Supporting Privacy Control and Personalized Data Usage Explanations in a Context-Based Adaptive Collaboration Environment

Mandy Goram(✉) and Dirk Veiel

Faculty of Mathematics and Computer Science, FernUniversität in Hagen,
58084 Hagen, Germany
{mandy.goram,dirk.veiel}@fernuni-hagen.de

Abstract. The General Data Protection Regulation, e.g., provides the "right of access by the data subject" and demands explanations of data usages, i.e. explanations where and for what purpose personal data is being processed. Supporting this kind of privacy control and related personalized explanations of data usage in context-based adaptive collaboration environments are big challenges. Currently, users cannot retrace the usage and the storage of their personal data in context-based adaptive collaboration environments. We address the aforementioned challenges by developing a context-based adaptive collaboration platform, the CONTact platform, that can be linked to or integrated into different kinds of collaboration environments (e.g., meinDorf55+, a novel community support system for elderly). The CONTact platform supports users with privacy control and personalized explanations of data usages. In this paper we present an excerpt of our extended domain model and two sample situations when privacy control and personalized explanations get relevant. We use a sample ontology that is based on our domain model to illustrate the related processes and rules. Using our approach users can control their data usage and are able to get personalized explanations of their data usage in a context-based adaptive collaboration environment. This helps us observing legal regulations, e.g. privacy laws like the GDPR.

Keywords: Context-based · Adaptive · Collaboration environment · Privacy control · Personalized explanations · Legal regulations · GDPR

1 Introduction

Considering legal regulations has become an important aspect of software development. The General Data Protection Regulation (GDPR) demands comprehensibility of personal data processing and provides the "right of access by the data subject"[1]. Due to that software providers must be able to reveal what data is stored and processed by their applications and services. The ongoing trend to personalize content and applications requires the development of more sophisticated approaches. These should take the current situation of their users into account and provide adequate support.

[1] https://gdpr-info.eu/art-15-gdpr/.

G. Bella and P. Bouquet (Eds.): CONTEXT 2019, LNAI 11939, pp. 84–97, 2019.
https://doi.org/10.1007/978-3-030-34974-5_8

Context-aware systems are able to support personalization with regard to the current situation of related users. To support users in certain situations (e.g. create documents in a collaborative work environment) the system must be aware of the user's situation and the related socio-technical environment, i.e. the context. Dey [1, P. 5] defines that "Context is any information that can be used to characterize the situation of an entity". Considering the user's context, a system becomes context-aware, as soon as it "uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task" [1, P. 5]. The disadvantage of context-aware systems is the predefined and fixed context to a certain domain and some few situations [2]. Because it is not possible to predefine all possible situations of the users and their interactions (i.e. at design time), the systems are limited in supporting users. [3] present an approach that uses a formal context model representing the socio-technical system. This enables modelling of and acting on different kinds of interactions and situations of a specific domain even after the final shipment of the underlying application(s). They support this kind of opportunity by adding an extra level of abstraction (i.e. a formal context model) and by separating the application from the so-called Adaptation Runtime Environment. Such a formal context model is part of a context-based system. It describes the relationships between objects which are relevant and significant for the current situation.

These kinds of systems are very complex and need extensive information about the situation including the user which shall be supported. The "Significant complexity issues challenge designers of context-aware systems with privacy control" [4, P. 59]. From the legal perspective of the GDPR the user must be able to restrict or even decline the data usage. From a user's perspective he or she wants to decide who will be able to access personal information and when it will be shared or processed. Due to that the privacy control is strongly related to intelligible explanations. That demands a way to explain system processes and data usage, to help users to understand the current situation [5]. "The dynamic aspect of context implies that it is not possible to plan in advance the whole explanatory dialogue" [5, P. 123]. Our understanding of personalized explanations is that they "serve to clarify and make something understandable" [6, P. 498] to the user in a specific situation like the relevance of the GDPR and its consequences to the usage of the system.

According to [7], supporting user friendly intelligible and comprehensive explanations in context-based adaptive systems is a big challenge. They are important for a personalized system to support user acceptance and user trust [6]. Additionally, legal regulations make it necessary to explain data storage and data processing of personal information in a system.

We use a scenario to illustrate the above requirements. Alice uses a context-based adaptive collaboration environment which uses and stores personal information about her in order to, e.g., support personalization. So, we have to answer two questions:

1. Q1: How can Alice agree that her personal information can be stored in and processed by the system and its associated functionalities and services?
2. Q2: How can the "right of access by the data subject" (see Footnote 1) be realized for Alice?

