint violations a *revision* phase takes place which leads to a modification of the technical requirements or of the formal system description, i. e., a redesign of the system.

The envisioned ontology-based development process includes the following privacy enhancing phases:

1. **Identification:** *Identifying high-level privacy requirements* derived from general privacy principles; e. g., using approaches such as PIA [20]. The resulting requirements are typically described in an informal way. Tools supporting this phase are often limited; e. g., to structured forms.
2. **Translation**: *Mapping the abstract high-level requirements to a detailed formal description of privacy requirements* that can then be related to attributes of a *formal system model*.
3. **Realization:** *Realizing the formal requirements and formally modeling the system*, including its structure and information flows.
4. **Analysis and Verification:** *Matching the formal privacy requirements to the formal system model* to either verify that a given system satisfies the privacy requirements, or to assist a designer in changing the system to meet the privacy requirements. Therefore, the analysis must show at which points privacy requirements are violated and must indicate how to redesign the system structurally or to integrate and to configure existing PETs.

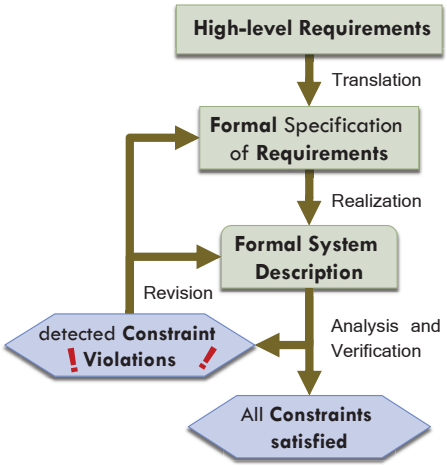

Figure 5: Development Process supporting formal analysis.

3.7 Giving Privacy Guarantees

The goal of applying a technical privacy analysis is to improve the implemented protection of privacy in the analyzed system. In addition to identify vulnerabilities of the system and to calculate indicators which support the transparency of applied privacy protection mechanisms, we want to give privacy guarantees for the analyzed system model. In the following, we describe prerequisites, possible guarantees, and limits of using our approach for giving privacy guarantees. To apply a technical privacy analysis our approach requires 1.) to provide a complete description of necessary system and privacy aspects in form of a system model, 2.) to consider domain specific aspects for creating the system model, privacy requirements, and privacy indicators, 3.) to provide a correct classification of processed data, 4.) to appropriately translate stakeholders privacy criteria into technical privacy requirements. Our suggested technical privacy analysis guarantees 1.) to detect operations on personal information and quasi identifiers, 2.) to localize components which store personal information and

execute operations on personal information, 3.) to detect violations of given privacy principles, 4.) to detect violations of given (domain specific) privacy requirements (of different stakeholders), and 5.) combined with a policy enforcement, we guarantee the realization of user defined privacy preferences. Thereby, the quality of the given guarantees depends on the quality of the used system model and privacy statements. We have to address problems such as ambiguous defined concepts (e. g., synonyms, homonyms, and domain specific concepts), constraints, and relationships as well as misinterpretations. The use of a standard privacy vocabulary improves the acceptance, flexibility, interoperability, and quality of the analysis results.

3.8 Defining a common Privacy Vocabulary

As identified in previous sections – especially in Section 2.3 – we have to address the following requirements: 1.) we require a language to formulate privacy requirements, system models, and privacy indicators; 2.) we must deal with ambiguous concepts (e. g., synonyms, homonyms, and domain specific concepts), constraints, and relationships; 3.) we must abstract the used data formats in the system model to consider the concepts behind the processed data items; 4.) we must classify the concepts and consider relationships between them such as generalization, part-of, equivalence; 5.) we must create and evaluate rules to derive indirect and complex privacy aspects from the described relationships. We use standard ontology languages to define a common privacy vocabulary combined with standard reasoning technologies based on description logic to address these requirements.

Significant work is already available in ontology-based engineering. An overview is provided by [9]. Lee and Gandhi present a framework supporting ontology-based requirements engineering to predict, control, and evolve system behavior [19]. Hartig et al. show how to integrate an ontology-based analysis in a component-based software design process [11].

