Assisted Generation of Privacy Policies using Textual Patterns

Nazila Gol Mohammadi, Jens Leicht, Ludger Goeke and Maritta Heisel
Paluno - The Ruhr Institute for Software Technology, University of Duisburg, Essen, Germany

Keywords: Data Protection, Pattern-based Approach, Textual Patterns, Privacy Policy.

Abstract: To comply with data protection legislation, privacy policies are a widely used approach and an important legal foundation for data handling. These policies are created by service providers. The creation of a privacy policy for a service is time consuming and compliance with legislation is hard to ensure. According to the General Data Protection Regulation of the European Union, service providers should provide a transparent privacy policy in a comprehensible way for end-users. This paper provides an approach for Assisted Generation of Privacy Policies Using Textual Patterns. We also provide a proof of concept implementation of a tool for the privacy policy generation approach. The proposed approach supports service providers in their task of providing a comprehensible privacy policy which allows better transparency.

1 INTRODUCTION

Nowadays, internet-based services, e.g. health-care services, collect information from various sources, like cellphones or watches, and process the collected information. Services collect a wide range of data about their end-users that allow conclusions about, e.g. their habits, behavior and lifestyle. By further processing this information, a lot of information can be inferred, which end-users may not even know about. Furthermore, many different stakeholders may use these information and data.

Data protection and privacy preservation are the main goals of legal regulations like the European General Data Protection Regulation (GDPR) (European Commission, 2016). Due to these regulations, service providers are obliged to provide insights into the processing of personal data to their end-users and to communicate their practice for protecting privacy. To comply with data protection legislation, privacy policies are widely used as an important legal foundation for data handling. Privacy policies should assist end-users by providing information about "what information will be collected, how it will be used, and with whom it will be shared" (Bhatia et al., 2019). Yet, a review of current privacy policies shows that privacy policies do not provide appropriate information to the end-users (Kelley et al., 2009; Kelley et al., 2010; Probst and Hansen, 2013; Bhatia et al., 2019). The key issues are that they are not well understandable and not sufficiently transparent. They overwhelm end-users with a wall of text. Our privacy policy generation approach has to improve the awareness aspect of privacy policies and assist service providers in generating an appropriate privacy policy document that complies to the data protection legislation and is comprehensible to end-users. Furthermore, incomplete privacy policies make it difficult for end-users to have control over their data and to know when any other stakeholder has access to their data. Consequently, information given in privacy policies can affect end-users' perception of their privacy (Bhatia et al., 2019; Acquisti et al., 2008). In addition, incompleteness in privacy policies prevents users from knowing the potential consequences of such disclosures. Service providers can improve the quality of their privacy policies to help end-users make more informed decisions about using their services. However, service providers and end-users have different perspectives on privacy policies:

Service Provider's Perspective. The service provider's duty is to ensure the lawful implementation of data protection and the practice of providing accessible information about the handling of end-users' data by using privacy policies. Thus, the service providers have an incentive to address their end-users' privacy concerns for financial or reputational success. However, in social networking service providers depend on large amounts of data to deliver their services accurately. The more data is available, the more reliable the data analysis is and the higher the added value for the service

Mohammadi, N., Leicht, J., Goeke, L. and Heisel, M.
Assisted Generation of Privacy Policies using Textual Patterns.
DOI: 10.5220/0009371103470358
In *Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2020)*, pages 347-358
ISBN: 978-989-758-421-3

(Alexander Rossnagel, 2016). Service providers often exploit the legal limits of data protection to collect as much data as possible. For convenience, most service providers inform their end-users about their privacy policy electronically. Before using an application, end-users need to accept the terms of use and privacy policy to gain access to the service's functionalities. Through this procedure, the service provider complies with the data protection regulation and gets the right for the usage of all data collected during the use of the service. Similarly, for the purpose of transparency of data processing, websites obtain end-users' agreements to the storage of cookies by the implementation of pop-ups before fully displaying their service.

End-user's Perspective. To protect the end-users from losing their freedom of self-determination and from unwanted intrusion into their personal data, regulations are specified to stipulate the right to be informed before accepting the terms of a service. Consequently, end-users are confronted with privacy policies whenever they use a service or interact with a website. Privacy concerned individuals are given the opportunity to read through lengthy texts of privacy policies as supplied by most service providers to accommodate the governmental regulations. Still, the majority does not seem to be interested in informing themselves on the consequences of creating a new user account at more or less reputable service providers. Accepting privacy policies has become a type of ritual that is being performed before access to a service is granted (Jeanette Hofmann and Benjamin Bergemann, 2016): scroll down, tick checkbox, continue. Most end-users are overwhelmed by the exercise of their rights. Furthermore, they confide in the legislative power to control service providers and feel safe when accepting all terms of use without proper study of the text.

In this paper, we provide the Pattern-based Privacy Policy Generation (3PG) approach. It assists service providers in generating privacy policies for their services. Our approach aims to satisfy the GDPR requirements. It is tool-supported and based on textual patterns. 3PG allows to instantiate textual patterns for creating a privacy policy for a specific service. The tool also allows to build or modify patterns that can be used later in the instantiation process. Although, the approach assists service providers, our patterns also consider end-users' needs. The statements generated using our patterns have fewer complex constructs, which also enhances comprehensibility for end-users.

The remainder of this paper is structured as follows: Section 2 introduces fundamental concepts of data protection. Section 3 discusses our pattern-based privacy policy generation approach. It also describes how 3PG helps to create comprehensible and GDPR compliant privacy policy statements for privacy policies. Section 4 presents the proof of concept implementation of our proposed approach along with a running example. We present related work in Section 5. Section 6 discusses our approach and gives hints about limitations and benefits and compares it to related work. Section 7 concludes this paper and gives some outlines for future work.

2 DATA PROTECTION

In this section, we briefly introduce the fundamental concepts for our approach.

General Data Protection Regulation of the European Union. The GDPR extends the range of application to all companies and institutions that process data of individuals, that are located in the EU. With this regulation, service providers not only have to ensure secure data processing, but they also must prove a proper implementation of it. Specifying rights for end-users, the GDPR declares amongst other things that the end-user must be informed about the data processing, its purpose and its manner. End-users can request access to processed personal data or even object to the processing. Latter may result in the service provider's obligation to delete end-users' data.