Q1 also includes that every change affecting her agreement, i.e. changing data storage location or usage in the system, invalidates it. Therefore, Alice must accept the changes and agree upon it once again. Q2 implies that Q1 has to be answered as well.

Currently, users cannot retrace the usage and the storage of their personal data in context-based adaptive collaboration environments. According to [5] it is not possible to place explanations to every situation in the system. Furthermore, it is not possible to agree to the usage of personal data for individual functions and applications. In the case of rejection, the entire system can no longer be used.

We address the aforementioned problems and challenges by developing a context-based adaptive collaboration environment supporting user control, comprehensibility and intelligibility. In this paper we present an approach

(1) to give privacy control back to the users according to Q1, and
(2) to create personalized explanations of data usages according to Q2

in our context-based adaptive collaboration environment, based on the CONTact platform (c.f. [3]). We use two typical scenarios ("Compliance by Design" and data usage explanation, cf. Sect. 4) to illustrate our approach consisting of (1) an extended domain model for legal regulations, (2) two process models, and (3) two related rules.

The paper is structures as follows: in Sect. 2 we present related work. We illustrate our extended domain model for legal regulations in Sect. 3, before we use the above scenario to present our answers to Q1 and Q2 in Sect. 4. We discuss our results in Sect. 5. Finally, we present some conclusions and future work in Sect. 6.

2 Related Work

Due to the development of mobile devices and applications as well as the development of personal recommendation systems and intelligent assistants [8], which support users at work or in private areas, many sensitive and personal data sets need to be saved, analyzed and processed. Since many years, researchers realized that the intensive use of sensitive and personal data is a challenge for data privacy. Privacy protection especially concerns the development of personalized application, e.g. collaborative environments, intelligent tutoring systems, (embedded) recommender systems, intelligent assistant systems and mobile assistants in smart devices, cars and even smart cities [9].

So far, research has raised questions concerning the data usage and data processing in systems and techniques mostly from the ethical-moral perspective [10] or from the perspective of supporting user trust [10, 11]. By the GDPR data collection and data usage must be considered also due to the legal necessity [10, 11]. This already applies to the planning and design of a system which is intended to process personal data.

Scientists who work on the design of personalized, adaptive environments focus on the mapping of user and domain-specific aspects. Some of them consider context information to support the users in certain situations. One promising technology on modelling context is ontology [2]. An ontology is a formal specification of a certain domain which describe a set of concepts, relationships and formal axioms that restrict the interpretation of concept instances [12]. The formal concepts can become a common

ground to describe a specific domain which can be shared and reused. Most of the concepts do not consider privacy control or intelligibility of the personal data usage what became so important through the GDPR.

[13] present an approach to support the intelligibility of complex context-aware systems. They point out that intelligibility must be accompanied by a control function for the user. In their work they present an extension of the Context Toolkit. "The Context Toolkit aims at facilitating the development and deployment of context-aware applications."[2] With a programming abstraction they support developers and designers to create explanations to support intelligibility and user control in context-aware applications build by the Context Toolkit. For that, they integrate meaningful explanations in the application Situation by exposing the internal processing of context-aware applications.

Enhancements to the explanation component in the Context Toolkit can generate explanations of the behavior of more popular machine learning techniques and enriched explanations for user control [14, 15]. According to [13] and [14], we consider user control and explanations about the context and the internal processing in our context-based collaboration environment with focus on integration and explanation of external policies. The Context Toolkit and its extensions [13–15] does not reveal any relation to data privacy compliant declarations of data usage and also does not provide information on whether context-based collaborative environments are supported.

Supporting privacy control in context-aware systems is the approach of [4]. They present annotations in information spaces to classify personal and sensitive information. The privacy tagging is used to mark privacy related information that can be identified during processing. The access of a user defined information space is used as a contextual trigger to ask for permission of the owner. The approach support users to get back control on their personal information.

Similar to our approach is the work of [16]. His approach considers the user privacy preferences in context-aware webservices. Therefore, he introduces the policy language Consumer Privacy Language (CPL). The CPL is used to specify the user's privacy preferences, who can insert their privacy setting through a web application. These preferences are considered during the webservice invocation. An adaptation mechanism uses the privacy preferences to get access to context information on a per case basis. The mechanism is integrated in the webservice infrastructure that applies the user's privacy preferences and manages the service execution. [17] extended the privacy module of the Linked Unified Service Description Language (USDL). The privacy module is used to describe privacy policies for the use of any webservice. For that they focus on the service provider and how the provider can communicate the policies considering a service. By using Linked Data they provide the opportunity to link policies and place them in context. The extension can use and include existing privacy policies to answer questions about what personal data is collected from the user, what the service provider does with the collected data and to whom it will shared. The approaches of [16, 17] focus on supporting privacy of user while using webservices. An interesting aspect is the separation of private and non-private data on the conceptual layer. Neither [16] nor [17] describe if and how to support an integrated collaborative environment and so they do not consider the requirements of a personalized collaboration environment. They also do not present

[2] http://contexttoolkit.sourceforge.net.

how a user can get access to a personalized explanation of stored and processed data in the system. In [17] the authors do not describe how users can accept or decline the data usage for certain applications or services and what consequences are related to it. Our approach considers, that users can make decisions about the data processing (accept or decline). For that, we integrate external policies, which are important for the situation, in the context and analyze which of the policies must applied in the specific situation.