An ontology-based privacy analysis and verification method is further justified by two specific needs. First, capturing of privacy requirements necessitates the manipulation of a wealth of concepts on privacy, privacy protection, security, storage protection, and others. Second, many constraints related to privacy are domain specific. Therefore, our work includes: a categorization of the different forms of privacy requirements and the presentation of domain specific privacy ontologies. These contributions are further detailed in the following Section 4.

4. DESCRIBING PRIVACY CONCEPTS

We perform privacy analysis on information systems to evaluate the implementation of privacy requirements and to calculate privacy indicators. The result may form the basis for a verification of the analyzed system. Thus, we need a formal and unambiguous description of system models, the technical requirements, and metrics for calculating privacy indicators. As mentioned in Section 3.8 we must base our analysis on well defined (ideally standardized) modeling languages and vocabularies. *Ontologies* provide in part such foundation. The use of ontologies allows us to abstract from implementation issues to identify and to define basic concepts for describing privacy aspects in a domain independent manner. We extend such basic concepts by domain dependent aspects as necessary, and define logic based rules

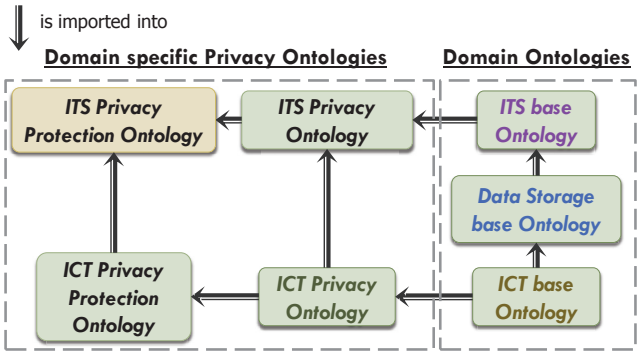

Figure 6: Dependencies between Partial Ontologies.

to derive new system properties and to check for consistency. We implemented our approach using common ontology technology and provide a privacy ontology framework which is available at `http://preciosa.informatik.hu-berlin.de/ontology/`.

4.1 Perspectives and Application Domains

In general, complex systems involve different stakeholders; e. g., users, manufacturers, and legal stakeholders [23, 24] such as data subjects, data processors, data controllers, and legal agencies. Every stakeholder comes with a different background and expectations resulting from their expertise, their cultural background, their interests, and other factors. Regarding privacy, we must identify the relevant domains and model those parts that reflect the concerns and interests of all stakeholders involved.

As an example, we use the domain of ITS to demonstrate how to apply our approach. In the following, we identify the relevant domains for ITS together with privacy related domains and describe their relationships. We describe those by identifying all necessary domain concepts and defining them in the appropriate ontologies. In each domain, we identify only those concepts that are required for privacy analysis. Additionally, we limit our ontologies to those terms that are necessary to define the fundamental concepts in an unambiguous manner. Those might be further refined and extended if necessary. In a second step, we then relate concepts of different domains with the same meaning by (non-automatically) defining mappings between those resulting in a comprehensive ITS Privacy Ontology.

We use the following sources to develop the identified domain ontologies: models of classical authentication and authorization [13], security ontologies [25], the knowledge extracted from specific privacy domains as privacy protection for data storage or communication, and knowledge extracted from legal documents [23, 24].

First, we define basic concepts of the ICT domain and represent them in the *ICT base Ontology*. Subsequently, we relate those terms by defining mappings between terms of different domains with the same meaning to include privacy relevant aspects. Figure 6 illustrates the domain ontologies and their relationships. For example, the *ICT base Ontology* defines fundamental concepts such as *Information*, *Data*, *System*, and others for describing concepts of the ICT domain. The *Policy base Ontology* contains the description of fundamental policy concepts such as *Policy*, *PolicyStatement*, *Context*, *Entity*, *Permission*, *Condition*, and others.