Terminology. A *data subject* is "an identifiable natural person, who can be identified directly or indirectly, in particular by reference to an identifier such as a name, identification number, location data, online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person" (European Commission, 2016). The term "data subject" is called "personally identifiable information" (PII) principal in ISO/IEC 29100.

The *data controller* is "the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data" (European Commission, 2016). The term "data controller" is called "PII controller" in ISO/IEC 29100. The service provider, in this work, acts as data controller and is responsible for the legal and compliant operation of the services.

Privacy Policy. Transparency is one of the six protection goals for privacy engineering defined by Hansen et al. (Hansen et al., 2015). It describes the understanding and reconstruction of legal, technical and organizational conditions that should be considered before, during, and after the process of data handling. Transparency can be achieved by defining privacy policies (Probst and Hansen, 2013). The latter assists end-users by disclosing "which information will be collected, how it will be used, and with whom it will be shared" (Kelley et al., 2009). Furthermore, privacy policies should aid and inform end-users about possible options in controlling, removing, or modifying gathered information during the use of a service. To fulfil these requirements, a well-designed and structured privacy policy should support end-users with their inquiry about data processing and their validation of its importance.

3 PATTERNS FOR GENERATION OF PRIVACY POLICIES

Our approach of privacy policy patterns is to provide guidance on the specification of common types of privacy policy statements and their relevant obligation statements. The obligation statements are required by data protection regulations on particular actions performed by a service on end-users' data. This approach decreases the creation time of privacy policies and makes them easier to draft. It also improves the quality of these statements in terms of comprehensibility and the completeness of information that should be made available to the end-users.

Our privacy policy patterns contain fixed text passages that mostly address either privacy requirements or data protection regulations, e.g. the GDPR. In addition to the fixed text passages, the structure of textual patterns may contain placeholders. These placeholders are replaced with information regarding the actions on personal data of end-users performed by the considered service during the instantiation of a pattern. In the structure of a privacy policy statement, the following information shall be provided: 1) actions performed on the end-user's data, 2) the condition under which an action is performed on the end-user's data, and 3) the purpose for which the end-user's data is used.

We first present a simplified conceptual model of our approach. Afterwards, we explain the process of our approach, followed by a description of elements necessary in the construction of statements that should be included in privacy policies.

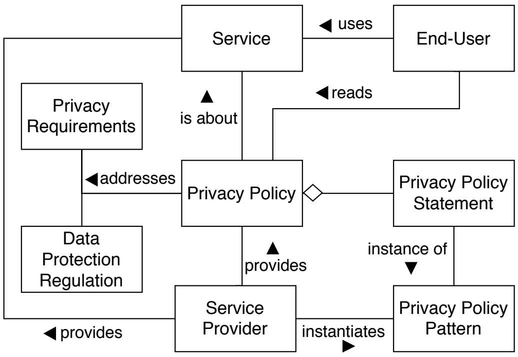


Figure 1: Simplified Conceptual Model.

3.1 Conceptual Model

Figure 1 shows the simplified conceptual model of our pattern-based approach. The privacy policy and consequently the service itself are restricted by *Data Protection Regulation*, e.g. the GDPR. These regulations state the rights and duties with regard to the end-user's privacy. The service provider is interested in adhering to the data protection regulations because substantial penalties are the consequence of disobeying the regulations. In addition to the data protection regulation, the privacy policy of a service should address *Privacy Requirements*. End-users use the service and provide their personal data to the service. A *Service* is provided by a service provider and processes end-users' data.

A service provider represents a data controller as a legal entity providing a service which stores and/or processes personal data. The service provider has, with respect to the stored/processed data, the obligations stipulated by the GDPR. The service provider that acts as data controller shall manage the privacy policies together with the data and data protection practices and other involved third-party service providers in a way that the services provided by these providers only perform actions that are permitted. The privacy policies are used by the service providers as legal foundation for data handling. The end-users read privacy policies to raise awareness about the provided service and how their data is handled by this service.

In this work, the end-user has the role of the data subject as defined in data protection regulation. The data subject is a person whose data is stored and/or processed by the service. The end-user has, with respect to his/her personal data, the rights stipulated by the GDPR.

A privacy policy consists of one or several *Privacy Policy Statements*, which address *Privacy Requirements* as well as *Data Protection Regulation*. Service

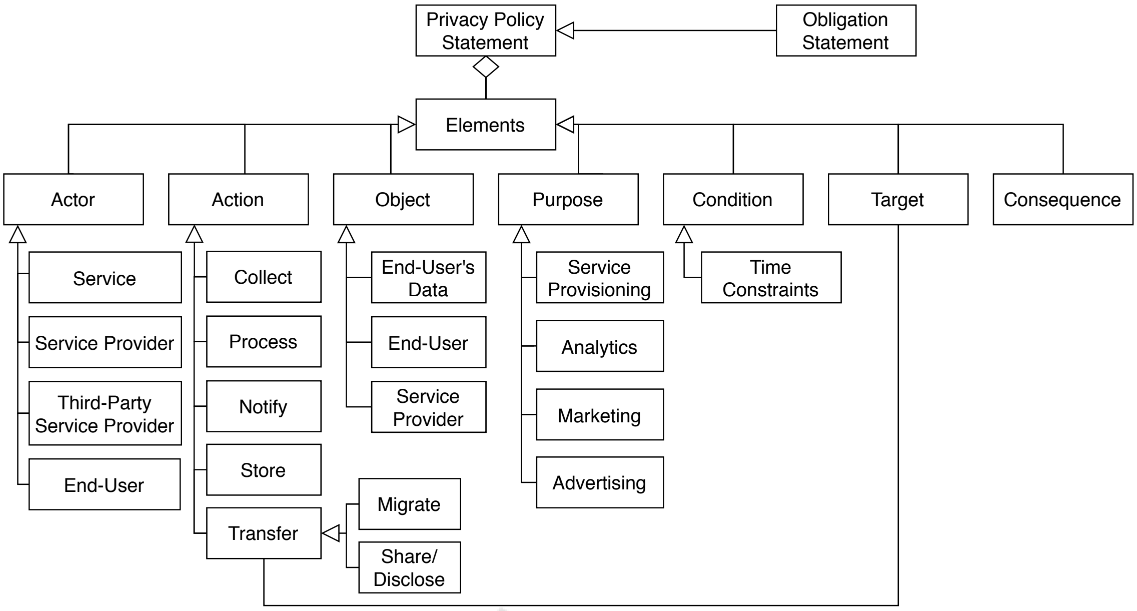


Figure 3: Elements of a Privacy Policy Statement.

