Using Models to Enable Compliance Checking against the GDPR: An Experience Report

Damiano Torre*, Ghanem Soltana*, Mehrdad Sabetzadeh*, Lionel C. Briand*†, Yuri Auffinger§, Peter Goes§
*SnT Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg
†School of Electrical Engineering and Computer Science, University of Ottawa, Canada
§Linklaters LLP, Luxembourg
{torre, soltana, sabetzadeh, briand}@svv.lu, {yuri.auffinger, peter.goes}@linklaters.com

***Abstract*—The General Data Protection Regulation (GDPR) harmonizes data privacy laws and regulations across Europe. Through the GDPR, individuals are able to better control their personal data in the face of new technological developments. While the GDPR is highly advantageous to individuals, complying with it poses major challenges for organizations that control or process personal data. Since no automated solution with broad industrial applicability currently exists for GDPR compliance checking, organizations have no choice but to perform costly manual audits to ensure compliance. In this paper, we share our experience building a UML representation of the GDPR as a first step towards the development of future automated methods for assessing compliance with the GDPR. Given that a concrete implementation of the GDPR is affected by the national laws of the EU member states, GDPR's expanding body of case law and other contextual information, we propose a two-tiered representation of the GDPR: a generic tier and a specialized tier. The generic tier captures the concepts and principles of the GDPR that apply to all contexts, whereas the specialized tier describes a specific tailoring of the generic tier to a given context, including the contextual variations that may impact the interpretation and application of the GDPR. We further present the challenges we faced in our modeling endeavor, the lessons we learned from it, and future directions for research.**

***Index Terms*—General Data Protection Regulation, Regulatory Compliance, UML, OCL.**

I. Introduction

With the growing concerns about data protection and privacy, it is becoming increasingly important to assess compliance with the relevant regulations. In Europe and indeed worldwide, the General Data Protection Regulation (GDPR) [1] is now widely viewed as a benchmark for data protection and privacy regulations. The GDPR came into effect in May 2018, replacing the previous Data Protection Directive, 95/46/EC. The GDPR has been designed to harmonize data privacy laws across Europe in order to provide further protection and capabilities to individuals for controlling their personal data in the face of new technological developments [2]. While undoubtedly beneficial to individuals in many ways, the reality with the GDPR is that organizations are having severe difficulties in understanding what compliance means in this new environment and how to implement the GDPR [3].

In order to comply with the requirements of the GDPR, organizations need to consider the principles of personal data processing as set out in the GDPR and to make regular reviews of their measures, practices and processes regarding the collection, use and protection of personal data. Failure to comply with the GDPR may result in fines of up to €20m or 4% of an organization's global turnover for specific breaches [4]. In addition, organizations are liable for damages and other remedies towards individuals in case of data breaches [1]. For this reason, there is now a fast-growing need for cost-effective methods that will help different business sectors achieve, demonstrate and maintain compliance with the GDPR. Given the sheer complexity of the systems and services that are subject to the GDPR, e.g., e-Government applications and cloud-based services, automated support for GDPR analysis is critically important. At the moment, there is a lack of such support on the market. This gap will become even more evident once individuals start to exercise their rights under the GDPR, likely resulting in an onslaught of new legal challenges for companies. Due to the absence of automated solutions, we have started a long-term investigation, involving both IT researchers and legal experts, into GDPR compliance automation. Our ultimate goal is to bring scalability to GDPR compliance assessment and create opportunities for developing innovative GDPR-related services.

The GDPR is considered the most far-reaching and technically demanding personal data privacy regulation ever established. The high level of rigor that ensuring GDPR compliance entails is increasingly comparable to what is required for demonstrating compliance to safety standards and regulations. GDPR compliance analysis can thus benefit from existing work where models have been employed for systematic compliance analysis in the context of safety certification, e.g., [5]. While highly advantageous, encoding the GDPR and its compliance mechanisms into a model-based representation is a complicated task. In particular, the level of abstraction of such a representation has to be suitable for ensuring a consistent implementation and interpretation of the regulation, national laws and case law.

In this paper, we draw on Model-Driven Engineering (MDE) [6] for building a machine-analyzable representation of the GDPR as a first step towards the development of future automated methods for assessing GDPR compliance. Although MDE is primarily a paradigm for reducing the complexity of systems development [7], over the years, MDE has outgrown its traditional use and is now increasingly applied as a general mechanism for structuring domain knowledge. When employed in this broader sense, as we do in our work, MDE provides an effective communication bridge between IT

experts and domain experts, such as legal experts, who may have little software development expertise.