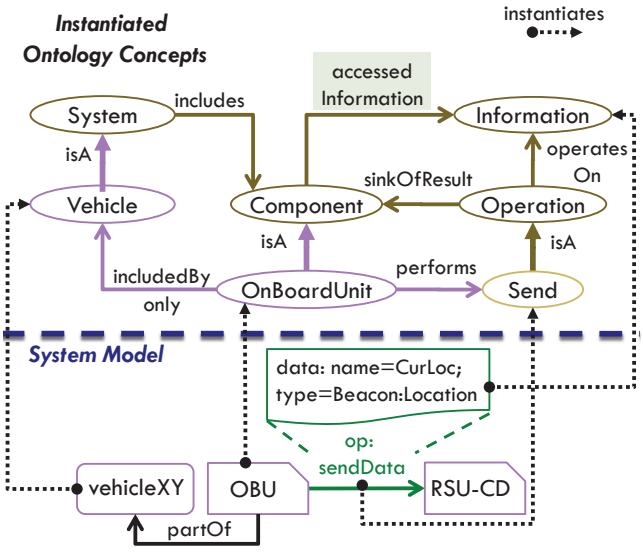

Figure 7: ITS Scenario—System based Description.

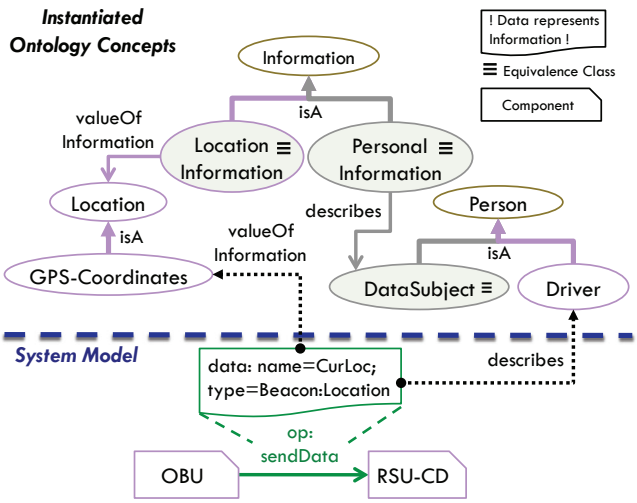

Figure 8: ITS Scenario—Information based Description.

Then, with the *ICT Privacy Ontology* we combine the *ICT base Ontology* and the *Policy base Ontology* and expand the set of definition by some privacy concepts such as *DataController*, *DataProcessor*, *DataSubject*, *PersonalInformation*, and others.

4.2 Base Ontologies

We designed the ontologies to analyze a system regarding different privacy aspects. Therefore, the base ontology (see Figure 10 in the Appendix) defines the semantics of the data processing model which we introduced in Section 3.5. A *Component* as part of a *System* may access *Information* by processing information or creating a result item. We classify and specialize operations (with domain specific ontologies) to evaluate more precisely the effects which result from their execution. Further, we distinguish between Information and *Data* (which represents information) to address the issue that information may have several forms of serialization. We introduce the concept of *ComplexInformation* to model information which is composed of other information. For instance, we introduce address information which is composed of location information such as city, postal code, street, and more. In Section A we describe parts of the ICT Base Ontology, the ICT Privacy Ontology, and the ICT Privacy Protection Ontology to illustrate the definition of concepts and the integration of additional concepts from other domains.

4.3 Defining Indicators and Requirements

The privacy ontologies define privacy indicators as described in Section 3.4 by using logic based rules. A reasoning tool may evaluate those in order to derive new information and to check for consistency.

We define rules—as part of the ontologies—which we evaluate to infer information about the described system model. These rules evaluate system information thus deriving information about privacy relevant aspects. For instance, we characterize information as personal identifiers by evaluating statements which relate information uniquely to individuals (such as the object property *identifies* does). Furthermore, we define equivalence classes and refine concepts such as *Information* and *Operation* by defining hierarchies to classify individuals regarding their privacy relevance. For instance, we define *ssid* as a personal identifier and *address information* as personal identifier as well as personal information. We use equivalence classes and property chains to detect privacy threats; e. g., components which perform operations that operate on information which have been classified as personal or identifying.