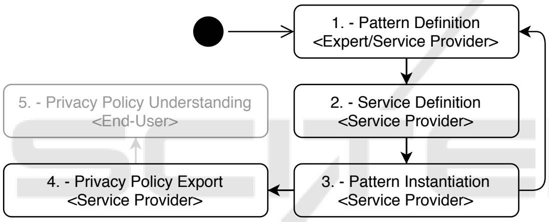

1. - Pattern Definition <Expert/Service Provider>
2. - Service Definition <Service Provider>
3. - Pattern Instantiation <Service Provider>
4. - Privacy Policy Export <Service Provider>
5. - Privacy Policy Understanding <End-User>

Figure 2: Policy Generation Process.

providers may use privacy policy patterns to generate privacy policies by instantiating these patterns. Each *Privacy Policy Statement* is then an instance of a privacy policy pattern.

3.2 Process

The policy generation process involves two parties (see Figure 2), privacy *experts* and the *service provider* or their representatives. The experts are responsible for the definition of privacy policy patterns. Service providers are responsible for the definition of a model of the provided service (cf. Section 4), which includes all information about data that is handled and purposes for data collection and processing. In a next step, the service provider instantiates the given privacy and obligation patterns. In case the service provider notice a gap in the existing patterns, service providers can either give feedback to the experts or directly define new patterns. The last step of the policy generation is the export of the finished privacy policy. This can be done in the well-known textual format or in alternative formats, e.g. nutrition label (Kelley et al., 2009) (cf. Section 5). Finally, the result of the process is the end-user being able to understand the privacy policy. Thus, the approach achieves transparency, which is required by data protection legislation.

3.3 Structure and Elements of Statements

Figure 3 illustrates the different elements that a privacy policy statement may contain. *End-User's Data* is a type of *Object*, which is one of the *Elements* that should be included in *Privacy Policy Statements* as a key element to specify which data of the end-user is used by the service. There are different categories of *End-User's Data*, e.g. health information, which we described in previous work (Gol Mohammadi et al., 2019). *Conditions* (e.g. *Time Constraints*) and rules refer to the *Actions* that are performed by the *Service*. The association of *End-User's Data*, *Actions*, and *Purposes* is important, because it is necessary to track which *Actions* are allowed to be performed for which *Purpose*.

The different categories of *Actions* define what can be done with *End-Users' Data* that is provided to the *Service*. Categories of *Actions* are as follows:

- *Collect*: Collection of *End-Users' Data* that is directly or indirectly supplied by the end-user
- *Process*: Processing of data resulting in new data
- *Notify*: Notification of the end-user of specific events

- *Store*: Storing of data supplied by the end-user
- *Transfer*: Transferring data to other locations
  - *Share/Disclosure*: Sharing data with other parties (is a kind of *Transfer* action)
  - *Migrate*: Migrating is a kind of transfer action that changes the location of data. This action has been considered separately because of the special requirement of the GDPR on migration.

Each of these *Actions* can be included in the *Privacy Policy Statement* as one of its elements. The *Transfer* action requires a *Target* element, which describes where the data is transferred.

Every *Action* has a *Purpose*. The *Purpose* of an *Action* is another category of information that is part of a *Privacy Policy Statement* (see Figure 3). We distinguish the following four categories of purposes:

- *Service Provisioning* as purpose defines whether the *Action* is necessary for service provision. It means that the *Action* is necessary to be able to provide the service, e.g. account information for online banking.
- *Analytics* as purpose defines whether end-users' data is used for performing data analytics, which means the usage of *End-Users' Data* for the creation of statistics.
- *Marketing* as purpose defines whether the data supplied by the end-user is used for marketing.
- *Advertising* as purpose defines whether *End-Users' Data* is used or shared with others for advertising.

Because of this purpose-binding of *Actions*, it is possible to decide whether the usage of the *End-User' Data* is actually necessary to provide the service or whether the end-user can decide whether to use this service. *Marketing*, *Analytics*, and *Advertising* are optional purposes, stating that the action is not actually necessary for the service to function. The difference between *Marketing* and *Analytics* is as follows. *Marketing* means the usage of end-users' data to inform new end-users about the *Service*, whereas, *Advertising* means the usage of the data to provide advertisements to the end-user that provides the data.

Some *Actions* may have certain consequences for the *End-User*, which can be expressed with the *Consequence* element.

Using the privacy policy, *Service Providers* can communicate their privacy protection practices to the *End-Users*. In addition to the *Privacy Policy Statements*, the generated privacy policy contains further statements e.g. information about the data protection officer and applied obligations. These *obligation statements* can, for example, specify the prohibitions in the processing (*Action*) of the data for specific purposes or retention/deletion conditions (see Figure 4).

We provide textual patterns for generic privacy policy statements, as well as more specific privacy policy statements. Examples of specific privacy policy statements (see Figure 4) are as follows:

- A *Notification Statement* specifies the rules and conditions regarding the question about what the *End-User* needs to be informed and how.
- A *Consent Statement* describes in which cases and under which condition the *End-Users* are required to give their consent.
- A *Collection Statement* specifies which data is allowed to be collected by the *Service Provider*. This means, it will restrict the *Collect Action* of the *Service Provider*.
- A *Usage Statement* specifies for which *Purpose End-Users' Data* is allowed to be used and processed.
- A *Storage Statement* specifies how long and where the data will be stored.
- A *Share/Disclosure Statement* defines the rules and conditions specifying which data of the *End-Users* is allowed to be disclosed to which audience.
- A *Migration Statement* describes the rules and conditions for the migration of the data to different locations.

Depending on a specific *Privacy Policy Statement* with its concerned action types, one or more obligations can be relevant to the statement. For example, upon a *Storage Statement* the *Obligation Statement*

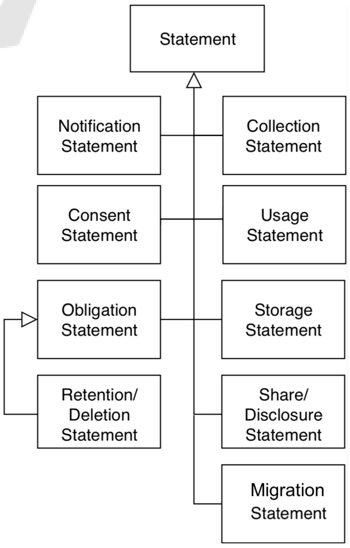


Figure 4: Different Types of Statements.

should be added to specify when and how the service provider deletes this data. In our work, we also provide patterns for *Obligation Statements* that can be part of privacy policy patterns. The structure of obligation statement patterns is similar to the structure of privacy policy patterns.