What we pursue in this paper through the application of MDE is a visual and yet precise representation of the textual content of the GDPR. Since a concrete implementation of the GDPR is affected by the national laws of the EU member states, the GDPR's expanding body of case law and other contextual factors, we propose a two-tiered representation of the GDPR: a generic tier and a specialized tier. The generic tier captures the concepts and principles of the GDPR that apply to all contexts, whereas the specialized tier describes a specific tailoring of the generic tier to a given context, including the contextual variations that may impact the interpretation and application of the GDPR. We represent both the generic and specialized tiers using UML class diagrams [8] and a set of invariants expressed in the Object Constraint Language (OCL) [9]. In particular, as we explain in detail in Section III, we provide an overview of our long-term research project involving four steps: (1) building a generic tier of the GDPR, (2) tailoring the generic tier into a specialized one, (3) developing tool support for representing technical and legal documents in a structured form, and (4) enabling checking GDPR compliance. In this paper, we focus exclusively on our experience conducting steps 1 and 2; steps 3 and 4 are work in progress and left to future work.

Several strands of work employ models for expressing legal requirements and assessing whether and to what extent these requirements are met by a given system. These strands include the large body of research concerned with the application of goal models to laws and regulations, e.g., [10], [11], as well as a number of conceptual modeling techniques aimed at representing the semantics of legal texts, such as key legal abstractions and modalities, e.g., [12]–[14], and the structural representation of legal texts, e.g., [15]–[17].

As we discuss in more detail in Section VII, existing model-based approaches for compliance verification have one of the following limitations as far as the GDPR is concerned: they (1) have a different focus than the GDPR, e.g., [5], (2) present guidelines only for the manual application of the GDPR, e.g., [18], or (3) focus exclusively on specific GDPR use cases, e.g., [19], [20]. To the best of our knowledge, there are no proposals in the literature aimed at providing a holistic model-based representation of the GDPR. We attempt to address this gap in this paper. Specifically, we tackle the following three research questions (RQ):

- *RQ1: How can we develop a generic and adaptable model-based representation of the GDPR to support automated compliance checking?*
- *RQ2: How can we tailor the generic GDPR model according to the specific needs of a given context?*
- *RQ3: What are the challenges in modeling the GDPR?*

Contributions. Our contributions are as follows:

(1) We build a generic model of the GDPR using UML class diagrams and OCL constraints. We use the term "generic" to signify the fact that the model is based only on the content of the GDPR and not encompassing any complementary information that may be necessary to contextualize the GDPR for use in a particular situation.

(2) The exact realization of the GDPR is subject to some variability depending on context. We present guidelines for tailoring the generic GDPR model (first contribution) into a specialized model that is suitable for application in a specific context. To this end, we describe what variations are admitted by the GDPR and our strategy for handling these variations.

(3) We reflect on the lessons learned from encoding the GDPR into a model-based representation. Our lessons, which cover model validation, traceability and contextualization, provide a useful stepping stone for UML-based specification of other complex laws and regulations.

(4) We present the challenges we identified during our modeling endeavor alongside a number of future directions aimed at addressing these challenges.

Structure. Section II introduces basic concepts related to the GDPR. Section III provides an overview of our approach. Section IV addresses our research questions. Section V and VI present lessons learned and future directions, respectively. Section VII compares with related work. Section VIII concludes the paper.

II. GDPR OVERVIEW

The GDPR [1] is a complex piece of legislation comprised of 173 recitals, and 99 articles divided into 11 chapters. The GDPR applies primarily to businesses established in the EU. However, the regulation may also apply to businesses outside the EU, e.g., when these businesses offer goods or services to, or monitor individuals in the EU. If a business is subject to the GDPR, it has to identify itself as either a data controller or data processor. A controller determines the purpose and means of the processing, whereas a processor acts on the instructions of the controller. The responsibilities of a given business under the GDPR vary depending on whether it is a processor or a controller and depending on the kind of data processed.

