Considering Legal Regulations in an Extendable Context-based Adaptive System Environment

Mandy Goram[a] and Dirk Veiel[b]
Faculty of Mathematics and Computer Science, FernUniversität in Hagen, 58084 Hagen, Germany

Keywords: Context-based Adaptive Environment, Personalized Explanation, Compliance by Design, Copyright Law.

Abstract: Legal regulations demand that applications consider legal aspects of the application domain. Regulations equally concern software designers, developers, legal experts, providers and users. Not only the application must be legally compliant, the users must also comply with the law when using the application. Therefore, it is important to explain the user the current situation and the related consequences to the usage of the system. But it is a big challenge to support users with explanations of the law and the related actions and consequences to the usage of the system. We address the aforementioned challenges by developing an extendable context-based adaptive system environment, which considers legal policies and generate personalized explanations for users. This paper presents an approach to integrate legal regulations into context-based systems and an excerpt of our legal domain model. We describe a process on how legal experts can configure the adaptive interaction with the domain-specific application and the generating of personalized explanations. For that, we use a sample collaboration situation when Copyright Law and personalized explanations get relevant.

1 INTRODUCTION

The development and usage of legal compliant software applications has become an important aspect. To prove and check its legal conformity it is important to know what an application is doing. That must be explained to the involved people and regulators. Legal regulation like the General Data Protection Regulation (GDPR) demands the explanation of data usage and the compliant processing of personal information to the users. Additionally, there is a right of the subjects to get an overview of the processing and data usage of any system that stores and processes user information (https://gdpr-info.eu/art-15-gdpr/). Every application which processes user data must consider the GDPR. Therefore, a classification of user data is necessary. To classify user data, (Kapitsaki et al., 2018) presents a formal model to separate sensitive and non-sensitive data for webservice requests of business applications. But there must be more considered than the data classification. The GDPR is a complex regulation with relationship to other regulations, which must be considered too. Therefore (Bartolini et al., 2015) developed an ontology-based legal domain model that represents the regulations from the regulatory text in a formal model as ontology. An ontology is a formal specification of a certain domain which describes a set of concepts, relationships and formal axioms that restrict the interpretation of concept instances (Guarino, 1998). Legal knowledge engineering tries to establish ontologies and formalizes norms and legal subsumptions (Baumann et al., 2010) to reuse them (Gangemi, 2007), what is difficult because of the different scenarios.

The design and development of software which considers legal requirements needs a tight collaboration between software designers, developers and legal experts (who check and confirm legal compliance). Software providers must be assured that the applications they use and serve to customers or users runs compliant to the law. For that it is also important to support users to act according to the law through explanations of the situation like content upload which concerns the Intellectual Property Law (IP Law). This aspect is relevant to the provider, because he or she is responsible for users' legal breaches. But legal regulations are interconnected,

[a] https://orcid.org/0000-0003-0264-841X
[b] https://orcid.org/0000-0003-0228-103X

Goram, M. and Veiel, D.
Considering Legal Regulations in an Extendable Context-based Adaptive System Environment.
DOI: 10.5220/0009565003670376
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 2*, pages 367-376
ISBN: 978-989-758-423-7

even if there is no clear connection at first sight. In order to ensure legal conformity, it is therefore not sufficient to simply include a specific legal requirement in a system (Baumann et al., 2010), (Casellas et al., 2010).

For illustration purposes, we will use a sample scenario in this paper in which the user Alice wants to upload the photo of a group trip in a community application, as a supplement to a report. Since it is a group photo, she must fulfill the legal requirements of Copyright Law in order to publish it in the community application. The legal requirements include, among other things, a written consent of all persons depicted in the photo. Alice is only allowed to make the photo accessible to third parties within the agreed scope (the legal situation is simplified) when these are available. The consequences of any violation are regulated in Copyright Law and other laws (e g GDPR) and the personal rights.

The well-defined domain model in (Delgado et al., 2003) considers related legal regulation and add to the IP Law also the subordinated Copyright Law and the Exploitation Right to the ontology. The regulatory model is used to develop applications that deal with multimedia interchange. Other related law like the personal rights or the data privacy is not contained in the domain model of (Delgado et al., 2003). More sophisticated applications may lead to different situations, as illustrated in Alice scenario. In that simplified case the system must be able to sense and also know, e. g., that the action concerns the personal right and the data privacy too.

To use an application according to the law, the user needs an explanation about the current situation. Therefore, users should be made aware of legal aspects and be supported in complying with them that relate to their actions, e. g. Alice scenario. Existing approaches neither support users in specific situations nor explain the processing, actions or data usage. To support users in certain situations (e. g. Alice) the system must be aware of the user's situation and the related socio-technical environment, i. e. the context. According to (Dey, 2001), the context contains any information that can be used to describe a situation within a socio-technical system. We extend the definition of (Dey, 2001) and also consider all related law which is relevant in and for a specific situation. Context-aware systems are used to support personalization with regard to the current situation of related users.

