Privacy Verification using Ontologies

Martin Kost and Johann-Christoph Freytag
Databases and Information Systems
Humboldt University Berlin
Berlin, Germany
{*kost,freytag*}*@informatik.hu-berlin.de*

Frank Kargl
Distributed and Embedded Security
University of Twente
Enschede, Netherlands
f.kargl@utwente.nl

Antonio Kung
Trialog
Paris, France
antonio.kung@trialog.com

***Abstract*—As information systems extensively exchange information between participants, privacy concerns may arise from its potential misuse. A Privacy by Design (PbD) approach considers privacy requirements of different stakeholders during the design and the implementation of a system. Currently, a comprehensive approach for privacy requirement engineering, implementation, and verification is largely missing. This paper extends current design methods by additional (formal) steps which take advantage of ontologies. The proposed extensions result in a *systematic* approach that better protects privacy in future information systems.**

***Keywords*-Privacy Verification; Privacy Ontologies; Privacy Analysis; Privacy Requirements;**

I. Introduction

By increasing the ability to collect and to retain vast amount of data, ICT (Information and Communications Technology) has enabled the advent of applications that have transformed our lives. However, applications such as search engines, social networks, location oriented services, or smart grids have caused a growing concern regarding privacy. To address this concern, data protection authorities [1] call for the use of a Privacy-by-Design (PbD) approach that integrates privacy requirements into the design process right from the beginning.

While informal processes to identify privacy requirements already exist [2], PdB methods still lack a technical perspective. Today, most PbD processes only target identifying high-level privacy requirements that designers and developers are then expected to implement in their systems. The latter is, however, a rather intuitive and error-prone step and resulting privacy protection is hard to evaluate. The resulting implementation is often to complex to be verified by hand. Overall, the situation resembles the state of software engineering of the 80's or 90's.

Today, software engineering is characterized by automation and tools that allow engineers to identify and to manage high-level requirements. With the help of tools and domain experts, designers then translate those high level requirements into technical requirements and formal specifications. Developers then (semi-)automatically build systems. Tools also support to later verify if the specified or implemented system fulfills the technical requirements.

Similar, a PbD approach ensures that privacy criteria of different stakeholders are adequately considered during the different phases of the design and the implementation of a system. Several researchers already have contributed towards a better technical support for PbD. Spiekermann and Cranor identify and contrast two approaches: privacy-by-architecture and privacy-by-policy. The former focuses on data minimization while the latter focuses on enforcing policies in data processing [3]. Gürses et.al. single out data minimization as the foundational principle for PbD [4]. Kargl et.al. describe a privacy policy enforcement system based on a protected distributed perimeter [5]. In [6], Kung et.al. generally describe a PbD process applied to ICT applications based on the three principles minimization, enforcement, and transparency. Despite all of these efforts, a comprehensive support for privacy requirements engineering, implementation, and verification is largely missing which contrasts the state-of-the-art for general (functional) requirements.

This paper contributes towards a formal approach based on ontologies to narrow this gap. Using our approach potentially results in systems that better respect the privacy requirements of different stakeholders in ICT systems.

The rest of the paper is structured as follows. Section II further motivates our general approach of applying formal verification to privacy requirement engineering and discusses related work. Section III explains how to describe privacy concepts by using ontologies. Section IV discusses how ontologies are used for verification during system analysis.

II. General Approach

A. Integrating Privacy into the Design Process

Our approach aims to verify formally systems with respect to the implementation of privacy criteria. Therefore, we realize a *privacy assessment cycle* as shown in Figure 1. This cycle involves two types 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 [2]. The resulting privacy requirements include privacy criteria such as user preferences and privacy regulations. The latter stakeholders use these non-technical

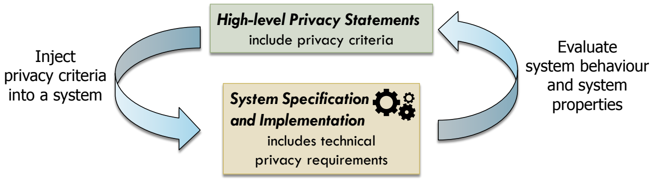
High-level Privacy Statements
include privacy criteria
System Specification and Implementation
includes technical privacy requirements
Inject privacy criteria into a system
Evaluate system behaviour and system properties

Figure 1. Privacy Assessment Cycle

privacy requirements by translating them into technical privacy statements, carrying out a design and implementation of the system, and evaluating 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 the system design, this approach could significantly increase the acceptance of the system by users.