4.4 Extending Vocabulary by Domain specific Concepts

The *ICT Privacy Protection Ontology* provides a basic vocabulary for describing information flows and privacy criteria to model the application of privacy protection mechanisms in ICT. For completeness, we also model the data storage domain, the communication domain, and the ITS domain by corresponding ontologies. The *ITS base Ontology* imports concepts from the *ICT base Ontology*. In addition, the ontology includes fundamental concepts of ITS such as *Location*, *Localization*, *LocationTracking*, *Vehicle*, *RSU*, and more. Expanding the base ontologies by ITS concepts leads to a vocabulary which we use to adequately describe system models (especially its processing of information) and (privacy) requirements in the context of ITS. The used modularization and refinement approach may be adapted by other application domains.

4.5 Describing the ITS Scenario

We illustrate the use of privacy ontologies by the ITS scenario which we introduced in Section 2. Figures 7 and 8 partially describe the system model of the scenario as well as the relationships between the model elements and concepts defined in the ITS ontology. In Figure 7 we illustrate how we translate the elements of the described system model (see Figure 4) into instantiations of the ontologies. The illustrated system model consists of the system vehicle *vehicleXY* and the two components: the OnBoardUnit *OBU* which is *partOf* the *vehicleXY* and the communication device *RSU-*

CD. The OBU sends data about the current location to the RSU.

In Figure 8 we illustrate the mapping of the processed data to corresponding ontology concepts. Since the processed data is of type *Beacon:Location* this data represents information that *describes* a specific *Driver*. Furthermore, ITS domain includes a mapping from data of type *Beacon:Location* to the concept *GPX* representing a format for describing location information using *GPS-Coordinates*. Assuming that all drivers are identifiable a *Driver* becomes a *DataSubject* allowing us to infer the following information: 1) All three components process *LocationInformation*; 2) the information processed becomes personal information because the equivalence class *PersonalInformation* comprises *Information* which itself describes a *DataSubject*.

Our ontology framework consists of nine base ontologies, eight domain ontologies (such as ITS, data storage, communication) and four application specific ontologies. Those define about 380 concepts and 150 object properties. We use the Web Ontology Language (OWL) to describe all ontology statements, Protégé to edit and to visualize them, and Pellet to validate and reason about them. The description logic expressivity is SRIQ(D) if we include the definition of the object properties *identifies*, *consistsOf*, *partOf*, and *memberOf* as transitive. Otherwise the expressivity is ALCRIQ(D).

5. PERFORMING PRIVACY ANALYSIS USING ONTOLOGIES

So far, we described the main contribution of the paper in form of an approach for enhancing the software development process with formal privacy analysis by using ontologies for describing privacy related aspects of a predefined system model. In the following, we describe how to apply the ontology based privacy analysis and illustrate its application by the introduced ITS scenario.

5.1 Instantiating Privacy Ontologies

We use reasoning tools for evaluating the system model by inferring about the given axioms and rules. Therefore, we need a description of the system model and given privacy requirements in form of ontology statements. Usually, designers create system models using standard modeling languages such as UML and modeling tools. We have to translate these system models and given privacy requirements into ontology statements. Therefore, we map the required information with instances of ontology concepts. In particular, we map those parts of the system model which we described in Section 3.5 as part of the data processing model. Based on this mapping, we transform the selected elements to instances of the corresponding concepts in the target ontology by using transformation rules (Figure 9). Besides structural and operational system information we must define data classifications to evaluate privacy aspects (e. g., to identify personal information). Therefore, designers annotate the system model by *data classifications* which map data types of the system to corresponding (domain specific) ontology concepts (as described in Section 3.5).