Processors notably have to: (i) implement adequate technical and organizational measures to keep personal data safe and secure, and, in cases of data breaches, notify the controllers; (ii) appoint a statutory data protection officer and conduct a formal impact assessment for certain types of high-risk processing; (iii) keep records about their data processing; and (iv) comply to the GDPR restrictions when transferring personal data outside the EU.

In comparison to processors, controllers are subject to more GDPR obligations. In particular, in addition to having to meet the obligations mentioned above, controllers have to: (i) adhere to six core personal data processing principles, namely, fair and lawful processing, purpose limitation, data minimization, data accuracy, storage limitation, and data security; (ii) keep identifiable individuals informed about how their personal data will be used; and (iii) preserve the individual rights envisaged by the GDPR, e.g., the right to be forgotten and the right to lodge a complaint.

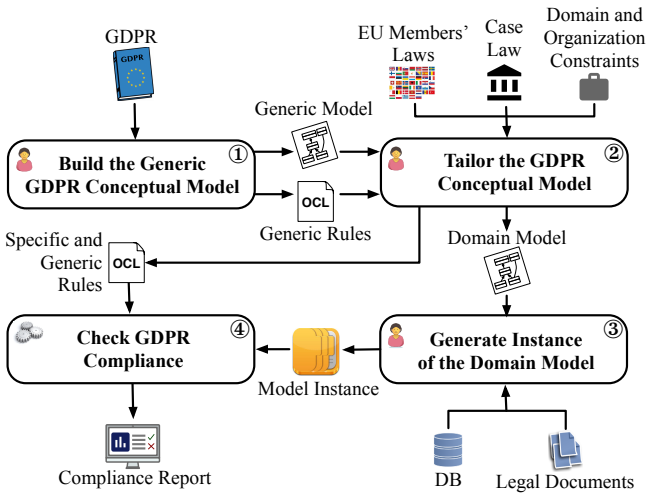

Fig. 1. Approach for Automated GDPR Compliance Checking

III. Towards a Model-based Approach for Automated GDPR Compliance Checking

Our approach for enabling automated GDPR compliance checking has four steps, as depicted in Fig. 1.

Step 1 is a manual, one-off task aimed at building a generic model of the GDPR with the help of legal experts. More specifically, the goal of this step is to build, using UML class diagrams and OCL, a context-independent representation of the GDPR that does not take into account specific situations where EU member states' national laws, case law, or domain/organization decisions may affect the operationalization of the regulation. In this step, we develop, through a qualitative study, the following: (i) a generic model of the GDPR's main concepts and relationships, (ii) generic OCL constraints and obligations that verify GDPR compliance, (iii) a glossary to facilitate the understanding of the GDPR, and (iv) the variation points describing specific situations where the generic representation needs to be adapted to a given domain or organizational context.

In step 2, we process the generic model and OCL constraints of step 1 in order to tailor them into a specialized model and a (specialized) set of OCL constraints. The goal of step 2 is to build an actionable basis for implementing the GDPR according to (i) the national laws of EU member states, (ii) GDPR case law, and (iii) other contextual information that may complement the GDPR. Among other things, step 2 yields two outputs that will later enable automated compliance checking in step 3. These outputs are: (i) a specialized model that represents the model tailored according to the application context, and (ii) a set of specialized OCL constraints which contain revised versions of the generic constraints developed in step 1 and potentially new constraints.

Step 3 concerns the development of a model-instance generation tool in order to create instances of the specialized model obtained from step 2. This is done via a model-editing tool that allows legal experts to create representations of legal and technical documents in the form of an instance of the specialized model. An example of legal documents would be privacy policy statements, and an example of technical documents would be system requirements specifications. Stated otherwise, step 3 generates a model instance providing a structured representation of the legal and technical documents that have a bearing on GDPR compliance.

Finally, in step 4, the model instance generated from step 3 is checked against the specialized OCL constraints obtained from step 2. The compliance diagnostics resulting from the constraint checking process are then delivered to end-users, typically legal experts, in a user-friendly manner.

In this paper, we describe our experience conducting steps 1 and 2. Step 3 and 4 are left to future work. Steps 1 and 2 along with their inputs and outputs are discussed in detail in Sections IV-A and IV-B.