Privacy and privacy control come along with intelligible explanations. Explanations are needed to help users to understand why and how their data is used in the system and to whom it will be accessible [18].

[19] present a generic four-layer framework for modelling context in a collaboration environment, a generic adaptation process, and a collaboration domain model for describing collaboration environments and collaboration situations. [3] implements the framework, using an extended domain model and the related adaptation process. The resulting CONTact platform is able to sense and formalize users' interaction with the system at runtime, and to adapt according to the user's current collaboration situation. These adaptations may confuse users. Therefore, [20] use context enriched explanations to help them understand the adaptation behavior. [3] and [20] take the aspects of the comprehensibility of system behavior, decisions and data processing into account, but do not satisfy the legal requirements. Furthermore, the explanations provided are not presented in a way that is intelligible to the users. So far, there are no known context-based collaborative systems that support comprehensibility and intelligibility for users.

No approach is known to us for context-based collaborative systems that considers the requirements of the GDPR and taking up the topic Compliance by Design.

3 Domain Model: Legal Regulations

In this section we introduce the domain model and explain its concepts and relationships. We used the OWL 2 Web Ontology Language (informally OWL 2) and the Protégé Ontology Editor for modelling. In this paper we focus on concepts of user control, comprehensibility and intelligibility by considering the legal requirements.

3.1 Context Modelling

We use our approach presented in [19] consisting of the generic four-layer framework for modelling context in a collaboration environment and the related collaboration domain model for describing collaboration environments and collaboration situations. The framework contains the knowledge layer, the state layer, the contextualized state layer and the adaptation layer [19]. The knowledge layer describes a domain model with abstract (e.g. classes, properties) and concrete (e.g. individuals) predefined knowledge, mapped to corresponding concepts and relations. The state layer uses sensing rules to instantiate related concepts and relationships from the domain model (cf. knowledge layer) to represent the current collaboration environment of all users. The contextualized state layer applies contextualization strategies to extract a subset from the state (cf. state layer) and/or domain model (cf. knowledge layer) which are relevant for the current collaboration situation. This creates a contextualized state (the context). The adaptation

layer evaluates the adaptation rules and executes applicable adaptation rules. This leads to the adapted state that is mapped to the collaboration environment.

To address the GDPR, we extended our domain model (cf. [19]). Figure 1 shows an excerpt of our resulting ontology (i.e. domain model and relevant instances from the state required to illustrate our approach). For readability reasons, we omitted concepts and relationships, and focused on the concepts, relationships, and instances helping to describe situations, when user control, comprehensibility and intelligibility is needed. Therefore, we use Alice who has already created an account in the app meinDorf55+ (a novel community support system for elderly) which demands personal information.