Explanations are important for intelligent personalized systems to support user acceptance and user trust. According to (Dey and Newberger, 2009), the privacy control is strongly related to intelligible explanations. Therefore, it is necessary to explain system processes and data usage, to help users to understand the current situation (Brezillon, 1994). "The dynamic aspect of context implies that it is not possible to plan in advance the whole explanatory dialogue" (Brezillon, 1994, P. 123). Personalized explanations shall "serve to clarify and make something understandable" (Gregor and Benbasat, 1999, P. 498) to the user in a specific situation. If a user's action, e. g. affects the GDPR or the Copyright Law, the consequences of the usage of the system should be explained.

To provide this support a more sophisticated system approach is necessary like a context-based adaptive system environment which is used to provide personalized support in the current situation. To address the aforementioned aspects, we have to answer two questions:

Q1: How to design a context-based adaptive software application which considers the concerning law?

Q2: How to explain users what they must do to act in compliance to legal regulations in the specific situation?

We are implementing an intelligent extendable context-based adaptive system environment (eCBASE) to develop or integrate context-based applications and integrate technical, content and legal requirements in a common domain model. For that we use the knowledge of domain experts (technical, content, legal) to define sets of rules for the specific application. In case of legal regulations, the context-based adaptive environment must know when and how it must do something. For that, a legal expert must also predefine what needs to be considered and what action should be executed by the resulting context-based adaptive application. In this paper we present an approach on how the knowledge and explanations of a legal expert can be integrated into eCBASE and a context-based adaptive application. For that we describe two sample situations that regards the GDPR and the Copyright Law. We illustrate the integration of the legal regulation and the generating of explanations to users.

The paper is structures as follows: in section 2 we present related work. Section 3 explains our process of integrating legal regulations into our four-layer context model. In section 4 we describe our legal domain model and a sample of integrating aspects of the copyright law for a specific application. Generating personalized explanations is described in section 5. We discuss our results in section 6. Finally, we present some conclusions and future work in section 7.

2 RELATED WORK

There are many approaches in the area of privacy support for personalized systems. (Jiang and Landay, 2002) presents one that support privacy control in context-aware systems. They use annotations on source code level to classify personal and sensitive information for so-called information spaces. Privacy related information is marked with privacy tags which are used to identify the privacy information during processing. If anyone different than the owner access an information space, a contextualized trigger asks for permission of the space owner. The approach support users to get back control on their data.

(Bartolini et al., 2015) shows a coarse ontology-based legal domain model of the GDPR designed by a legal expert. The legal domain model contains the regulation with the aspects of the basic data protection principals, the data processing rules, and the data subject's rights. They use the ontology to assist data controllers during the development cycle of software. For that, they integrated the ontology in a Business Process and Model Notation (BPMN) workflow to express the GDPR requirements within the workflow. The GDPR is a relevant legal domain in our approach. The data protection ontology of (Bartolini et al., 2015) was designed by a legal expert. The aspects of executable rules for an adaption and the explanation of a situation and the data usage during application runtime are not covered by (Bartolini et al., 2015).

The work presented in (Kapitsaki, 2013) considers user privacy preferences in context-aware webservices using Simple Object Access Protocol SOAP messages as adaptation mechanism. The paper 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. The user's preferences are considered during the webservice invocation. Through an adaptation mechanism the privacy preferences are used to get access to context information on a per case basis. The mechanism is integrated into the presented webservice infrastructure that applies the user's privacy preferences and manages the service execution. The approach of (Kapitsaki, 2013) is tailored to support the users by (Kapitsaki et al., 2018). They enhanced and extended the approach with focus on the provider side with aspects of the policy communication to any business service. They 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. 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 users, how the service provider uses the collected data and with whom it will be shared. The approaches of (Kapitsaki, 2013) and (Kapitsaki et al., 2018) focus on supporting privacy of users while using webservices. An interesting aspect is the separation of private and non-private data on the conceptual layer. Neither (Kapitsaki et al., 2018) nor (Kapitsaki, 2013) describe how to support a community application.

While using community applications or systems, also the Copyright Law or IP Law has got attracted researches and technicians to develop compliant architectures and approaches. A more sophisticated approach using a regulatory ontology is presented by (Delgado et al., 2003). They are using ontologies to get a more flexible and modular system that allows to extend and adapt the concepts and relationships without losing the connection to the base system. They describe the regulatory Intellectual Property Rights ontology (IPROnto), allowing the combination of results from different areas or domains. The IPROnto contains a static and dynamic view to separate the design of the intellectual property regulation and its derived rights from the using of the defined rules and interdependencies. The static view describes legal concepts (defined by law), legal entities (possessing capacity in law) and the IP Law in detail. The dynamic view is used to create business models for its use in e-commerce applications. It contains events which represent the processes of the intellectual property right. For that they describe a content life cycle and sample events like the creation of origin content and its related law and rights of the legal entities. At least they describe how they can develop and transfer their design on an ontology for the GDPR based on their IPROnto framework.