B. *Applying Formal Verification in PbD*

Figure 2 illustrates an abstract design process which integrates formal verification. 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 PbD 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 [2]. The resulting requirements are typically described in an informal way. Tool support for this phase is often limited, for example 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:** *Match the formal privacy requirements to the formal system model* to either verify whether a given system fulfills the privacy requirements (or show where they are violated), or to assist a designer in changing the system to fulfill the privacy requirements; i.e., to redesign structurally the system or to integrate and to configure existing Privacy Enhancing Technologies (PETs).

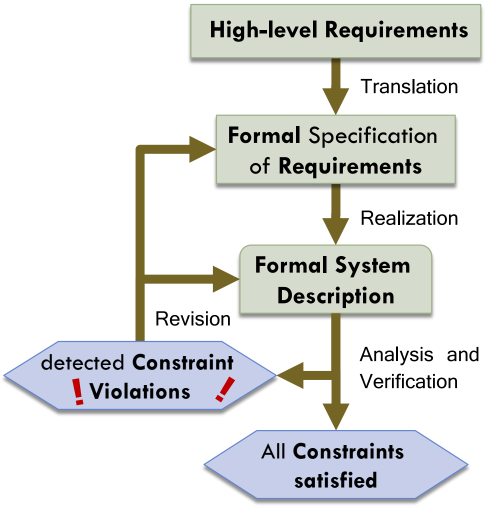
High-level Requirements
Translation
Formal Specification of Requirements
Realization
Formal System Description
Revision
Analysis and Verification
detected Constraint ! Violations !
All Constraints satisfied

Figure 2. Design Process supporting formal analysis

C. *Related Work*

Translation: While the private impact assessment process [2] provides guidelines on how to elicit high-level privacy statements, no guidance is provided on how to translate those statements into technical requirements.

Goal-driven requirements engineering employs goals (enhanced with descriptions of scenarios and purposes) to elicit, specify, analyze, and validate requirements [7]. He and Anton applied this approach to privacy in the area of access control and permissions [8]. 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.

Formalization for Privacy: The creation of a formal description to integrate privacy constraints involves aspects such as failures of a system or vulnerabilities. A goal-oriented approach is proposed in [9] 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 [10]. They have already been successfully utilized for the modeling of attacks on inter-vehicle communication systems [11].

Verification approaches: Model checking mechanisms process a model of a system and test automatically whether this model meets a given specification [12]. 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 [13]. While policy languages such as P3P were defined to automatically enforcement privacy specifications those languages usually lack a formal semantics [14]. Barth et.al. define a formal language using temporal logic that is integrated in a logical framework based on contextual integrity [15]. This framework allows users to describe norms regarding the transmission of personal information (e.g. how it is transmitted). Fu proposes a logic based framework focusing on the privacy protection of web applications [16]. A first order extension of computational tree logic is used to specify a policy. Verification of policies uses a static control/data flow analysis. Métayer proposes a formal framework to deliver the individuals consent regarding the processing of its personal information through software agents [17].

D. Rationale for Ontology-based Verification

Significant work is already available in ontology-based engineering. An overview is provided in [18]. Lee and Gandhi present a framework supporting ontology-based requirements engineering to predict, to control, and to evolve system behaviour [19]. Hartig at. al. show how to integrate an ontology-based analysis in a component-based software design process [20].

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: (1) a categorization of the different forms of privacy requirements and (2) the presentation of domain specific privacy ontologies. These contributions are further detailed in the next sections.

III. Describing Privacy Concepts by Ontologies

We perform privacy analysis on (application) systems and components to evaluate the implementation of privacy requirements/constraints and to calculate privacy indicators. The result forms the basis for a verification of the analyzed system. Thus, we need a formal and unambiguous description of the system model, the technical requirements, and metrics for calculating privacy indicators. Consequently, we must base our analysis on a well defined (ideally standardized) modeling languages and vocabularies. *Ontologies* provide in part such foundation that allow us to abstract from implementation issues to identify and to define basic concepts for describing privacy aspects in a domain independent manner, to extend such basic concepts by domain dependent aspects as necessary, and to define logic based rules to derive new system properties and to check for consistency.