Both types of patterns include the following constructs:

- Fixed text: Text passages of a pattern that cannot be modified during the instantiation of the pattern.
- Generic placeholders: References to certain types of elements of the considered service. During the instantiation of a pattern a generic placeholder is replaced by information of a service element, whose type corresponds to the referenced type in the generic placeholder. Generic placeholders are marked by opening and closing square brackets "[", "]". As an example, the generic placeholder *"Action"* within square brackets in the privacy policy pattern in Example 1. references all elements of the category *Action*. During the instantiation of the privacy policy pattern, the designer of a specific service can replace the placeholder with the information of a service element of a subtype of *Action*. Example 2. shows a more specific collection statement pattern and an instance of the given pattern.

Example 1. A Generic Statement Pattern: *The [Actor] will [Action] [Object] for [Purpose].*

Example 2. A Collection Statement Pattern: *The [Service] will [Collect] [End-User's Data] for [Service Provisioning].*

An Instance of the Collection Statement Pattern: *The navigation service will collect your location data for service provisioning.*

For structuring both, fixed and appropriate generic parts of our patterns, we have identified a number of keywords for each of the elements, e.g. for expressing actions or conditions such as time constraints. We have analyzed the state of the art and terminologies used in current privacy policies. Then, we created a mapping of synonym words to the terms used in the GDPR (Gol Mohammadi et al., 2019). For example, multiple terms may be used in order to express "collecting" end-users' data, e.g. obtain, gather, get, receive, etc. In our patterns, we use our identified keywords (in this case "collect"), which are based on the GDPR terminology. In this way, the privacy policies created using our textual patterns will be consistent and uniform.

Further examples for privacy policy and obligation statement patterns are presented in the next section.

4 PROOF OF CONCEPT IMPLEMENTATION

We provide tool support for our textual privacy policy patterns and obligation statement patterns, namely *Pattern-based Privacy Policy Generation (3PG)*-tool. This tool provides two editors: 1) the *Pattern-Editor*, and 2) the *Pattern-Instantiation-Editor*. The Pattern-Editor supports the creation and management of privacy policy patterns and obligation statement patterns. The Pattern-Instantiation-Editor supports the instantiation of privacy policy and obligation statement patterns and the management of corresponding pattern instances.

Both editors are realized as Eclipse plugins by using the *Eclipse Modeling Framework (EMF)*[1]. Eclipse plugins are implemented in the programming language *Java*. EMF supports the implementation of Eclipse plugins by providing a model-driven approach. Therefore, the framework provides a set of language constructs for the creation of metamodels that enable the specification of an abstract syntax of model information. The provided constructs correspond to the elements of class models that are part of the *Unified Modeling Language (UML)*. Thus, EMF-metamodels contain elements like classes and associations.

Information about the service and the data used by the service is necessary for the instantiation of privacy policy patterns. To make these data usable, a model of the service is created, using a metamodel. The tool provides a tree editor, that allows the service provider to create a model of the service. Figure 6 shows the service definition editor with some example data.

The Pattern-Editor and Pattern-Instantiation-Editor are explained in the Sections 4.2 and 4.3. The description of the editors is illustrated with the *Pay-As-You-Drive (PAYD) Insurance* example, which is presented in Section 4.1 below.

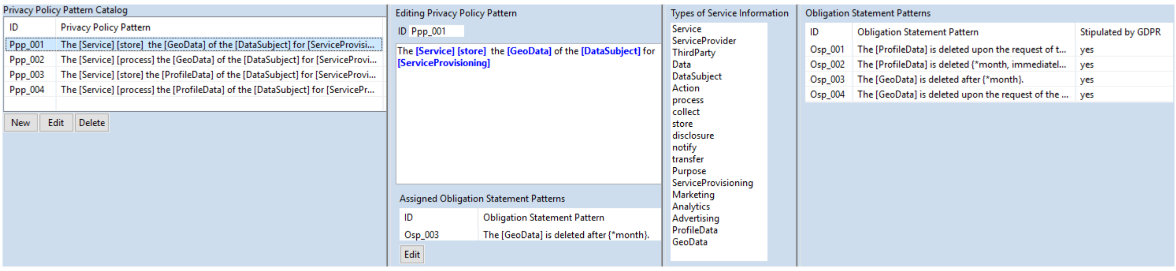
4.1 The Pay-As-You-Drive Case Study

This section presents an example of a real-life cloud computing service that is studied in the RestAssured[2] project.

The PAYD case study is a cloud computing service for automotive insurance companies. PAYD enables insurers to offer innovative, cost-effective and usage-based automotive insurance products. This is achieved by collecting and analyzing the driving data of insurance customers. Since these driving data are

[1] https://www.eclipse.org/modeling/emf/

[2] See https://restassuredh2020.eu



Privacy Policy Pattern Catalog

ID	Privacy Policy Pattern
Ppp_001	The [Service] [store] the [GeoData] of the [DataSubject] for [ServiceProvisi...
Ppp_002	The [Service] [process] the [GeoData] of the [DataSubject] for [ServiceProvi...
Ppp_003	The [Service] [store] the [ProfileData] of the [DataSubject] for [ServiceProvi...
Ppp_004	The [Service] [process] the [ProfileData] of the [DataSubject] for [ServicePr...

Editing Privacy Policy Pattern

ID Ppp_001

The [Service] [store] the [GeoData] of the [DataSubject] for [ServiceProvisioning]

Assigned Obligation Statement Patterns

ID	Obligation Statement Pattern
Osp_003	The [GeoData] is deleted after {*month}.

Types of Service Information

Service
ServiceProvider
ThirdParty
Data
DataSubject
Action
process
collect
store
disclosure
notify
transfer
Purpose
ServiceProvisioning
Marketing
Analytics
Advertising
ProfileData
GeoData

Obligation Statement Patterns

ID	Obligation Statement Pattern	Stipulated by GDPR
Osp_001	The [ProfileData] is deleted upon the request of t...	yes
Osp_002	The [ProfileData] is deleted {*month, immediatel...	yes
Osp_003	The [GeoData] is deleted after {*month}.	yes
Osp_004	The [GeoData] is deleted upon the request of the ...	yes

Figure 5: Pattern-Editor Tab for Creation and Management of Privacy Policy Patterns.

personal data of the insurance customers, the requirements concerning data protection and data security are very high. The *GDPR* is one of the effective regulations for the provided services. Accordingly, insurers take the *data controller* role from GDPR. The insurance customers as the end-users (the drivers) have the role of *data subjects* from GDPR.