The design and development of legal decision support systems created a research area on the formalization of the law. The main aspects are the legal support of lawyers and advise seeking persons, to get direction of the legal subsumption of a specific case. Some legal support systems use an ontology-based approach to represent and formalize the knowledge about the legal regulation (Baumann et al, 2010), (Gangemi, 2007), (Casanovas et al, 2016), (Gangemi et al., 2003). But most of them do not consider intelligible explanations and actions in their formal description. Therefore, it is not possible to explain the user what happened and why something happened in a specific situation.

To support privacy and privacy control it is important to provide intelligible explanations. Explanations are needed to help users to understand why and how their data is used in the system and who has access (Bellotti and Sellen, 1993).

Supporting intelligibility of complex context-aware systems is the approach of (Lim and Dey, 2010). They point out that intelligibility must be accompanied by a control function for the user. Their work focus on an extension of the Context Toolkit. “The Context Toolkit aims at facilitating the development and deployment of context-aware applications” (http://contexttoolkit.sourceforge.net). The extension support developer and designer who use the Context Toolkit to integrate intelligible explanations and user control while building a context-aware application. 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 are presented in (Lim and Dey, 2011). They generate explanations of the behaviour of more popular machine learning techniques and enriched explanations for user control (Lim and Dey, 2010), (Lim and Dey, 2011). It is not known to us that the Context Toolkit supports context-based adaptive community applications as well as legal regulations.
(Haake et al, 2010) 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. (Veiel et al., 2013) implement 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. Applying adaptations at runtime may confuse users.

In conclusion, most of the related work mentioned above describe the integration and the scope of data protection in a specific area but do not take other regulations into account. The development of a system transferable to other areas requires a common formal language (Delgado et al., 2003). (Kapitsaki et al., 2018) tries to put this approach on a broader basis with CPL and an extended privacy module of the USDL. The formal classification of the data categories described in (Kapitsaki et al., 2018) is a central information and basis for further processing. They mentioned the application of the Webservice Framework to copyright. But there is no information available on how the copyright law is supported.

Although the approaches mentioned address legal requirements, they do not or only marginally consider the provision of explanations to the user. The focus mostly is on the system architecture or on the creation of a specific legal taxonomy and not on the users. To support them, especially in the understanding of automated procedures, the system has to be designed to meet the users' needs. Therefore, we integrate legal requirements and explanations to all relevant objects of an application in a way to support users in specific situations (e. g. collaboration situation) with suitable explanations. In this paper, we explain how a context-based system could be designed to do so.

3 PROCESS OF INTEGRATING LEGAL REGULATION IN CONTEXT-BASED APPLICATIONS

From our point of view, only a legal expert of the application domain can evaluate which legal regulations are relevant for an application. Therefore, we propose a process to integrate legal policies and appropriate actions into the domain model of a domain-specific context-based (collaboration) application as shown in Fig. 1. The domain-specific application based on the core system eCBASE. Good and best practices can then be integrated and provided through eCBASE. For that, eCBASE contains a legal domain model (cf. 4) to represent the legal domain and to define legal required actions and explanations.

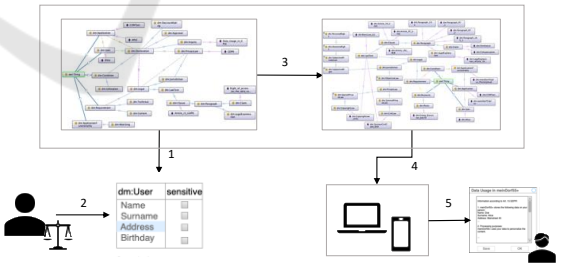

Figure 1: Process illustration of legal policies integration.

A legal expert has the expertise to identify regulations for the domain-specific application which has to be provided. At the same time, he or she can define policies for dealing with the application as well as the legally compliant behaviour of a domain-specific application in order to comply with the legal regulations. The legal experts are involved in the design of the application, i. e. the domain model (compliance by design). Therefore, they get an