As a result of executing the described transformation we get the required information expressed by instances of the *ICT base Ontology* or domain specific ontologies. These instances now become instances of the privacy extended ontologies such as the *ICT Privacy Ontology* and the domain

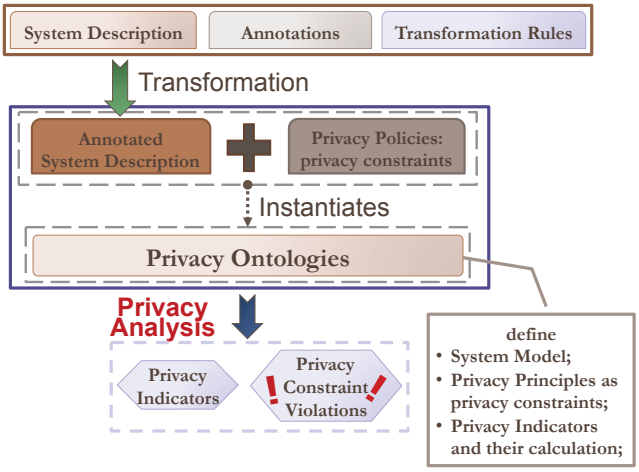

Figure 9: Transform System Model for Ontology based Privacy Analysis.

specific privacy ontologies. We can use these extended ontologies to evaluate privacy specific aspects of the system.

5.2 Analyzing the Annotated System Model

After transforming the annotated system model we use reasoning tools to perform the privacy analysis. Thereby, the reasoner evaluates the ontology statements in form of defined instances, axioms, and rules to infer new information and to check for inconsistencies. Thus, we classify the information regarding its type and privacy aspects by using hierarchies and selected classes such as *PersonalInformation*, we derive new information such as localization of information, check given privacy constraints, and detect vulnerabilities and privacy threats. In this way, we realize the following required functionalities: 1.) evaluating in an implementation independent way the specified a.) information flow and b.) the realization of the given privacy requirements, 2.) calculating privacy indicators which describe a.) detected/identified privacy issues, b.) inferred (new) privacy requirements by evaluating general privacy rules/patterns, c.) values of privacy metrics; e. g., to describe privacy risk, 3.) verifying that all identified privacy issues have been addressed by applying appropriate measures such as PETs or redesign patterns.

The current implementation of our analysis is limited because we use common ontology techniques. The implementation of the analysis does not support probabilistic privacy and does not consider the order and time of operation sequences. Probabilistic privacy is important because privacy has a statistical nature. Traditionally (as in security), the information flow is considered as black and white. We only check if a specific information flow has occurred which we classify as a violation. In privacy analysis, we must consider the probabilities with that an event may occur; e. g., that an adversaries successfully obfuscate information, infer personal information, or relates individuals with sensitive information. Thus, we need to extend formal methods to assure statistical guarantees rather black-and-white correctness guarantees. Probabilistic ontologies may be a good solution to combine our approach with statistical methods. As well as probabilistic privacy, the aspects of time and or-

der introduce a new complexity for evaluating systems. To detect privacy threats effectively we must consider time intervals and the order of operation sequences. Most ontology techniques do not support directly time aspects and the order of sequences.

5.3 Analyzing the ITS Scenario

We address the identified privacy issues of Table 2 in Section 2.3 by evaluating rules which describe possible combinations of ITS data to derive personal information (privacy issues 1 and 2), considering technical information (privacy issues 2, 3, and 5), and considering the system composition (privacy issues 3 and 4).

Applying our privacy analysis approach to the ITS scenario described in Section 2 we detect several privacy constraint violations. One type of violation concerns the privacy principles such as limited retention and limited use. Regarding the store operation of the system model we miss specifications (e. g. in form of a specified remove operation) which realize the limited retention principle. Furthermore, the processing of data is not bound to a purpose which might also trigger a privacy violation rule. To address such privacy violations we may revise the system model by adding specifications limiting the use of the data to a specific context, e. g., defined by the constraints *Purpose = CollisionDetection and SystemType = RSU*, and defining a retention time such as *Retention = 3 Minutes*. Furthermore, we may define individual privacy preferences; e. g., to limit the communication range and to transmit only obfuscated location information. In the same way, we evaluate its realization by the system model.