The first service that is provided is the *PAYD web application*. This web application is used by the *insurance customers* (end-users). By using the *PAYD web application*, the insurance customers are able to register for an insurance product and get information about their current insurance conditions. The end-users provide their personal data as *insurance customer data* to the *PAYD web application*. The access, processing and storage of the *insurance customer data* is described in the provided privacy policy.

The second service involved in this scenario is a *telematic service*. This service is responsible for receiving the *driving data* of the insurance customers. The collected driving data is transferred to the cloud infrastructure of the corresponding insurer. There, the driving data is analyzed by *insurance analysts* to determine the individual insurance premiums for insurance customers.

The privacy policy should include information on the access, processing and restrictions provided by the service providers with regard to the amount of the driving data.

In the description of the Pattern-Editor (Section 4.2) and Pattern-Instantiation-Editor (Section 4.3), the processing of personal customer data and driving data within the PAYD insurance scenario is considered as a running example. Since the focus lies on the design of patterns and their instances, only partial data from the PAYD insurance scenario is considered.

4.2 The Pattern-Editor

The *Pattern-Editor* of the 3PG-tool allows privacy experts to create and manage privacy policy patterns and obligation statement patterns. For this purpose, the 3PG-tool provides an editor tab for each type of pattern. In the following, we describe the main constructs of a pattern.

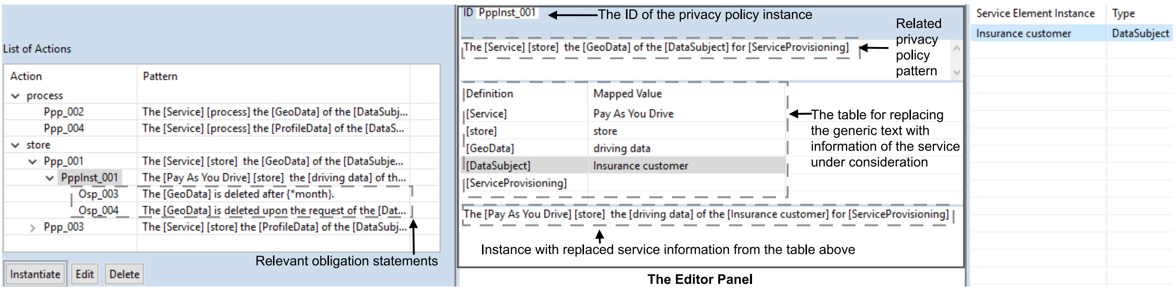
A pattern, which is identified by a unique identifier, consists of *fixed text* and *generic text*. The *fixed text* represents the text in a pattern that cannot be changed during the instantiation of the pattern. In contrast to the fixed text part of a pattern, different types of *generic text* are replaced by appropriate information during the pattern instantiation. A generic text has a length and is inserted at a certain position in the fixed text (marked with placeholders). In the following, the different types of generic text are described.

Patterns are instantiated in a semi-automatic manner by using the aforementioned service definition model. A *service element placeholder* defines a placeholder that references one or several types of service elements. During the instantiation of a pattern, a service element placeholder is replaced by the information represented by a service element of the appropriate type (see Section 4.3). Such a placeholder in a pattern is indicated by an opening and closing square bracket ([,]).

Another type of generic text is a *text choice*. A *text choice* contains one or more text choice elements. During the instantiation of a pattern, one of the specified text choice elements can be selected to replace the text choice placeholder. A text choice element represents either an invariable keyword or a piece of text that contains a placeholder that should be replaced with appropriate information. This information is supplied by the service provider, who instantiates the pattern. Such a text choice placeholder is indicated by the symbol "*".

Free text is the last type of generic text. It can be replaced with any appropriate information by the service provider, during the pattern instantiation. It contains only a description that explains the domain of information that the service provider should specify. Free text is indicated by angle brackets (<, >).

Figure 5 shows the definition of a privacy policy pattern for services in the dedicated tab of the Pattern-



List of Actions

Action	Pattern
process	
Ppp_002	The [Service] [process] the [GeoData] of the [DataSubj...
Ppp_004	The [Service] [process] the [ProfileData] of the [DataS...
store	
Ppp_001	The [Service] [store] the [GeoData] of the [DataSubje...
PppInst_001	The [Pay As You Drive] [store] the [driving data] of th...
Osp_003	The [GeoData] is deleted after {*month}.
Osp_004	The [GeoData] is deleted upon the request of the [Dat...
Ppp_003	The [Service] [store] the [ProfileData] of the [DataSubj...

ID PppInst_001

The [Service] [store] the [GeoData] of the [DataSubject] for [ServiceProvisioning]

Definition	Mapped Value
[Service]	Pay As You Drive
[store]	store
[GeoData]	driving data
[DataSubject]	Insurance customer
[ServiceProvisioning]	

The [Pay As You Drive] [store] the [driving data] of the [Insurance customer] for [ServiceProvisioning]

Service Element Instance	Type
Insurance customer	DataSubject

Instantiate
Edit
Delete
Relevant obligation statements
The ID of the privacy policy instance
Related privacy policy pattern
The table for replacing the generic text with information of the service under consideration
Instance with replaced service information from the table above
The Editor Panel

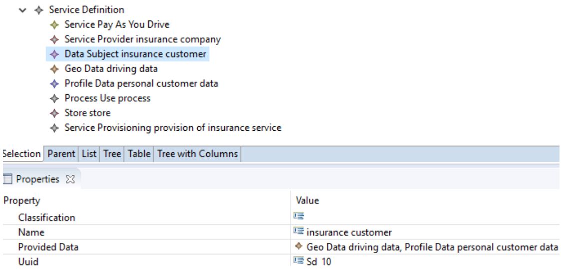
Figure 7: Instantiation of a Privacy Policy Pattern in the Pattern-Instantiation-Editor.