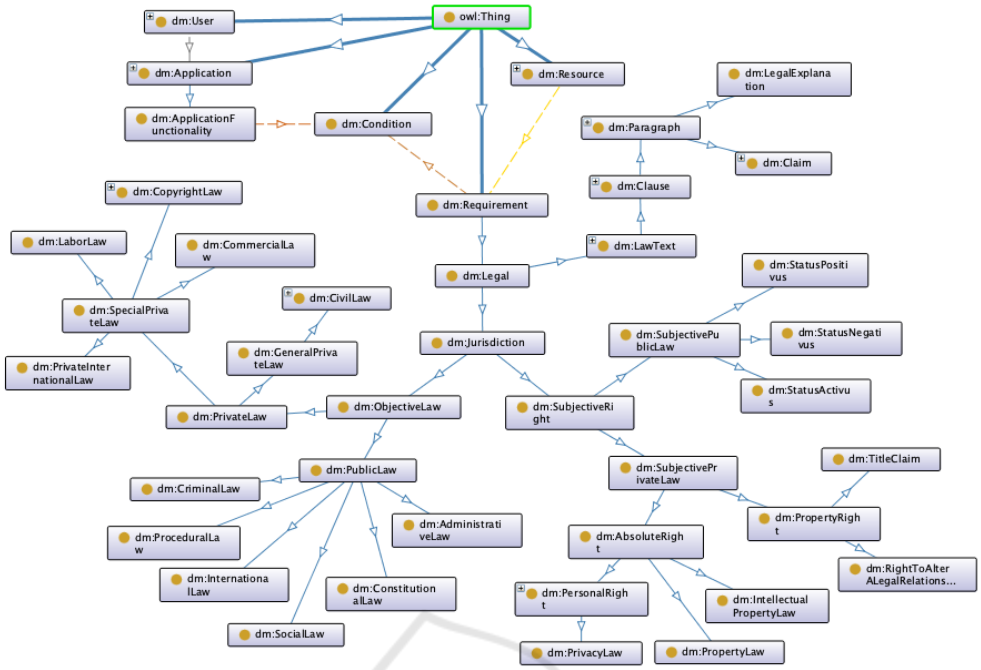

Figure 2: Core concepts of the legal domain model.

overview of the information in the domain model (cf. 1+2 in Fig. 1). They decide which legal conditions apply to the information, e. g. data protection (sensitive data). The defined requirements are used to store the legal regulations, their references to other areas of law and the consequences for data processing in the system. The information is stored in the context model. Corresponding rules and actions are derived or linked to them (cf. 3 in Fig. 1). The legal experts also define the templates which represent the legal explanations. The templates are used to generate personalized explanations for users (cf. 4+5 in Fig. 1).

4 CONSIDERING LEGAL REGULATIONS

This section introduces the legal domain model of eCBASE and explains its concepts and relationships. We briefly introduce the overall approach before we present the legal domain model. Due to space reasons, we omit core concepts and relationships which represent technical and non-legal concepts. We presented the core model in (Goram and Veiel, 2019).

4.1 Context Modelling Approach

We use the CONTact platform (Veiel et al., 2013) to implement a context-based adaptive application and decouple the context from the content as much as possible. That makes it possible to reuse our system and to modify it according to required changes of the context model.

For context modelling we use the Web Ontology Language (OWL2) and the generic four-layer framework for modelling context in a collaboration environment and the related collaboration domain model presented in (Haake et al, 2010). The generic four-layer framework consists of the knowledge layer, the state layer, the contextualized state layer, the adaptation layer and related components to implement a generic adaptation process (Haake et al, 2010). 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 sensing engine at 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 contextualization engine at the contextualization 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 engine at 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.

4.2 Concepts of German Jurisdiction Domain Model

In order to support legal regulations in the context model, we have to extend the domain model presented in (Haake et al, 2010). The German jurisdiction is complex and may be connects different law areas for one case. Regulations, like the GDPR, which concerns all software applications, should become core concepts within the domain model.

Fig. 2 shows the legal domain model *dm:Legal* as a subclass of *dm:Requirement*. Requirements (*dm:Requirement*) are linked through conditions (*dm:Condition*) to resources (*dm:Resource*), applications (*dm:Application*) or application functionality (*dm:ApplicationFunctionality*) which may be used by users (*dm:User*). Requirements are external policies (e. g. law) and must be checked and taken into account during processing.

We model the structure of law texts in the domain model which is represented by the concept *dm:LawText* including its clauses (*dm:Clause*) and paragraphs (*dm:Paragraph*). A paragraph can either represent a claim (*dm:Claim*) or an explanation of the right (*dm:LegalExplanation*). Fig. 2 shows a taxonomy of German jurisdiction through the concept *dm:Jurisdiction*. For readability reasons Fig. 2 contains only an excerpt of the main areas of legal regulations. The specifics of the German law can be separated into two major areas: objective law (*dm:ObjectiveLaw*) and subjective right (*dm:SubjectiveRight*).

The first branch of objective law is the private law (*dm:PrivateLaw*), with the general private law (*dm:GeneralPrivateLaw*) and its subclass *dm:CivilLaw*. And the special private law (*dm:SpecialPrivateLaw*), with its subclasses *dm:CommercialLaw*, *dm:CopyrightLaw*, *dm:LaborLaw* and *dm:PrivateInternationalLaw*. The second branch is public law (*dm:PublicLaw*) with the subclasses *dm:CriminalLaw*, *dm:ProceduralLaw*, *dm:InternationalLaw*, *dm:SocialLaw*, *dm:ConstitutionalLaw* and *dm:AdministrativeLaw*.