5.4 Evaluation

In order to determine the effort of using our approach we identified the requirements for performing the proposed technical privacy analysis in Section 3.7. Most of these requirements are the same requirements for performing an obligatory (non-technical) privacy analysis (as described in Section 2.2) or are based on its results. We may invest additional effort to create the required formalization in form of a system model and necessary domain ontologies. If we apply our analysis to another application of the same domain, we may reuse the domain ontologies and the results of the analyses become comparable. Furthermore, we circumvent to analyze systems by hand, which is an error-prone process; e. g., manual review of non-technical specifications, or code inspections. As for non-technical privacy analysis, the quality of the results depends on the quality of the used descriptions in form of the system model and privacy statements. The use of a standard privacy vocabulary improves the acceptance, flexibility, interoperability, and quality of the analysis results. The evaluation of completeness criteria is out of scope of this paper. Currently, a catalog/collection which contains a (in-)formal description of known privacy leakages is not available. Therefore, we miss completeness criteria evaluating our approach.

6. RELATED WORK

While the private impact assessment process [20] provides guidelines on how to elicit high-level privacy statements, no guidance is provided on how to translate those statements into technical requirements. Model checking mechanisms process a model of a system and test automatically whether this model meets a given specification [8]. As most verification techniques, model checking explores all possible system states making it appropriate for infinite state space systems. M. Tschantz and J. Wing provide a comprehensive overview about formal methods to model and evaluate privacy aspects identify challenges concerning models, logics, languages, or tools [28]. In [13], the authors provide a privacy ontology which is based on privacy principles of the legal perspective. The authors aim to support the building and evaluation of privacy-aware applications. To the reader it is not clear how to apply the mentioned ontology for evaluating a real system because the approach does not integrate information about a system model. In [22], the authors suggest to use a privacy ontology to derive the level of privacy for e-commerce applications. The process of describing and evaluating applications regarding privacy is done by designing and evaluating P3P policies in combination with the proposed privacy ontology. The approach is restricted to the e-commerce domain and is not integrated into existing development processes. We propose the missing comprehensive approach to evaluate a system regarding the implementation of stakeholder's privacy criteria.

7. CONCLUSION

In this paper, we describe a new approach using ontologies to evaluate a system model regarding its realization of given privacy requirements. We describe how to integrate and use existing techniques for supporting the privacy assessment of systems. Therefore, we extend the development process of applications and systems by technical privacy analysis and verification. We implemented our approach using common ontology technology and provide a privacy ontology framework[1].

In future work, we plan to apply the technical privacy analysis for selected use cases to evaluate its benefits. To make the evaluation results transparent we require the creation of benchmarks for technical privacy analysis. In addition, we investigate in detail what kind of privacy guarantees we can give for a given set of conditions. We plan to introduce a formalism to prove the correctness of such guarantees. Furthermore, we want to provide tools and guidelines to support the application of the introduced privacy analysis approach. For instance, in order to support designers in creating a formal system description we require declarative (query and policy) languages; statements in these languages express the processing of information and its requirements, respectively. Statements of this language must reference the concepts of the introduced (privacy) ontologies. Therefore, we directly express and analyze the intended information flow and the implementation of the specified privacy requirements resulting in a simplification of the mapping into ontology instances. In addition, components which execute such language statements might monitor and control the intended information processing thereby monitoring and enforcing the specified privacy statements [15, 17].

8. ACKNOWLEDGMENTS

This paper describes results based on work carried out within the FP7 PRECIOSA project [3] for which we acknowledge the support of the European Commission DG INFSO.

[1] `http://preciosa.informatik.hu-berlin.de/ontology/`

9. REFERENCES

[1] Carnegie mellon cylab - project nudging users towards privacy. http://www.cylab.cmu.edu/index.html.

[2] DESWAP (Development Environment for Semantic Web APplications) project., 2007.

[3] PRECIOSA (Privacy Enabled Capability in Co-operative Systems and Safety Applications) FP7 project., 2010.

[4] A. Cavoukian (Information & Privacy Commissioner Ontario, Canada). Privacy-by-design. http://www.privacybydesign.ca/.

[5] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *28th VLB Conference*, Hong Kong, China, 2002.

[6] A. Aijaz, B. Bochow, F. Dötzer, A. Festag, M. Gerlach, R. Kroh, and T. Leinmüller. Attacks on inter-vehicle communication systems - an analysis. In *3rd Int. Workshop on Intelligent Transportation (WIT 2006)*, March 2006.