In general, complex systems involve different stakeholders such as users, data subjects, data processors, data controllers, manufacturers, and legal agencies. Every stakeholder comes with a different background and expectations resulting from their expertise, their cultural background, their interests, and possibly more 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 Intelligent Transportation Systems (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 explicitly (non-automatically) defining mappings between those resulting in a comprehensive ITS Privacy Ontology.

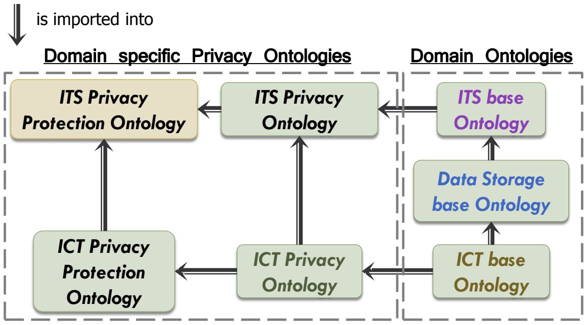
is imported into
Domain specific Privacy Ontologies
Domain Ontologies
ITS Privacy Protection Ontology
ITS Privacy Ontology
ITS base Ontology
Data Storage base Ontology
ICT Privacy Protection Ontology
ICT Privacy Ontology
ICT base Ontology

Figure 3. Dependencies between Partial Ontologies

We use different sources to develop the identified domain ontologies such as models of classical authentication and authorization, security ontologies, the knowledge extracted from specific privacy domains as privacy protection for data storage or communication, and knowledge extracted from legal documents.

As a starting point 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 3 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. 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.

In part, we describe 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 mechanism which protect against specific privacy threats. Pseudonymization, anonymization, and access control are all privacy protection mechanisms.

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.

We illustrate the use of privacy ontologies by a scenario supporting ITS safety. Vehicles send beacon messages containing the current vehicle position to Roadside Units (RSUs) to enable / support services such as traffic monitoring and some safety applications like intersection collision warnings. The data processor, i.e., the RSU, stores and processes personal information including the current location of every car and therefore of every driver. Thus, the application designer (on behalf of the data controller) has to identify general (application or domain independent) and domain specific (high level) privacy requirements. In the next step the designer translates the high level requirements into (formal) technical privacy requirements to provide a basis for a (semi-) automatic analysis by appropriate tools.

Figure 4 partially describes aspects of the system model for this scenario and the relationships between the elements and concepts defined in the ITS ontology. The illustrated

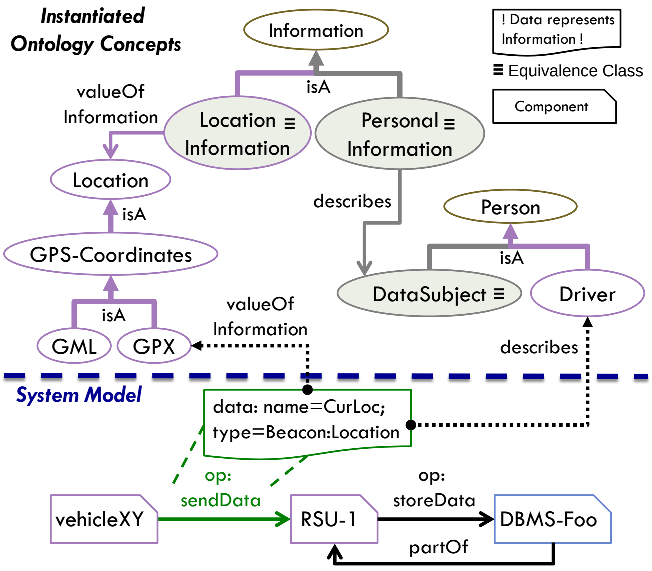

Figure 4. Ontology Description of the ITS Scenario

system model consists of the three components: the vehicle *vehicleXY*, the Road Side Unit (RSU) *RSU-1*, and the Data Base Management System (DBMS) *DBMS-Foo* which is *part of* the *RSU-1*. The vehicle sends data about the current location to the RSU which takes the data and stores it into the DBMS. 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).