The first branch of subjective right is the subjective private law (*dm:SubjectivePrivateLaw*), with the absolute right (*dm:AbsoluteRight*) and its subclasses *dm:PersonalRight* including their subclasses *dm:PrivacyLaw*, *dm:PropertyLaw* and *dm:IntellectualPropertyLaw* on the one hand and the property rights (*dm:PropertyRight*) with the subclasses *dm:TitleClaim* and *dm:RightToAlterALegalRelationship* on the other hand. The second branch is subjective public law (*dm:SubjectivePublicLaw*) with its subclasses *dm:StatusPositivus* (freedom through the state–protection), *dm:StatusNegativus* (freedom from the state–defense against state intervention) and *dm:StatusActivus* (freedom in and for the state–participation). Concepts of the class *dm:Jurisdiction* provides attributes to store the purpose and provision of the law and references to special law or retrains. This will be defined during the legal framework extension phase by a legal expert who is in charge.

```
#################################################################
#    Object Properties
#################################################################

### http://www.semanticweb.org/CONTact/domainmodel#hasSensitiveData
:hasSensitiveData rdf:type owl:ObjectProperty ;
                  rdfs:domain :Profil ,
                              :User ;
                  rdfs:range :SensitiveData .

#################################################################
#    Classes
#################################################################

### http://www.semanticweb.org/CONTact/domainmodel#User
:User rdf:type owl:Class ;
      :regulation :GDPR ;
      :sensitiveData :SensitiveData .

###  http://www.semanticweb.org/CONTact/domainmodel#Request
:Request rdf:type owl:Class ;
         rdfs:subClassOf :Action ;
         :approvalRequest "true"^^xsd:boolean ;
         :pictureUpload """For uploading the picture you...
            You can send an approval request to the affected...
            Until the requested approval... """^^xsd:string .

#################################################################
#    Annotation properties
#################################################################

### http://www.semanticweb.org/CONTact/domainmodel#email
:email rdf:type owl:AnnotationProperty ;
       rdfs:subPropertyOf :sensitiveData .

### http://www.semanticweb.org/CONTact/domainmodel#name
:name rdf:type owl:AnnotationProperty ;
      rdfs:subPropertyOf :sensitiveData .

###  http://www.semanticweb.org/CONTact/domainmodel#pictureUpload
:pictureUpload rdf:type owl:AnnotationProperty ;
               :description "This is regulated in the §19a ...
                of the copyright law. In case of ignoring ...
                compensation which are derived ... "^^xsd:string ;
               rdfs:subPropertyOf :regulation .
```

Figure 3: Excerpt of the configured legal domain model.

4.3 Representation of the Copyright Law in the Legal Context Model for a Collaboration Situation

Using our sample community application (cf. section 1) where Alice should be able to upload photos, we illustrate how a legal expert is involved into the development and extension of the context-based adaptive application (cf. Fig. 1). The process starts when the legal expert checks the domain model of the

application, e. g. using Protégé. First (cf. 1 in Fig. 1), the current version of the domain model is read in. The classes, properties and annotations are presented to the legal expert within the user interface. The context modelling with Protégé supports different ways to define attributes. We use annotations to define class attributes. Second (cf. 2 in Fig. 1), the legal expert extends the context model by adding new classes, relationships and annotations via the user interface. Additionally, he or she has to choose which information concern the data privacy law, which is a core concept that splits class annotations respectively attributes into sensitive (cf. Fig. 3, lines 5-9, 32-38) and non-sensitive data.

Figure 3 shows an excerpt of the legal domain model in Turtle format. It contains classes (cf. Fig. 3, lines 11-26), relationships (*subPropertyOf*, *subClassOf*) and annotations (cf. Fig. 3, lines 28-45) for aspects of the GDPR and the Copyright Law (cf. Fig. 3, lines 20-26, 40-45). After the legal expert finished the configuration, the changes will be written back to the context model (cf. 3 in Fig. 1). The extended version becomes the common ground for the related context-based applications (cf. 4 in Fig. 1). When a user interacts with the context-based systems he or she get information about the data usage and processing (cf. 5 in Fig. 1). The sensing and adaptation engine of the context-based system can analyze the current situation and react accordingly to the legal regulation as soon as the user does something what was regulated by the legal expert beforehand, e. g. a photo upload.

The process of extending the context model was implemented prototypically with Python and OWLReady2 using a structured interface (JSON format). The JSON structure uses the class names of the domain model, e. g. *dm:Application* or *dm:Action* (cf. Fig. 2), as a key and adds the information from the configuring persons (i. e. the legal expert in our scenario) as values, when necessary.

When writing back the changes, a Python module first creates new Python classes or extends existing ones which makes it possible to create and save new domain model classes with OWLReady2 to the domain model itself. The updated version of the domain model can be used to create instances during runtime. This currently works through a static configuration and a Python Command Line Interface (CLI). The program will be modified to a dynamic runtime in further development, when the sensing and adaptation engine are integrated. A user interface for the configuration of the context model has not yet been developed and is subject of future work.

Separate domain models are used to describe the specific domain and its relationships. In case of legal regulation, the domain model contains the structure and core classes of the specific jurisdiction, i. e. the German jurisdiction. But the core model does not contain all specifications and paragraphs of the law, because it would become too big and complex for processing. Instead the legal regulation processing.