[7] Y. Asnar, P. Giorgini, and J. Mylopoulos. Goal-driven risk assessment in requirements engineering. *Requirements Engineering*, 2010.

[8] E. M. Clarke and E. A. Emerson. Synthesis of synchronization skeletons for branching time temporal logic. In *In Logic of Programs: Workshop*. Springer-Verlag, 1981.

[9] D.Gasevic, N.Kaviani, and M.Milanovic. Ontologies and software engineering. In S. Staab and R. Studer, editors, *Handbook on Ontologies*. Springer Publishing Company, 2009.

[10] S. F. Gürses, C. Troncoso, and C. Diaz. Engineering privacy by design. In *Computers, Privacy & Data Protection*, 2011.

[11] O. Hartig, M. Kost, and J.-C. Freytag. Automatic component selection with semantic technologies. *Proc.s of the 4th Int. Workshop on Semantic Web Enabled Software Engineering (SWESE) at ISWC*, 2008.

[12] Q. He and A. I. Anton. A framework for modeling privacy requirements in role engineering. *Proc.s of the 9th Int. Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03)*, 2003.

[13] M. Hecker and T. Dillon. Privacy support and evaluation on an ontological basis. In *Proc. of the IEEE 23rd Internat. Conf. on Data Engineering Works.*, Washington, DC, USA, 2007. IEEE Computer Society.

[14] ISO TC 204/SC/WG 1. Intelligent transport systems – system architecture – privacy aspects in its standards and systems. Technical report, ISO, 2008.

[15] F. Kargl, F. Schaub, and S. Dietzel. Mandatory Enforcement of Privacy Policies using Trusted Computing Principles. In *Intelligent Information Privacy Maangement Symposium, AAAI Spring Symposium Series*, Stanford, 2010. AAAI.

[16] E. Kavakli. Goal oriented requirements engineering: a unifying framework. *Requirements Engineering Journal, Springer-Verlag London*, 6, 2002.

[17] M. Kost, B. Wiedersheim, S. Dietzel, F. Schaub, and T. Bachmor. PRECIOSA PeRA: Practical enforcement of privacy policies in intelligent transportation systems. In *Proc. of the Demo. Session at the Fourth ACM Conf. on Wireless Network Security*, 2011.

[18] A. Kung, J.C.Freytag, and F.Kargl. Privacy-by-design in its applications. In *2nd Int. Workshop on Data Security and PrivAcy in wireless Networks*, Lucca, 2011.

[19] S. W. Lee and R. A. Gandhi. Ontology-based active requirements engineering framework. *Asia-Pacific Software Engineering Conf.*, 0, 2005.

[20] I. Linden Consulting. Privacy impact assessment. http://www.ico.gov.uk/for_organisations/data_protection/topic_guides/privacy_impact_assessment.aspx, 2007.

[21] S. Mauw and M. Oostdijk. Foundations of attack trees. In D. Won and S. Kim, editors, *ICISC*, volume 3935 of *Lecture Notes in Computer Science*. Springer, 2005.

[22] E. C. Michael Hecker, Tharam S. Dillon. Privacy ontology support for e-commerce. *IEEE Internet Computing*, 12, 2008.

[23] E. Parliament and of the Council of 24 October 1995. Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, online (access july 31, 2009), 1995.

[24] T. E. PARLIAMENT and T. C. O. T. E. UNION. Directive 2010/40/eu of the european parliament and of the council. *Official Journal of the European Union*, L 207/1, 2010.

[25] M. Schumacher. Security engineering with patterns: Toward a security core ontology. *Springer-Verlag*, LNCS 2754, 2003.

[26] S. Spiekermann and L. Cranor. Privacy engineering. *IEEE Transactions on Software Engineering*, 35(1), January/February 2009.

[27] C. Troncoso, G. Danezis, E. Kosta, and B. Preneel. Pripayd: privacy friendly pay-as-you-drive insurance. In *WPES '07: Proc.s of the 2007 ACM workshop on Privacy in electronic society*, New York, NY, USA, 2007. ACM.