Editor. The middle part in the editor tab enables editing of privacy policy patterns. It contains a list named *Types of Service Information*. This list includes all types of service element instances that can be referenced by a service element placeholder. The list is created, when the appropriate service definition metamodel, which corresponds to the 3PG-metamodel (cf. Section 4), is loaded into the Pattern-Editor for the first time. For the actual editing of a privacy policy pattern, a text field is provided, which is placed below the text field for depicting the *pattern ID*. A service information type can be inserted into a privacy policy pattern by double-clicking on the corresponding list item or manually typing it into the text field. In case of a manual entry by pattern designers, a syntax check ensures that incorrect type names and other syntax errors are detected. Figure 5 shows the definition of a privacy policy pattern that refers to the action of storage that is performed in the context of a service. The storage action concerns geodata that belongs to end-users of the service for service provisioning.

The table below the stylized text field for editing privacy policy patterns contains the obligation statement patterns that are related to the current privacy policy pattern. The relations of a privacy policy pattern to obligation statement patterns can be added and removed through a dialog that is started by clicking the "Edit" button.

On the left-hand side of the editor tab, the catalog of privacy policy patterns is shown. It provides functionalities for starting the development or modification of privacy policy patterns, as well as the deletion of privacy policy patterns.

The right-hand part of the editor tab displays the specified obligation statement patterns. These obligation statement patterns have been defined using the second editor tab of the Pattern-Editor. In contrast to privacy policy patterns, the obligation statement patterns specify the additional information that indicates whether an obligation is stipulated by the GDPR. Since the functionalities and techniques for developing and managing obligation statement patterns are quite similar to corresponding features for privacy policy patterns, this editor tab is not further explained.



- Service Definition
  - Service Pay As You Drive
  - Service Provider insurance company
  - Data Subject insurance customer
  - Geo Data driving data
  - Profile Data personal customer data
  - Process Use process
  - Store store
  - Service Provisioning provision of insurance service

Property	Value
Classification	
Name	insurance customer
Provided Data	Geo Data driving data, Profile Data personal customer data
Uuid	Sd 10

Figure 6: EMF-Tree Editor of *PAYD Insurance* Service.

4.3 Pattern-Instantiation-Editor

The *Pattern-Instantiation-Editor* allows the instantiation of existing privacy policy patterns and their corresponding obligation statement patterns, which are defined with the Pattern-Editor (cf. Section 4.2). During the instantiation, the generic text of a pattern is replaced by concrete information from the service definition model.

In our example, service element placeholders are replaced by service element instances that represent service elements from the PAYD use case (cf. Section 4.1). Figure 6 shows the definition of the service elements of the PAYD use case in the corresponding EMF-tree editor. The upper tree view lists the different service elements. A service element comprises of the actual service information and its type. The service information displayed in the tree editor, is contained in the service element instances created by the Pattern-Instantiation-Editor. During the instantiation of privacy policy and obligation statement patterns for the PAYD use case, service element placeholders are replaced by service element instances of the appropriate type. For clarity, only a partial extract of the PAYD service information is represented in the tree.

The properties of the currently selected service element are displayed in the lower property panel of the tree editor. These properties include the name of the selected service element and its unique identifier

(Uuid), as well as associations of the service element with other service elements.

Figure 6 shows the properties for the service element "insurance customer" of the type *data subject*. It contains the reference *Provided Data* that implements the association of a data subject to the data that is provided to the relevant service. Within the PAYD use case, insurance customers provide their *driving data* and *personal data* to the service.

Figure 7 shows the instantiation of the privacy policy pattern *Ppp_001* in the Pattern-Instantiation-Editor.

Privacy policy patterns and their corresponding instances are assigned to the types of actions in the service under consideration. For the PAYD use case, the appropriate privacy policy patterns are assigned to the actions *store* and *process* (see Figure 6). This assignment is indicated by a tree in the left-hand side of the editor (see Figure 7). The tree for the PAYD use case in Figure 7 shows that only such privacy policy patterns are assigned to a certain action, when a privacy policy pattern contains a service element placeholder that refers to the type of the action. For example, all privacy policy patterns that refer to the *process* action contain the service element placeholder "[process]".

A privacy policy pattern for a particular action can be instantiated by selecting the pattern in the tree and clicking the *Instantiate*-button (bottom left in Figure 7). The created instance can then be edited in the middle panel of the editor. In the following, the different components of editing are explained in a top-down order. First, the editor panel provides a text field for editing the ID of the privacy policy instance. Another text field displays the related privacy policy pattern. The table (in the middle of the editor panel) allows replacing the generic text with information of the service under consideration. To do so, the table lists all definitions of generic text in the associated pattern in the *Definition* column. The *Mapped Value* column contains the information that instantiates the corresponding placeholder in the instance. In the instantiation of the privacy policy pattern *Ppp_001* (see Figure 7), the *Definition* column contains only service element placeholders. Thus, the *Mapped Value* column contains only service element instances. The service element placeholders "Service", "store" and "GeoData" have already been replaced by appropriate service element instances. The value for the placeholder "store" that represents the action is assigned automatically. The replacement of a generic text is started by a double-click on the appropriate row in the table. All suitable service elements are then displayed in the table on the right, where the desired element can be selected by double-clicking on the appropriate row. Figure 7 shows the substitution for the placeholder "[DataSubject]". In the right-hand part of the editor, the table provides the only relevant service element instance, the "insurance customer", as a data subject.

At the bottom in the middle of the editor, the text field displays the current text of the privacy policy instance during the instantiation process. If a generic text is not replaced yet, the generic text of the related pattern is displayed.

After the instantiation of a privacy policy pattern, the relevant obligation statement patterns are displayed in the tree view under the created privacy policy instance. The obligation statement patterns *Osp_003* and *Osp_004* are associated to this instance in Figure 7 (left-hand side), which correspond to the privacy policy instance *PppInst_001*, that is an instance of the privacy policy pattern *Ppp_001*.

The instantiation of obligation statement patterns is similar to the previously described instantiation of privacy policy patterns and thus not further described.

5 RELATED WORK

In software engineering, patterns are an adequate approach for solving problems that occur frequently in a specific domain. There are various types of patterns, for example design patterns, requirements patterns, and risk patterns (Buschmann et al., 1996; Brambilla et al., 2010).

Privacy Policies. A variety of research approaches address deficiencies of privacy policies. Pollmann and Kipker suggest a competitive rating system, where end-users and experts can rate service providers in their attitude towards personal data processing (Pollmann and Kipker, 2016). This approach encourages the service providers in respecting end-users' privacy.