IV. Using Ontologies for Privacy Analysis and Verification

Now we are ready to describe how to integrate the ontology based privacy analysis into the proposed PbD process. The main idea is to map selected parts of the system model into instances of ontology concepts (similar to the approach in [20]) to perform formal privacy analysis. In particular, those parts of the system model which describes the processing of information and the composition of components are the focus of our mapping.

The translation phase is followed by the analysis phase which involves: 1.) evaluating in a implementation independent way the specified a) information flow and b) the realization of the specified privacy statements, 2.) calculating privacy indicators which describe a) detected/identified privacy issues, b) inferred (new) privacy statements 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.

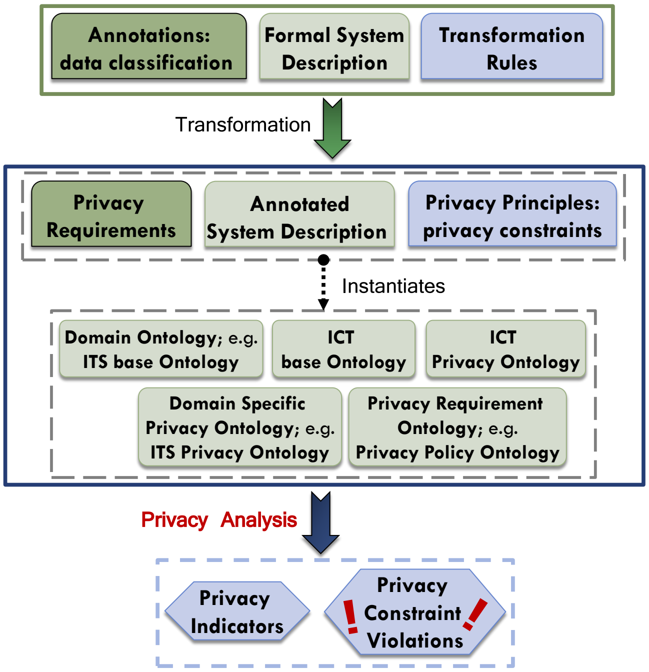
Annotations: data classification
Formal System Description
Transformation Rules
Transformation
Privacy Requirements
Annotated System Description
Privacy Principles: privacy constraints
Instantiates
Domain Ontology; e.g. ITS base Ontology
ICT base Ontology
ICT Privacy Ontology
Domain Specific Privacy Ontology; e.g. ITS Privacy Ontology
Privacy Requirement Ontology; e.g. Privacy Policy Ontology
Privacy Analysis
Privacy Indicators
Privacy Constraint Violations

Figure 5. Translating System Models for Ontology based Privacy Analysis

The translation phase involves the mapping and transformation of elements of the system model to the corresponding concepts in the target ontology (see Figure 5) through transformation rules. As part of the system model we map elements of selected UML diagram types to refer to concepts defined by the *ICT base ontology*; e.g., *UML:component* → *ICT:Component*, *UML:aggregation* → *ICT:includedBy*, *UML:information* → *ICT:Information*. Defined once, designers may reuse these mappings within the same ICT context and possibly adapt them by domain specific concepts; e.g., mapping a send operation to the corresponding concept of the *ITS base Ontology*. Besides structural and operational system information we need 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 to corresponding (domain specific) ontology concepts. Two alternatives exist to create such mappings: Either, we relate data items to standard data types for which we create or reuse mappings to ontology concepts, or we directly map data items to their corresponding ontology concepts.

The result of the overall mapping is 1.) a formal system description with 2.) annotations. Both are expressed by instances of the *ICT base Ontology* or domain specific ontologies. Those instances now become instances of the extended ontologies such as the *ICT Privacy Ontology* and the domain specific privacy ontologies. We can now use these extended ontologies to evaluate privacy specific aspects of the system.