[28] M. Tschantz and J. Wing. Formal methods for privacy. In A. Cavalcanti and D. Dams, editors, *FM 2009: Formal Methods*, volume 5850 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009.

APPENDIX

A. PRIVACY ONTOLOGIES

As described in Section 4.2 we designed the ontologies to analyze a system regarding different privacy aspects. The base ontology (see Figure 10) defines the semantics of the data processing model which we introduced in Section 3.5. A *Component* as part of a *System* (object property *includes*) may access *Information* by processing information (object property *operatesOn*) or creating a result item (object property *creates*). To detect information which was accessed by a component (object property *accessedInformation*) we define two corresponding property chains. We classify and specialize operations (with domain specific ontologies) to evaluate more precisely the effects which result from their execution. Further, we distinguish between Information and *Data* (which represents information) to address the issue that information may have several forms of serialization. We introduce the concept of *ComplexInformation* to model information which is composed of other information. For instance, we introduce address information which is composed of location information such as city, postal code, street, and more.

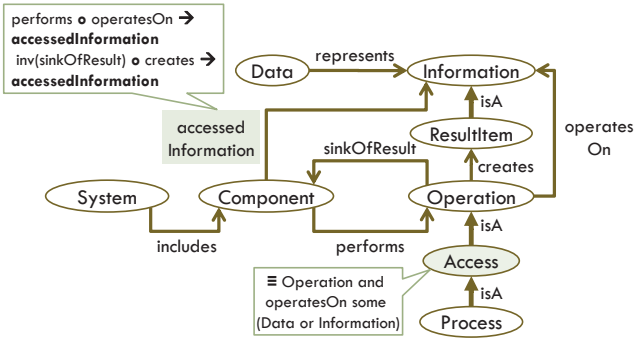

Figure 10: Data Processing Model in ICT ontology.

In Figure 11 we describe parts of the ICT Base Ontology, the ICT Privacy Ontology, and the ICT Privacy Protection Ontology to illustrate the definition of concepts and the integration of additional concepts from other domains. The *ICT base Ontology* defines general concepts such as *Threat*, *Information*, *Identifier*, *Mechanism*, *ProtectionMechanism*, *Component*, and *ProtectionComponent* and their relationships. Based on these definitions, other ontologies define additional concepts, relationships, and axioms. For instance, the *ICT Protection Ontology* defines concepts such as *Privacy Threat*, *Pseudonym*, *Pseudonymization*, *Anonymization*, *AccessControl*, *PrivacyProtectionMechanism*, and *PrivacyProtectionComponent*. In addition, this ontology defines relationships which model the following statements. Privacy protection components implement some privacy protection mechanisms which protect against specific privacy threats. Pseudonymization, anonymization, and access control are privacy protection mechanisms.

For more information about the ontologies and examples for detecting privacy leakages we refer to the homepage (`http://preciosa.informatik.hu-berlin.de/ontology`).

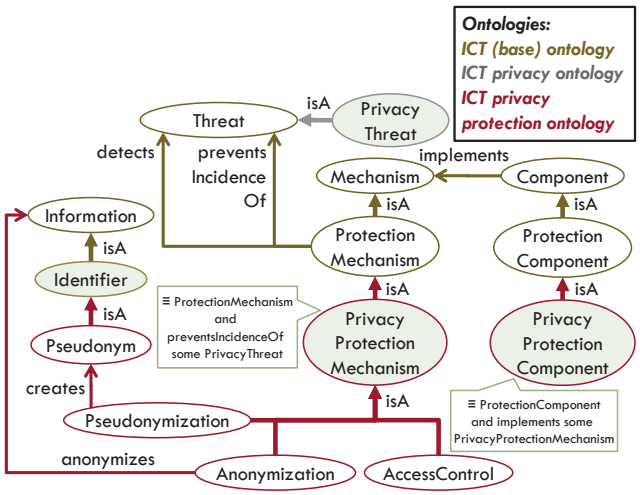

Figure 11: ICT Privacy Protection Ontology.