There are several works that address the improvement of comprehensibility of privacy policies by redesigning the structure of privacy policies. Kelly et al. discovered that lengthy full-text policies cause disadvantages in comprehension and retrieval of required information by end-users (Kelley et al., 2009). The authors developed the *Privacy Nutrition Label* approach, that displays privacy policies in an easier to understand way using a grid layout. A study confirmed that the presentation and comprehension of information within the *Privacy Nutrition Label* is less time-consuming and less complex compared to the scanning of lengthy full-text policies (Kelley et al., 2009; Kelley et al., 2010).

Ghazinour et al. (Ghazinour et al., 2009) developed a *Model for Privacy Policy Visualization* that improves comprehensibility of privacy policy statements. Their *Model for Privacy Policy Visualization* defines different elements (e.g. entity, relation, privacy notation, group attributes, and default values) that are displayed using various symbols (Ghazinour and Albalawi, 2016).

Our work differs from this kind of visual representation approaches, since we aim at improving the quality of privacy policy documents using textual patterns. We also address the obligation from the regulation that comprehensibility of textual versions of privacy policies should be improved. Our pattern approach can be combined with the nutrition label representation of privacy policies, with the full text policies as legal background.

Bhatia et al. analyze and identify semantic roles in the privacy practice statements provided by service providers (Bhatia et al., 2019). We considered their analysis and results in the structure of our textual patterns for the privacy policy statements.

IBM developed a tool for P3P privacy policy generation (IBM Corp., 2000). This tool was abandoned together with the P3P language. The “IBM P3P Policy Editor” was developed to support owners of websites in the creation of P3P policies. The editor is of no use today, as P3P is discontinued, and does not improve comprehensibility of the created policies.

Reidenberg et al. developed a scoring system to evaluate ambiguity in privacy policies (Reidenberg et al., 2016). In other work the authors showed that end-users and privacy experts interpreted the same policy differently (Reidenberg et al., 2015). With our approach we envision to reduce ambiguity in privacy policies and assist service providers in creating understandable privacy policies.

Da Silva et al. proposed a framework and tool for assistance in privacy policy generation and management (da Silva et al., 2016). The tool can be used to analyze existing privacy policies using natural language processing. The tool however does not provide any information on what statements the policy should contain, whereas the framework we propose in this paper provides statements, which just need to be filled with the service information. The authors of the *RSLingo4Privacy* approach make use of formal languages, which allows for reasoning about the policies. A combination of these languages with our textual pattern approach may improve our framework.

Many online privacy policy generators exist, but these generators do not help in creating compliant privacy policies. These websites clearly state that the created policies are no legal advice and should not simply be added to websites. Our approach is no automatic generator but should assist service providers in creating compliant and easily understandable privacy policies. A combination of our pattern-based approach with the automatic generators could improve policy generator websites.

Textual Patterns. Textual patterns are used in the specification of requirements (Withall, 2007). Withall identifies 37 requirement patterns, that are divided into eight domains: fundamental requirement patterns, information requirement patterns, data entity requirement patterns, user function requirement patterns, performance requirement patterns, flexibility requirement patterns, access control requirement patterns, and commercial requirement patterns. The author provides guidelines and examples for formulating requirements in natural language. He aims at writing textual requirements, which also consider domain knowledge. Our work differs from Withall’s, because we provide patterns for privacy policy statements based on the GDPR.

Pohl introduces patterns which describe scenarios in a structured textual format. Pohl’s requirements templates (Pohl, 2010) include possible exception scenarios.

In agile software development, for specifying requirements there are textual patterns for drafting the user stories (Cohn, 2004).

The idea of using a structured textual format in our privacy policy statements pattern is similar to these requirements templates. These works do not provide specific notations to document data protection requirements nor any other non-functional requirements, and only consider functional requirements.

Del-Río-Ortega et al. develop some textual patterns for defining process performance indicators in business process management (Del-Río-Ortega et al., 2012). The authors do not provide specific notations to document and model privacy policies or even privacy requirements and only consider key performance indicators in business processes.

Beckers et al. present a method for the specification of security requirements for cloud computing systems (Beckers et al., 2011). The authors provide textual security requirement patterns. We build our approach upon this work and present more specific patterns for privacy policy statements and obligations. For this purpose, our work extends the approach presented by Beckers et al. by adding elements that address privacy requirements and data protection requirements. The added elements conform to the GDPR.

6 DISCUSSION

The procedure presented in this paper was developed based on discussions with practitioners from a European project on data protection. Parts of our approach have been discussed with privacy consultants. The privacy consultants mentioned that this structured approach supports the specification of privacy policies, as well as the generation of privacy requirements for requirements specification.

It increases the usage of models of the service and generating privacy policies in a semi-automatic manner instead of texts from standards and law, which eases the effort of understanding the service and its data handling process and data protection practices. Additionally, it facilitates the comprehension of the privacy policy document and unifies used terminologies in documentation significantly. Furthermore, our approach provides the means for abstraction of a complex system and structured reasoning for data protection. It also ensures that the privacy policy statements describe concretely and explicitly which actions are performed for which purpose on end-users' data. Accordingly, instead of long, complex and abstract statements, our approach leads to simple, short and more comprehensible statements for end-users. Furthermore, privacy policies are uniform and consistent through the adoption of a set of pre-defined, GDPR-compliant terms and keywords for describing data protection matters.

One issue that needs further investigation is scalability, both, in terms of effort needed, by the privacy experts and service providers, in order to enter all information about the service, as well as the tool support. We will use the approach for different scenarios to investigate if the method scales for different domains and applications.

In this paper we focused on the generation of simple and easily understandable privacy policy statements, however in previous work we also considered the structured representation of privacy policies (Gol Mohammadi et al., 2019).

We aim to conduct an empirical study with our tool in order to analyze the amount of time that can be saved when using our approach. Furthermore, we will analyze the achieved user experience (in terms of comprehensibility) when reading such documents generated using our approach. We aim to compare it against conventional text-based approaches. Our approach will also undergo a series of further usability tests, which shall discover issues with its use in a productive environment. We aim to identify usability issues and resolve these in order to further improve the user experience (for designers of the service).