IV. Modeling GDPR

A. Building a Generic Model for the GDPR (RQ1)

In the first step of our approach (Fig. 1), we build a generic model representing the GDPR without accounting for the specificities of the application domain. This modeling activity addresses RQ1 and yields: (1) a UML Class Model (CM) that captures the GDPR's key concepts and their relationships, and the variability in the CM; (2) a set of OCL constraints over the CM reflecting the GDPR's obligations and rules. Most of these constraints are only partly specified at this stage. Given a specific context, the applicable constraints need to be completed so that one can evaluate them in an automated and precise manner; (3) a glossary of terms including an intuitive textual description of each OCL constraint; (4) a table that maps the obligations and rules to their corresponding constraints; and (5) a table that summarizes all variation points extracted from the GDPR. The two output tables mentioned above aim to facilitate the work of analysts in the subsequent tailoring step (Section IV-B). Below, we explain the methodology we employed to create these outputs. We then illustrate the outputs using concrete examples.

Modeling methodology. This modeling activity was performed in an iterative and incremental manner. Each iteration was interleaved with a thorough validation session with legal experts, noting that legal experts were already trained to understand the CM notation. The second author of this work, who has 6 years of formal training in computer science and 5 years of experience in MDE, did most of the modeling and constraints writing. Building the generic model for the GDPR took four iterations with each iteration requiring on average two weeks. In addition to off-line validation, we had several face-to-face validation sessions with legal experts, with each of these sessions lasting between 2 to 3 hours.

During the first iteration, we read the GDPR in its entirety and tried to extract important definitions, concepts, rules, obligations and possible variations from it. Fig. 2 illustrates the information extracted from Art. 8 – the article regulating how a child data subject can provide consent for processing her personal data in the context of information society services. In particular, eleven concepts (shaded gray), one

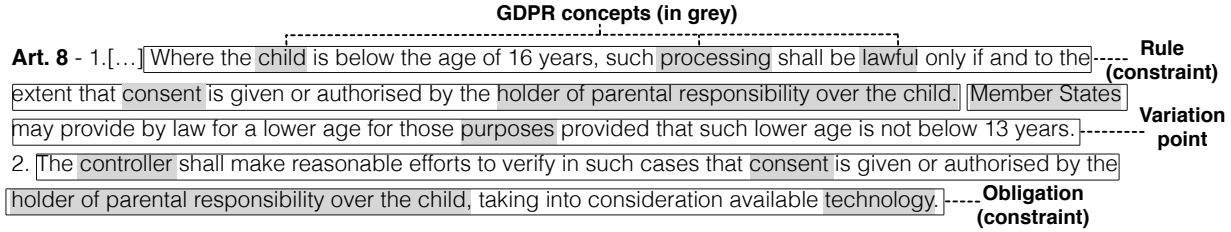

Fig. 2. Example of Information Extracted from (Excerpt of) Article 8 of the GDPR

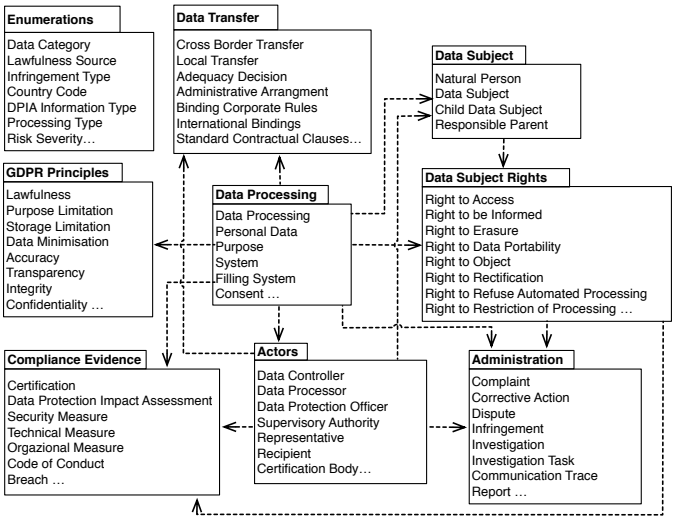

Fig. 3. (Simplified) Package Representation of the CM

rule, one variation point, and one obligation were extracted from Art. 8. Recent work uses natural language processing techniques to extract such legal information in an automated manner [21]. Nevertheless, we opted for a manual strategy to avoid overlooking any important information while deepening our understanding of the GDPR. Among other reasons, a manual strategy was essential for enabling the identification of GDPR rules and obligations in a fully precise manner. For example, we have mapped the rule and obligation in Art. 8 to their corresponding OCL constraints as we illustrate later.