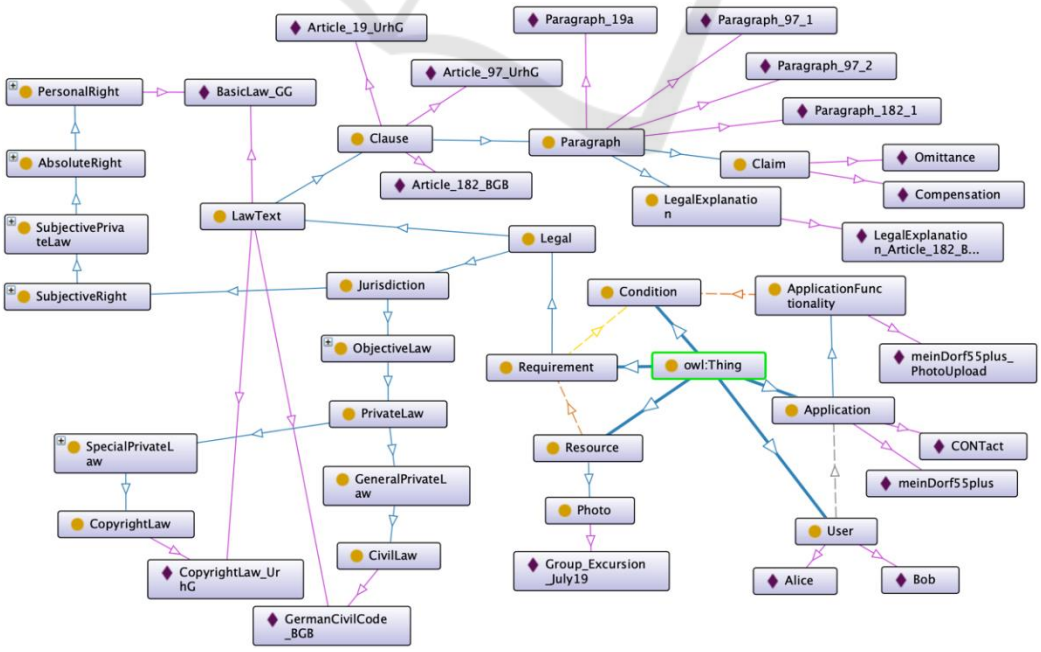

Figure 4: Contextualized state of Alice' picture upload scenario.

Instead the legal regulation which are relevant for a domain-specific application or application domain, e. g. a community system like meinDorf55+, are applied and instantiated for it and not in general. Therefore, the responsible legal expert of an application can decides what must be considered and how to deal with certain situations. The separation of applicable law from the core model makes it possible to change the legal context, when the application e. g. is used in another country with its own regulations and policies.

After this modelling step is finished, the application is available to the users using the context model shown in Fig. 4. For readability reasons we omitted concepts and instances in Fig. 4 to illustrate the scenario when Alice, a user of the application meinDorf55+, uploads a group photo which demands an approval of all pictured persons, e. g. the user Bob. According to the four-layer framework (Haake et al, 2010), Fig. 4 contains the state with concepts from Fig. 2 and instances which are needed to analyze and adapt according to the situation of Alice.

As Fig. 4 shows, Alice and Bob are instances of the concept *dm:User*. *CONTact* and *meinDorf55plus* are instances of the concept *dm:Application*. *meinDorf55plus_PhotoUpload* is an instance of the concept *dm:ApplicationFunctionality* and part of *meinDorf55plus*. Alice photo *Group_Excursion_ July19* is an instance of the concept *dm:Photo* which is a subclass of *dm:Resource*.

The usage of resources like the *Group_ Excursion_July19* demand the consideration of legal requirements (*dm:Legal*). The requirements lead to a set of rules with conditions (*dm:Condition*) which must be fulfilled before the upload process is finished. The photo upload affects the personal right (*dm:PersonalRight)*, that is regulated in the German basic law and represented by the instance *BasicLaw_GG*, the copyright with the instance *CopyrightLaw_UrhG* (of class *dm:CopyrightLaw*) and the German civil law with the instance *GermanCivilLaw_BGB* (of class *dm:CivilLaw*). The related clauses to the *CopyrightLaw_UrhG* are §19a UrhG (*Article_19_UrhG and Paragrapgh_19a*) which refers to §97 1,2 UrhG (instances *Article_ 97_UrhG*, *Paragraph_97_1* and *Paragraph_97_1*). The *dm:Claim* instances *Omittance* and *Compen-sation* are derived from §97 1, 2 UrhG. To fulfill the legal requirements of the upload in the scenario, Alice requires the consent of all person depicted. In case, the picture shows Bob, Alice become aware of this by an action (of the concept *dm:Action*), which creates a view or dialog from the instance *LegalExplanation_ Article_ 182_BGB* (class of *dm:LegalExplanation*). The explanation contains the descriptions of the instances *Article_182_BGB* (§182 1 BGB the approval) and the related paragraph *Parapgraph_182_1*. Claims and legal explanations serve to explain the need for an action and the consequences when Alice disregards the legal rule. An approval request to Bob will be created when Alice continues the upload process. This time the action (*dm:Action*) creates an *ApprovalRequest* which can be accepted or declined by Bob (cf. Listing 1). If he accepts, the context (acceptance, related photo, Alice request, etc.) will be stored in an approval object for transparency and verifiability, else the process will be aborted.

```
rule "Approval Request"
 when
  user: getUserInContext("dm:User")
  app: getAppInContext(user,"dm:Application")
  req: getReqInContext(app,"dm:Requirement")
  appr: requestApproval(user, app, req)
 then
  createOrUpdateAcceptedApproval(appr)
  notify(user, appr)
 end
end
```