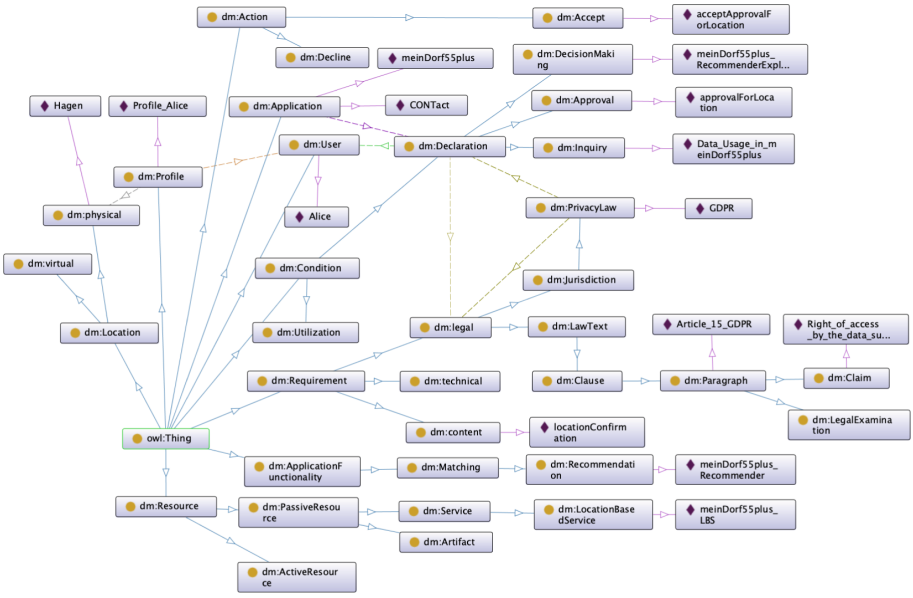

Fig. 1. Ontology representing legal and comprehensibility concepts and relations

As Fig. 1 shows, *Alice* is an instance of the concept *dm:User* in our sample ontology. *CONTact* (representing the CONTact platform) and *meinDorf55plus* (representing the novel community support system for elderly) are instances of the concept *dm:Application*. The concept *dm:Profile* is related to *dm:User* and includes the address represented as *dm:physical* that is a subclass of *dm:Location*. Despite physical locations we support *dm:virtual* as a subclass of *dm:Location*, e.g. to support URLs. Applications usually provide different kinds of functionalities. We map these to related concepts of *dm:ApplicationFunctionality* and *dm:Resource* when modelling the related opportunities. The concept *dm:Resource* can be either *dm:PassiveResource* or *dm:ActiveResource*. A *dm:PassiveResource* can be split up into a *dm:Service* and *dm:Artifact*.

Using the presented concepts and relationships we can create instances in our ontology representing related situations in our context-based adaptive collaboration environment. In case of Alice that means that *Alice* is an instance of *dm:User* and

Profile_Alice is an instance of *dm:Profile*. As soon as Alice tries to use a special *dm:ApplicationFunctionality* of *meinDorf55plus*, e.g. *meinDorf55plus_LBS* as an instance of *dm:LocationBasedService* or *meinDorf55plus_Recommender* as an instance of *dm:Recommendation*, she has to provide her address (in our ontology *Hagen* as an instance of *dm:physical*). This is when the legal regulations have to be represented in our domain model.

Figure 1 shows the concepts *dm:Requirement, dm:Condition* and *dm:Declaration* and their dependencies which are used for adaptation, user control and comprehensible explanations to the users. The aforementioned concepts are used to answer the questions *What happened?* (*dm:Requirement*), *Why does it happened?* (*dm:Condition*) and *What kind of explanation should be provided?* (*dm:Declaration*) in the specific context.

3.2 Concept dm:Requirement

Requirements are conditions for applications and define what an application (*dm:Application*) or application functionality (*dm:ApplicationFunctionality*) must check and take into account during processing. The requirements are no fixed set of rules instead they are used at runtime to find out what the application has to do in the current situation. Therefore, requirements can be seen as external policies which must be considered by an application (we use the term rule to illustrate that a related policy can be implemented in our CONTact platform). Requirements can be technical conditions (*dm:technical*), content definitions (*dm:content*) and legal regulations (*dm:legal*). These three aspects are separate domain models that are subordinated to the concept *dm:Requirement*. Technical requirements can be hardware resources that limit the execution of certain functionalities, e.g. by using mobile devices with less powerful hardware. The application has to react to this, e.g. by organizing a provision via other devices (e.g. by computing on servers). Content definitions can result from the domain of an application or a service. Figure 1 shows *locationConfirmation*, a content related requirement of the *dm:Application* instance *meinDorf55plus* that provides a location-based service (*dm:LocationBasedService*) represented in the instance *meinDorf55plus_LBS*. For that *meinDorf55plus_LBS* needs a conformation of the users location which is requested by the instance *locationConfirmation* of the concept *dm:content*.

The ontology shows an excerpt from the legal domain model *dm:legal*. It describes the German jurisdiction by depicting its taxonomy as part of the concept *dm:Jurisdiction*. The law taxonomy has different legal areas, e.g. the privacy law or the civil law. For readability reasons, Fig. 1 only contains the privacy law (*dm:PrivacyLaw*). The instance *GDPR* of the concept *dm:PrivacyLaw* represents the applicable law. Furthermore, the legal domain model depicts the general structure of the legal texts through the concept *dm:LawText* including its clauses (*dm:Clause*) and paragraphs (*dm:Paragraph*). The instance *Article_15_GDPR* of the concept *dm:Paragraph* is used to identify the claim. The instance *Right_of_access_by_the_data_subject* of the concepts *dm:Claim* represents the claim which is derived from Article 15 (*dm:Paragraph*). A paragraph can either represent a claim (*dm:Claim*) or an explanation of the right (*dm:LegalExplanation*). Both determine the activities of an application.