Based on the extracted information, and using our understanding and interpretation, we created the modeling artifacts listed earlier. Next, these artifacts were presented to legal experts for feedback. In addition to pointing out issues and omissions, our collaborating legal experts were encouraged to bring to our attention any GDPR article that they suspected might have been misinterpreted, i.e., incorrectly modeled. By doing so, we boosted subsequent iterations since we no longer needed to analyze the entire GDPR again.

In practice, we observed that the corrections suggested by the legal experts were, by and large, based on conventions or articles that were not part of the GDPR itself, e.g., articles from the Article 29 Working Party (WP)[1]. For example, a data controller might need to simultaneously communicate with many supervisory authorities; such authorities are established by individual European member states to supervise compliance with the GDPR. In such a case, the controller has to designate a unique *lead* supervisory authority (Art. 56). Subsequently, the controller should only communicate with the lead supervisory authority, which in return, will coordinate any investigation or administrative task with the other concerned authorities. Although not explicitly stated in the GDPR, the choice of the lead supervisory authority is not arbitrary. The lead supervisory authority should be selected based on predefined rules that account, among other things, for the location of the main establishment of the controller and where the actual data processing is taking place (Working package 244 of the WP).

In the next modeling iteration, we re-read the GDPR parts and other annex documents that were noted by the legal experts in the previous iteration. Then, we refined the outputs according to expert feedback, and so on. Once the specialized model started to stabilize, we put together a general report including all the resulting outputs for off-line validation. The modeling step terminated when the general report was approved by the legal experts.

Illustration of the modeling artifacts. Fig. 3 depicts a simplified view of the CM's packages. To keep the CM manageable and easy to grasp as it grows in size, we spread the CM classes over nine packages as follows, noting the package names are self-explanatory. Packages *GDPR Principles*, *Data Subject Rights*, and *Data Transfer* respectively cover chapters 2, 3, and 5 of the GDPR. Concepts from chapters 1, 4, 8 and 9 were spread over the remaining packages based on their meanings and roles. For example, concepts from chapter 4, which is the longest chapter and where most GDPR compliance requirements are defined, are grouped in packages *Data Processing*, *Compliance Evidence*, and *Actors*. Chapters 6, 7, 10, and 11 have little to no impact on compliance checking, and subsequently were excluded after the first modeling iteration. For example, chapter 6 regulates the internal functioning and composition of the public data supervisory authorities. We then show in Fig. 4 an excerpt of the *Data Processing* package that covers most concepts extracted from Art. 8 in Fig. 2.

Intuitively, the CM in Fig. 4 presents the information that has to be collected when the lawfulness of data processing is based on consent. In the CM, only data processing manipulating some personal data should be considered (see *manipulates* association between *Data Processing* and *Personal Data*). Other kinds of processing are out of scope. The purposes for

[1]Art. 29 WP is the independent European working party that dealt with issues relating to the protection of privacy and personal data until 25 May 2018 (date at which the GDPR took effect). All archives from Art. 29 WP are available at: https://ec.europa.eu/newsroom/article29/news-overview.cfm. Art. WP 29 has been replaced by the European Data Protection Board; see https://edpb.europa.eu