Listing 1: Rule “Approval Request”.

Listing 1 uses pseudocode to illustrate our approach to implement “Approval Request”. The rule consists of a condition part (when to then) and an action block (then to end). *getUserInContext* retrieves the user interacting with the application (in our scenario Alice and Bob). The function *getAppInContext* determines the application used by the user which is of type *dm:Application*. The function *getReqInContext* 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 photo upload. When the user approved the action beforehand, the return value of the function is empty. When there is no or an inapplicable approval instance present in the current context, the approval is requested from the user.

5 GENERATE PERSONALIZED EXPLANATIONS

To generate personalized explanations through a domain-specific application, we use the templates and explanatory texts of the experts. For that, the core

concepts *dm:Requirement*, *dm:Condition* and *dm:Declaration* and their dependencies from eCBASE will be used in the current situation of the domain-specific application, i. e. meinDorf55+. Through them, we support personalized adaptations, user control and explanations for the users.
The purpose of these concepts is to explain 'What happened?' (*dm:Requirement*), 'Why does it happened?' (*dm:Condition*) and 'What kind of explanation should be provided?' (*dm:Declaration*).

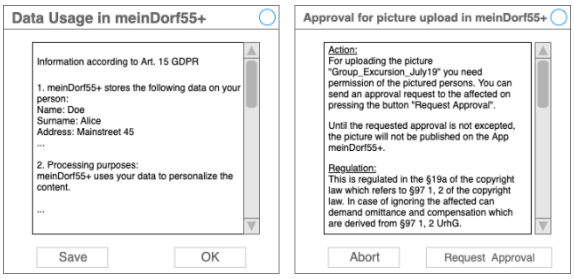

Figure 5: Sample dialogs for personalized explanations.

Figure 5 shows two examples of user interface dialogs. They will be generated using the requirements and explanations which are defined and configured by the legal expert. The left dialog in Fig. 5 shows an explanation of the data usage of Alice concerning the usage of meinDorf55+. The dialog on the right gets displayed when a user of meinDorf55+ tries to upload a photo. The dialog contains an explanation what has happened and why the interaction is needed. Additionally, it shows the required action and the conditions (defined by the legal expert) which must be fulfilled before the photo becomes accessible to others. Fig. 5 illustrates how the two explanations could be built.

6 DISCUSSION

We integrate and support legal experts in the development of our eCBASE. For that, we presented a process of our on-going work for integrating legal regulations to support developers, legal experts and providers to design and develop compliant context-based applications. Our approach enables legal experts to participate in the software development process. They shall define the legal requirements, the explanations of the law and the related consequences which are transformed to adaptation rules. From this eCBASE generates personalized explanations for the users. We presented our legal domain model with its basic concepts which is used to represent the definitions. According to (Haake et al, 2010) we described a contextualized perspective with a situation and the rule "Approval Request" when the Copyright Law gets relevant.

The legal experts can have different expertises, depending on the software application and domain. In addition, not all legal requirements need to be defined in eCBASE and the initial domain-specific applications in advance. This enables us to achieve more flexibility in the application of our context-based adaptive system environment. We also consider that users should make decisions about the data processing (accept or decline) by themselves. For that, we integrate the defined legal policies, which are important for the situation (the context) and analyze which of the policies must be applied in the specific situation. Through our context-based adaptive approach it is possible to observe activities in our system which affects legal regulations and to explain the required actions and consequences to 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 legal requirements (defined by legal experts) with the domain-specific application by the concept *dm:Condition*. (II.) Our prototype currently has a static setup for the situation of data usage explanation and the picture upload. (III.) The explanation building process that realizes the dialogs is not yet implemented. The presented dialogs are still concepts which must be realized in the next step. (IV.) For readability or space reasons we presented an excerpt of our developed domain model, i. e. we have omitted other concepts and relationships for legal regulations.