3.3 Concept dm:Condition

Conditions are derived from the requirements. The concept *dm:Condition* is intended to verify the correctness and legitimacy of the processing. The legitimacy arises, e.g. from the legal regulations of privacy law like in Fig. 1. Conditions are a set of abstract rules which are defined in the application to map the external requirements to the application processing. At development time not all rules are known, so they are based on the concepts and domain models of *dm:Requirement* for the specific purpose of the application. The rules have the form *WHEN condition part THEN action block*. At runtime, the application uses these constructs to check which situation it is in, which actions has to be executed, and which conditions must be fulfilled for continuing processing. The conditions, on their part, can trigger a cascade of checks that are given on the basis of the requirements of the respective domain *dm:technical*, *dm:content* or *dm:legal*. In Fig. 1 the *locationConfirmation* caused a check of legal requirements that results in the creation of different kinds *of dm:Declaration* instances.

The condition for the use of certain application functionalities (*dm:ApplicationFunctionality*) maybe also be motivated from a legal perspective. Thus, a direct interaction with the user is maybe not necessary (e.g. encrypted data transmission). The concept *dm:Utilization* of the domain model can be used for that kind of required functionality.

3.4 Concept dm:Declaration

Declarations are the interface to users which can support comprehensibility and user control. As shown in Fig. 1, the provision of an explanation depends on the requirements (e.g. legal regulations). According to Article 15 of the GDPR, data subjects whose data are collected and processed have a right to obtain information about the usage. This includes the purposes of the processing, the categories of personal data processed, the recipients to whom the data are disclosed, the duration of the storage, the existence of a right of appeal and an overview of the origin of the data, if not collected from the data subject. In addition, Article 15 declares, the data subject has the right to limit the processing by the data processor. Furthermore, a right of objection against the processing exists at any time.

Addressing Q1 and Q2, our domain model contains the concept *dm:Declaration* to be able to represent the right to obtain information about data usage. Depending on the current context the concept *dm:Declaration* is used to provide comprehensible explanations (*dm:Inquiry*), demand an approval (*dm:Approval*) or to explain processing (*dm:DecisionMaking*). The user can accept (*dm:Accept*) or decline (*dm:Decline*) the usage of his/her data by the system through an approval (answering Q1). Approvals are needed to execute an action (*dm:Action*) and depend on the requirements for the application, e.g. when personal information shall be transmitted to a third party it must be approved by the user. Figure 1 shows the instance *approvalForLocation* of the concept *dm:Approval*, which is needed to approve the usage of the users location by himself or herself for the content requirement *locationConfirmation*. Accepting it leads to the creation of the instance *acceptApprovalForLocation* of *dm:Accept* which stores all relevant information to the approved data usage.

The instance *Right_of_access_by_the_data_subject* of the concept *dm:Claim* caused the creation of the *dm:Inquiry* instance *Data_Usage_in_meinDorf* (cf. Q2). Information must also be provided on whether and how automated decision-making, including profiling, takes place. According to Article 22[3] (1) and (4), meaningful information on the logic involved, the significance and the intended impact of such processing for the data subject must be provided. This requirement is considered separately in the domain model through the concept *dm:DecisionMaking*. It is used when application functionalities for decision-making, such as a personalized recommendation (*dm:Recommendation*), is performed based on user data. The instance *meinDorf55plus_RecommenderExplanation* (answering Q2) of *dm:DecisionMaking* results from the instance *meinDorf55plus_Recommender* of the concept *dm:Recommendation* which is a subclass of *dm:Matching*.

4 Scenarios

As illustrated in the above sections, collaboration environments have to support explanations where and for what purpose personal data is being processed. We use the above scenario to illustrate our rule-based approach of supporting 'Compliance by Design', i.e. giving users control over their personal data being processed by our CONTact platform (cf. Q1). The second scenario describes how we use our formal context model for creating explanations to support the aforementioned mandatory feature (cf. Q2).

In Fig. 2 we present the scenario 'Compliance by Design' where we attempt to give users control over their personal data being processed by related applications.