We introduce privacy indicators to describe privacy concerns such as inferred personal information, operations on personal information, and components which perform such operations. Additionally, privacy indicators may consist of privacy metrics to calculate quantitative values such as degree of anonymity. The privacy ontologies define privacy indicators by using logic based rules. Reasoner evaluate those in order to derive new information and to check for consistency. Using the calculated indicators we select appropriate PETs (e.g. an anonymization function to obfuscate location information) for addressing detected privacy leakages (e.g. the publication of personalized location information), for evaluating the protection potential of selected PETss, for calculating the privacy risk (or privacy implications as nudges [21]) when using the system, or for evaluating PETs in the context of specific privacy requirements.

We specify privacy requirements in form of privacy constraints which evaluate privacy indicators. Privacy constraints are used to 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 (the identification of unnecessary computation of information which exceeds specified purposes), and more.

If we apply our privacy analysis to the ITS scenario above, we detect several privacy constraint violations. The first type of violations concerns 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 represent a privacy violation. 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, if we define individual high-level-privacy preferences (e.g. to limit the communication range and to transmit only obfuscated location information), in the same way we could evaluate its realization by the system specification.

In order to support designers in creating a formal system description we introduce declarative (query and policy) languages; statements in these languages express the processing of information and its requirements, respectively. Statements of this language 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 (as already shown in [5]).

V. Conclusion

In this paper we described a PbD process which consistently supports privacy requirements engineering, system design, and formal verification. We examined how to integrate privacy requirements in the form of formal constraints into the design process of a system. In combination with ontologies we provided a formal method which evaluates a system specification regarding its realization of specified privacy constraints. In this paper we leave out technical details; e.g., how we support the PbD process by comprehensive privacy ontologies, developed declarative languages for expressing queries and privacy policies which we can combine with our ontology based analysis, and a query execution component which processes such language statements to control the information flow and to enforce the specified policies.

This paper mostly describes work carried out within the FP7 PRECIOSA project [22] for which we acknowledge the support of the European Commission DG INFSO.

References

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

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

[3] S. Spiekermann and L. Cranor, "Privacy engineering," *IEEE Transact. on Software Engin.*, vol. 35, no. 1, 2009.

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

[5] 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: AAAI, 2010.

[6] 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 (D-SPAN)*, Lucca, 2011.

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

[8] Q. He and A. I. Anton, "A framework for modeling privacy requirements in role engineering," *Proc.s of the 9th Int. Works. on Requirements Engin.: REFSQ*, 2003.

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

[10] S. Mauw and M. Oostdijk, "Foundations of attack trees," in *ICISC*, ser. Lecture Notes in Computer Science, D. Won and S. Kim, Eds., vol. 3935. Springer, 2005.

[11] Amer Aijaz et al., "Attacks on inter-vehicle communication systems - an analysis," in *3rd Int. Workshop on Intelligent Transportation (WIT)*, March 2006.

[12] 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.

[13] M. Tschantz and J. Wing, "Formal methods for privacy," in *Formal Methods*, ser. Lecture Notes in Computer Science, A. Cavalcanti and D. Dams, Eds. Springer, 2009, vol. 5850.

[14] "Platform for Privacy Preferences (P3P) Project," http://www.w3.org/P3P.

[15] Barth, Adam et al., "Privacy and contextual integrity: Framework and applications," in *Proc.s of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2006.

[16] X. Fu, "Conformance verification of privacy policies," in *Proc. of the 7th intern. conf. on Web services and formal methods*, ser. WS-FM'10. Springer, 2011.

[17] D. Métayer, "A formal privacy management framework," in *Formal Aspects in Security and Trust*, P. Degano, J. Guttman, and F. Martinelli, Eds. Springer-Verlag, 2009.

[18] D.Gasevic, N.Kaviani, and M.Milanovic, "Ontologies and software engineering," in *Handbook on Ontologies*, S. Staab and R. Studer, Eds. Springer Publishing Company, 2009.

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

[20] O. Hartig, M. Kost, and J.-C. Freytag, "Automatic component selection with semantic technologies," *Proc.s of the 4th Int. Works. on Semantic Web Enabled Software Engin.*, 2008.

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

[22] "Preciosa (Privacy Enabled Capability in Co-operative Systems and Safety Applications) FP7 project." 2010.