7 CONCLUSIONS AND FUTURE WORK

In this paper we presented a textual pattern-based approach for the generation of privacy policy statements. These privacy policy patterns can be used to improve comprehensibility of text in privacy policies and address incompleteness in the information presented to the end-users with respect to privacy practices. In this work, we analyzed GDPR obligations with respect to the transparency of information. Furthermore, we analyzed the structure of privacy policy statements and identified (i) the important information in each privacy policy statement, (ii) obligations defined in the GDPR, e.g. retention, and (iii) keywords that are used in these statements for expressing conditions. In addition, we defined several privacy policy patterns based on these categories and keywords that can be instantiated for a specific service.

The developed tool will be further extended in order to address all the different types of privacy policy requirements captured from the GDPR. Our textual pattern-based approach can be used by service providers to semi-automatically generate privacy policy documents and align them with the software's design and also with data protection regulations. Question wizards can also be developed to support users of our tool in instantiating privacy policy patterns. These question wizards could be used by requirements engineers or designers. Example questions can include "Which actions are performed on the personal data provided by the end-users?", "Which personal data of the end-users are logged?". In addition, designers can use our tool to developed further patterns that can be used to ensure that end-user's privacy is protected and to provide more concrete information about privacy protection practices.

Furthermore, the tool support for the definition of service information will be improved. In this context, a new graphical user interface will be implemented that enables the representation of service information in the form of service diagrams.

ACKNOWLEDGMENTS

Research leading to these results received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 731678 (RestAssured).

We gratefully acknowledge constructive discussions with our project partners.

REFERENCES

Acquisti, A., Gritzalis, S., Lambrinoudakis, C., and di Vimercati, S., editors (2008). *Digital Privacy: Theory, Technologies, and Practices*. New York: Auerbach Publications, 1st edition.

Alexander Rossnagel (2016). Datenschutz. Eine Zukunft ohne Selbstbestimmung? *Spektrum der Wissenschaft Kompakt: Der digitale Mensch*, pages 41–49.

Beckers, K., Schmidt, H., Küster, J., and Faßbender, S. (2011). Pattern-based support for context establishment and asset identification of the ISO 27000 in the field of cloud computing. In *6th Intl. Conf. on Availability, Reliability and Security*, pages 327–333.

Bhatia, J., Evans, M. C., and Breaux, T. D. (2019). Identifying incompleteness in privacy policy goals using semantic frames. *Requirements Engineering*, 24(3):291–313.

Brambilla, M., Campi, A., Ceri, S., and Quarteroni, S. (2010). Semantic resource framework. In *Search Computing - Trends and Developments [outcome of the 2nd SeCO Workshop on Search Computing]*, pages 73–84. Springer.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. (1996). *Software Patterns*. John Wiley & Sons.

Cohn, M. (2004). *User Stories Applied: For Agile Software Development* . Addison-Wesley, 1st edition.

da Silva, A. R., Caramujo, J., Monfared, S., Calado, P., and Breaux, T. (2016). Improving the specification and analysis of privacy policies. In *Proceedings of the 18th International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 336–347. INSTICC, SciTePress.

Del-Río-Ortega, A., Resinas Arias de Reyna, M., Durán Toro, A., and Ruiz-Cortés, A. (2012). Defining Process Performance Indicators by Using Templates and Patterns. In *Business Process Management*, volume 7481 of *Lecture Notes in Computer Science*, pages 223–228. Springer.

European Commission (2016). Regulation 2016/679 of the European General Data Protection Regulation. https://gdpr-info.eu. Accessed: 2018-03-02.

Ghazinour, K. and Albalawi, T. (2016). A usability study on the privacy policy visualization model. In *IEEE 14th International Conference on Dependable, Autonomic and Secure Computing, 14th Intl. Conf. on Pervasive Intelligence and Computing, 2nd Intl. Conf. on Big Data Intelligence and Computing and Cyber Science and Technology Congress, DASC/PiCom/DataCom/CyberSciTech*, pages 578–585.

Ghazinour, K., Majedi, M., and Barker, K. (2009). A model for privacy policy visualization. In *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference, COMPSAC*, pages 335–340.

Gol Mohammadi, N., Leicht, J., Ulfat-Bunyadi, N., and Heisel, M. (2019). Privacy policy specification framework for addressing end-users' privacy requirements. In *Proc. of the 16th Intl. Conf. on Trust, Privacy and Security in Digital Business - TrustBus 2019*. Springer.

Gol Mohammadi, N., Pampus, J., and Heisel, M. (2019). Pattern-based incorporation of privacy preferences into privacy policies:: Negotiating the conflicting needs of service providers and end-users. In *24th European Conference on Pattern Languages of Programs*, EuroPLoP. ACM.

Hansen, M., Jensen, M., and Rost, M. (2015). Protection goals for privacy engineering. In *IEEE Symposium on Security and Privacy Workshops, SPW*, pages 159–166.

IBM Corp. (2000). IBM P3P Policy Editor.

Jeanette Hofmann and Benjamin Bergemann (2016). Die informierte Einwilligung: Ein Datenschutzphantom. *Spektrum der Wissenschaft Kompakt, "Der Digitale Mensch"*, pages 50–59.

Kelley, P. G., Bresee, J., Cranor, L. F., and Reeder, R. W. (2009). A "nutrition label" for privacy. In *Proceedings of the 5th Symposium on Usable Privacy and Security, SOUPS*. ACM.

Kelley, P. G., Cesca, L., Bresee, J., and Cranor, L. F. (2010). Standardizing privacy notices: an online study of the nutrition label approach. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI*, pages 1573–1582. ACM.

Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer Publishing Company, Incorporated, 1st edition.

Pollmann, M. and Kipker, D. (2016). Informierte einwilligung in der online-welt. *Datenschutz und Datensicherheit*, 40(6):378–381.

Probst, T. and Hansen, M. (2013). Privacy protection goals in privacy and data protection evaluations. Technical report, Working paper, Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein.

Reidenberg, J. R., Bhatia, J., Breaux, T. D., and Norton, T. B. (2016). Ambiguity in privacy policies and the impact of regulation. *The Journal of Legal Studies*, 45(S2):S163–S190.

Reidenberg, J. R., Breaux, T., Cranor, L. F., French, B., Grannis, A., Graves, J. T., Liu, F., McDonald, A., Norton, T. B., and Ramanath, R. (2015). Disagreeable privacy policies: Mismatches between meaning and users' understanding. *Berkeley Tech. LJ*, 30:39.

Withall, S. (2007). *Software Requirement Patterns*. Microsoft Press, Redmond, WA, USA, 1st edition.