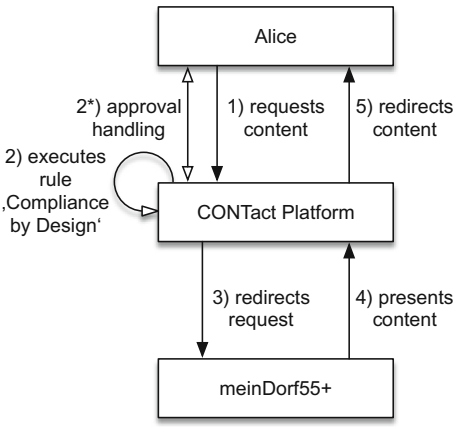

Fig. 2. Process of user interactions – Compliance by Design

The user (in our scenario Alice) requests content from the corresponding application 'meinDorf55+' through the CONTact platform (cf. (1–3)). After receiving the request,

[3] https://gdpr-info.eu/art-22-gdpr/.

the CONTact platform checks whether or not personal data will be processed (i.e. interpreting related annotations at source code level). When personal data will be processed, the rule (cf. (2)) presented in Listing 1 is executed to ensure that user (in our scenario Alice) has approved the usage of her/his personal data (represented as white arrows in Fig. 2). When the user has not authorized the data usage previously, she/he will be prompted to do so (cf. (2*)). After approving the data usage, the related request will be processed, and the content will be presented to the user (cf. (3–5)).

```
rule "Compliance by Design"
  when
    user: getUserInContext("dm:User")
    app: getApplicationInContext(user, "dm:Application")
    req: getRequirementInContext(app, "dm:Requirement")
    appr: requestApproval(user, app, req)
  then
    createOrUpdateAcceptedApproval(appr)
    notify(user, appr)
  end
end
```

Listing 1 Rule "Compliance by Design"

Listing 1 uses pseudocode to illustrate our approach to implement “Compliance by Design”. The rule consists of a condition part (`when` to `then`) and an action block (`then` to `end`). `getUserInContext` retrieves the user interacting with the CONTact platform (in our scenario Alice). The function `getApplicationInContext` determines the application used by the user which is of type *dm:Application* (in our scenario meinDorf55plus). The function `getRequirementInContext` retrieves all instances and relations connected to the domain concept *dm:Requirement* of the given application. The function `requestApproval` uses the context information about the user, the application and the requirement and ensures that the user has approved the data usage. We distinguish two different situations:

(I) When the user approved the data usage beforehand, the return value of the function is empty.
(II) When there is no or an inapplicable approval instance present in the current context, the approval is requested from the user.

   a. When the user declines the data usage, the return value of the function is empty.
   b. When the user accepts the data usage, the approval information will be returned.

The action part of the above rule is executed as soon as all conditions are met (i.e. the returned information are not empty). First, we create or update the approval instance in the current context. Next, we notify the user about it. This shows our answer of Q1.

Figure 3 shows the process of creating explanations of data usages. In our sample scenario Alice wants to know, where and how her personal data is being processed. She requests the information about data usage from the CONTact platform (cf. (1)).

The CONTact platform redirects the request to the application 'meinDorf55+' (cf. (2)) where the data is stored and used. The related application has all the information about the requested data usage and reports it to CONTact platform (cf. (3)). Based on the returned data the CONTact platform checks which conditions are affected and requests an inquiry from the Requirements Handler (cf. (4) & (8)), that determines which legal requirements are affected and requests related templates from the Explanation Template Builder (PrivacyLaw) (cf. (5–7)). The CONTact platform uses the inquiry (cf. (8)) and reported data usage (cf. (3)) and executes the rule 'data usage explanation' (cf. Listing 1) to create a personalized explanation about Alice's data usage and present it to her (cf. (9–10)).

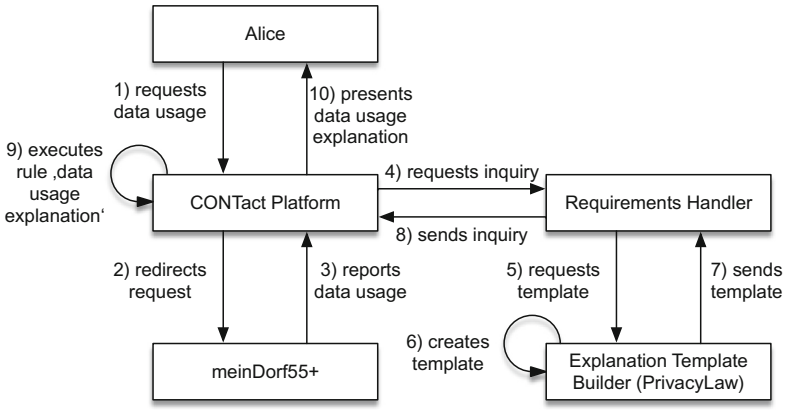

Fig. 3. Process of request and create an explanation of data usage

Listing 2 illustrates the creation of a personalized explanation about the data usage of a user (in our scenario Alice). In Listing 2, `getUserInContext` retrieves the user interacting with the CONTact platform (in our scenario Alice). The function `getApplicationInContext` determines the application used by the user which is of type *dm:Application* (in our scenario meinDorf55plus).

```
rule "data usage explanation"
  when
    user: getUserInContext("dm:User")
    app: getApplicationInContext(user, "dm:Application")
    req: getRequirementInContext(app, "dm:legal")
    tmpl: getExplanationTemplate(user, app, req)
    expl: createExplanationInContext(tmpl)
  then
    present(user, expl)
  end
end
```