7 CONCLUSION AND FUTURE WORK

In this paper we presented an approach to consider legal policies and personalized explanations in our eCBASE. For that, we use the knowledge of a legal expert who defines the relevant policies and explanations using our presented underlying legal domain model. The expert's definitions will be stored in and applied through our legal context model in eCBASE (answer to Q1). Therefore, our approach can support users to comply with the law when using the application by generating personalized explanations (answer to Q2). For that, we use the specific context to explain the user the current situation and the related consequences to the usage of the system. We illustrated that in our sample scenario when Alice uploads a group photo with Bob, what

concerns the Copyright Law and the GDPR. Alice' action affects the defined policies for the system. That triggers the generating of an explanation of further steps to Alice as well as a request approval to Bob.

The next steps include the investigating of what are intelligible explanations for users which are related to the relevant legal regulation. In addition, a user interface for depositing legal policies is in its design phase, which the legal expert can use for configuration. We will extend our legal domain model with relevant regulations and try to integrate the more specific models of (Bartolini et al, 2015) and (Delgado et al., 2003) into our domain model to apply it in our adaptation process. Additionally, we are looking for further legal regulations which are relevant in other application domains to investigate and integrate them into eCBASE.

ACKNOWLEDGEMENTS

This work was funded by the Research Cluster "Digitalization, Diversity and Lifelong Learning – Consequences for Higher Education" (D^2L^2) of the FernUniversität in Hagen, Germany.

REFERENCES

Bartolini, C., Muthuri, R., Santos, C., 2015. Using ontologies to model data protection requirements in workflows. In JSAI International Symposium on Artificial Intelligence (pp. 233-248). Springer.

Baumann, C., Peitz, P., Raabe, O., & Wacker, R., 2010. Compliance for Service based Systems through Formalization of Law. In WEBIST (2) (pp. 367-371).

Bellotti, V., Sellen, A., 1993. Design for privacy in ubiquitous computing environments. In: Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13–17 September 1993, Milan, Italy ECSCW'93, pp. 77-92. Springer.

Brezillon, P. J., 1994. Contextualized explanations. In: Proceedings of International Conference on Expert Systems for Development, pp. 119-124. IEEE.

Casanovas, P., Palmirani, M., Peroni, S., van Engers, T., Vitali, F., 2016. Semantic web for the legal domain: the next step. Semantic web, 7(3), 213-227.

Casellas, N., Nieto, J. E., Meroño, A., Roig, A., Torralba, S., Reyes, M., Casanovas, P., 2010. Ontological semantics for data privacy compliance: The Neurona project. In 2010 AAAI Spring Symposium Series.

Delgado, J., Gallego, I., Llorente, S., & García, R., 2003. Regulatory Ontologies: An Intellectual Property Rights approach. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 621-634). Springer.

Dey, A. K., 2001. Understanding and using context. In: Personal and ubiquitous computing, vol. 5, no. 1, pp. 4-7.

Dey, A. K., Newberger, A., 2009. Support for context-aware intelligibility and control. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 859-868. ACM.

Gangemi, A., 2007. Design Patterns for Legal Ontology Constructions. LOAIT, 2007, 65-85.

Gangemi, A., Prisco, A., Sagri, M. T., Steve, G., Tiscornia, D., 2003. Some ontological tools to support legal regulatory compliance, with a case study. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pp. 607-620. Springer.

Goram, M., Veiel, D., 2019. Supporting Privacy Control and Personalized Data Usage Explanations in a Context-Based Adaptive Collaboration Environment. In G. Bella & P. Bouquet (Eds.), Modeling and Using Context. Vol. 11939, pp. 84–97. Springer.

Gregor, S., Benbasat, I., 1999. Explanations from intelligent systems: Theoretical foundations and implications for practice. In: MIS quarterly, vol. 23, no. 4, pp. 497-530.

Guarino, N., 1998. Formal ontology in information systems. In: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy, vol. 46. IOS press.

Haake, J. M., Hussein, T., Joop, B., Lukosch, S., Veiel, D., Ziegler, J., 2010. Modeling and exploiting context for adaptive collaboration. International Journal of Cooperative Information Systems, 19(01n02), pp. 71-120.

Jiang, X., Landay, J. A., 2002. Modeling privacy control in context-aware systems. In: IEEE Pervasive computing, vol. 1, no 3, pp. 59-63.

Kapitsaki, G. M., 2013. Reflecting user privacy preferences in context-aware web services. In: 2013 IEEE 20th International Conference on Web Services, pp. 123-130. IEEE.

Kapitsaki, G., Ioannou, J., Cardoso, J., Pedrinaci, C., 2018. Linked USDL Privacy: Describing Privacy Policies for Services. In: 2018 IEEE International Conference on Web Services (ICWS), pp. 50-57. IEEE.

Lim, B. Y., Dey, A. K., 2010. Toolkit to support intelligibility in context-aware applications. In: Proceedings of the 12th ACM international conference on Ubiquitous computing, pp. 13-22. ACM.

Lim, B. Y., Dey, A. K., 2011. 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.

Veiel, D., Haake, J. M., Lukosch, S., Kolfschoten, G., 2013. 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.