Listing 2 Rule "data usage explanation"

The function `getRequirementInContext` retrieves all instances and relations connected to the domain concept *dm:legal* of the given application. `getExplanationTemplate` uses the context information about the user, the application and the requirement and generates the related explanation template that the function `createExplanationInContext` uses to create the related explanation. The action part of the above rule is executed as soon as all conditions are met (i.e. the returned information are not empty). After applying the rule to the CONTact platform, the personalized explanation is presented to the user (e.g. Alice). This illustrates our answer of Q2.

5 Discussion

The presented domain model shows the connection between applications and legal regulations in the context-based adaptive collaborative environment. The CONTact platform can be linked to or integrated into different kinds of collaboration environments that was illustrated with the novel community support system meinDorf55+. We explained the process of checking legal regulations from the privacy law GDPR while using a service that requires personal information about the user.

This paper does not cover some outstanding aspects. (I) Due to the limitation of the paper we could not explain in detail the connection of external policies with the application by the concept *dm:Condition*. (II) The presented extended domain model is only an excerpt. We focused on describing only specific concepts of the jurisdictions in it. (III) The extended domain model is only a basis for comprehensibility and personalized intelligible explanation of system processes. Users should be able to understand why something happens and how it happens in a personalized, adaptive system. The challenges of the comprehensibility of system processes includes their presentation and intelligibility. Presenting only technical information is not sufficient [7, 15]. In context-aware systems explanations "need to have access to information about complex real-world concepts that are not necessarily core to the application" [15, P. 166]. The mentioned explanation building process (cf. Fig. 3) is responsible for creating personalized explanations, e.g. when legal regulations demand it. The legal concepts of the domain model can be used to support intelligible legal explanations by the system. The intelligibility can be facilitated by the deposit of target-group-specific texts (e.g. texts created by experts) and explanations through integrated and linked dictionaries. The resulting templates could be used to provide explanations at runtime by creating instances of the concept *dm:Declaration*. (IV) While context-based adaptive collaborative environment needs adaption rules this paper does not cover it. Regarding to our four-layer context model [18], we modeled our domain model independently from adaptation rules.

6 Conclusions and Future Work

In this paper we presented an approach that enables us to answer the two questions how a user can agree that her/his personal information can be stored in and processed by the system, and how the "right of access by the data subject" (see Footnote 1) can be realized for a user. We used a sample scenario where Alice uses a context-based adaptive collaboration environment based on the CONTact platform which uses and

stores personal information about her in order to support personalization. We illustrated when the two questions, mentioned above, get relevant.

Our approach is based on three main contributions: 1. an extended domain model for legal regulations (cf. Fig. 1) and 2. two processes (cf. Figs 2 and 3) and 3. related rules (cf. Listings 1 and 2). This enables us to support user control, comprehensibility and intelligibility in a context-based adaptive collaboration environment, based on the CONTact platform. We introduced the domain model and explained its concepts (especially the concepts *dm:Requirement, dm:Condition* and *dm:Declaration*) to facilitate user control, comprehensibility and intelligibility. For readability reasons we omitted other concepts and relationships that are part of our domain model, e.g., considering legal regulations other than privacy laws.

We presented our approach to give privacy control back to the users (answer to Q1), and to create personalized explanations of data usages (answer to Q2) in our context-based adaptive collaboration environment. We used the above scenario to illustrate our rule-based approach of supporting 'Compliance by Design', i.e. giving users control over their personal data being processed by our CONTact platform, and how we use our formal context model for creating personalized explanations of data usages. This helps us observing legal regulations, e.g. privacy laws like the GDPR.

We argue that the presented approach answers the above questions, but also that it does not represent the developed approach in full detail. For readability or space reasons we presented only an excerpt of our developed domain model, i.e. we have to omit the other concepts and relationships for legal regulations, and, e.g., the representation of external policies of an application using the concept *dm:Condition*.

In the next step we will investigate the challenges of comprehensibility including presentation and intelligibility. Thereby, we have to support related explanations when the user has to approve the usage of her/his personal data (cf. Fig. 2). This will lead to a combination of the two processes and related rules presented in this paper. Furthermore, we have to investigate, how the explanations have to be created, personalized and presented so that users are able to understand the meaning of the presented text and the consequences of accepting or declining the approval. This will mainly influence the presented explanation builder process (cf. Fig. 3).

Acknowledgment. The project is supported (was supported) by funds of the Federal Ministry of Food and Agriculture (BMEL) based on a decision of the Parliament of the Federal Republic of Germany via the Federal Office for Agriculture and Food (BLE) under the rural development programme.

References

1. Dey, A.K.: Understanding and using context. Pers. Ubiquit. Comput. **5**(1), 4–7 (2001)
2. Abarca, M.G., Alarcon, R.A., Barria, R., Fuller, D.: Context-based e-Learning composition and adaptation. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006. LNCS, vol. 4278, pp. 1976–1985. Springer, Heidelberg (2006). https://doi.org/10.1007/11915072_106
3. Veiel, D., Haake, J.M., Lukosch, S., Kolfschoten, G.: On the acceptance of automatic facilitation in a context-adaptive group support system. In: 2013 46th Hawaii International Conference on System Sciences, pp. 509–518. IEEE (2013)

4. Jiang, X., Landay, J.A.: Modeling privacy control in context-aware systems. IEEE Pervasive Comput. **1**(3), 59–63 (2002)
5. Brezillon, P.J.: Contextualized explanations. In: Proceedings of International Conference on Expert Systems for Development, pp. 119–124. IEEE (1994)
6. Gregor, S., Benbasat, I.: Explanations from intelligent systems: theoretical foundations and implications for practice. MIS Q. **23**(4), 497–530 (1999)
7. Dey, A.K.: Explanations in context-aware systems. In: Proceedings of the Fourth International Conference on Explanation-Aware Computing (EXACT 2009), pp. 84–93 (2009)
8. Tahir, H., Brézillon, P.: Contextual graphs platform as a basis for designing a context-based intelligent assistant system. In: Brézillon, P., Blackburn, P., Dapoigny, R. (eds.) CONTEXT 2013. LNCS (LNAI), vol. 8175, pp. 259–273. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40972-1_20
9. Nigon, J., Verstaevel, N., Boes, J., Migeon, F., Gleizes, M.-P.: Smart is a matter of context. In: Brézillon, P., Turner, R., Penco, C. (eds.) CONTEXT 2017. LNCS (LNAI), vol. 10257, pp. 189–202. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57837-8_15
10. Bohn, J., Coroamă, V., Langheinrich, M., Mattern, F., Rohs, M.: Living in a world of smart everyday objects—social, economic, and ethical implications. Hum. Ecol. Risk Assess. **10**(5), 763–785 (2004)
11. Wachter, S.: Normative challenges of identification in the Internet of Things: privacy, profiling, discrimination, and the GDPR. Comput. Law Secur. Rev. **34**(3), 436–449 (2018)
12. Guarino, N.: Formal ontology in information systems. In: Proceedings of the First International Conference (FOIS 1998), Trento, Italy, 6–8 June 1998, vol. 46. IOS Press (1998)
13. Dey, A.K., Newberger, A.: Support for context-aware intelligibility and control. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 859–868. ACM (2009)
14. Lim, B.Y., Dey, A.K.: Toolkit to support intelligibility in context-aware applications. In: Proceedings of the 12th ACM International Conference on Ubiquitous Computing, pp. 13–22. ACM (2010)
15. Lim, B.Y., Dey, A.K.: Design of an intelligible mobile context-aware application. In: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, pp. 157–166. ACM (2011)
16. Kapitsaki, G.M.: Reflecting user privacy preferences in context-aware web services. In: 2013 IEEE 20th International Conference on Web Services, pp. 123–130. IEEE (2013)
17. Kapitsaki, G., Ioannou, J., Cardoso, J., Pedrinaci, C.: Linked USDL privacy: describing privacy policies for services. In: 2018 IEEE International Conference on Web Services (ICWS), pp. 50–57. IEEE (2018)
18. Bellotti, V., Sellen, A.: Design for privacy in ubiquitous computing environments. In: de Michelis, G., Simone, C., Schmidt, K. (eds.) Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13–17 September 1993, Milan, Italy ECSCW 1993, pp. 77–92. Springer, Dordrecht (1993)
19. Haake, J.M., Hussein, T., Joop, B., Lukosch, S., Veiel, D., Ziegler, J.: Modeling and exploiting context for adaptive collaboration. Int. J. Coop. Inf. Syst. **19**(01n02), 71–120 (2010)
20. Hussain, S.S., Veiel, D., Haake, J.M., Lukosch, S.: Facilitating understanding of team-based adaptation policies. In: 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010), pp. 1–8. IEEE (2010)