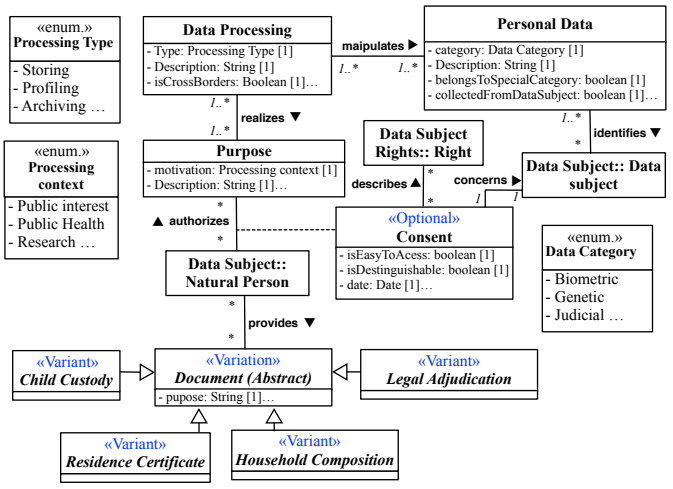

Fig. 4. Excerpt of the *Data Processing* Package

each processing have to be explicitly defined (see *realizes* association between *Data Processing* and *Purpose*), noting that several instances of processing can share a unique purpose. A well-designed consent form should, among other things, remind data subjects of all their applicable GDPR rights, e.g., right to lodge a complaint (see *describes* association between *Consent* and *Right*). Consent is given by data subjects, or their responsible parent in case of a child data subject, for one or more predefined processing purposes (see *authorizes* association between *Natural Person* and *Purpose* and *concerns* association between *Data Subject* and *Consent*). This is only possible when the treated personal data is sufficient for the precise identification of data subjects (see *identifies* association between *Personal Data* and *Data Subject*). Responsible parents might provide several kinds of documents to prove their eligibility to act on behalf of their children (see *provides* association between *Natural Person* and *Document*).

The stereotypes highlighted in blue in Fig. 4, i.e., «Optional», «Variation», and «Variant», capture the variability in the CM. These stereotypes come from work by Ziadi and Jezequel [22] which aims to model variability in domain models in the context of product lines. The stereotypes «Variation» and «Variant» explicitly specify variability associated with inheritance. In particular, the variation point is denoted by an abstract class and its variants are defined as its concrete subclasses [22]. For example, the type of accepted documents needed to prove that a person is indeed the responsible parent of a given data subject varies from one country to another. Subsequently, the *Document* class is stereotyped as «Variation» and all its subclasses are stereotyped as «Variant». This means that the number of subclasses of the *Document* class and how each subclass is defined, e.g., number and name of attributes, change from one context to another. The «Optional» stereotype is also used for non-mandatory entities which are not part of any inheritance. For example, the *Consent* class is only relevant for systems that claim lawfulness based on consent. In addition to capturing variability in an analyzable form, the aforementioned stereotypes give visual cues on where changes

```
1  context Data_Processing inv consentProvider:
2  self.isLawfulnessOnlyByConsent() implies
3  let identifiableSubjects: Set(Data_Subject) = self.personalData.
4  identifiableSubjects->flatten()->asSet() in
5  self.purposes->forAll(p: Purpose|
6      identifiableSubjects->forAll(ds: Data_Subject|
7          let eligibleToGiveConsent: Natural_Person =
8          if(ds.oclIsTypeOf(Data_Subject)) then ds
9          else ds.getResponsibleParent() endif in
10         p.getConsents()->exists(c: Consent|
11             c.provider = eligibleToGiveConsent
12             and c.target = ds)))
13 context Data_Subject inv VAR_DSAge:
14 let minDSAge: Integer = Variability.V_getMinimumAgeForDS(self) in
15 if(self.oclIsTypeOf(Child_Data_Subject)) then
16  self.getAge() < minDSAge else self.getAge() >= minDSAge endif
17 context Natural_Person inv VAR_isLegalParent:
18 self.children->forAll(c: Child_Data_Subject| self.
     V_checkParentDocuments(c))
```

Fig. 5. Examples of OCL Constraints

in the CM are more likely to occur during the specialization step (in Section IV-B). For example, all classes with variability stereotypes in the CM might be removed from the specialized CM based on the contextual information at hand.

The CM comes with 55 OCL constraints. OCL constraints are expressed as invariants denoting logical conditions that must always hold over all instances of a given class. As mentioned earlier, Fig. 5 presents three OCL constraints related to the CM fragment in Fig. 4. For example, the invariant named consentProvider checks that any instance from the class *Data Processing* satisfies some conditions as stipulated by the GPDR. Specifically, when the lawfulness of a given data processing is based only on consent (L. 2), all the concerned data subjects must provide material agreements for the underlying processing purposes (L. 3-6, L. 10 and L. 12). Additionally, the invariant checks that, as stated in Art. 8 of Fig. 2, consent for child data subjects is provided by their legal parent (L. 7-11). This constraint involves no variability and does not require any additional tailoring in the subsequent step.

Constraints involving variability are distinguishable by their name, which includes the "VAR_" prefix, e.g., VAR_DSAge. We handle variability in OCL constraints using partially specified operations that need to be later updated or redefined based on the context at hand. As a convention, we start the name of these special operations by the "V_" prefix, e.g., V_getMinimumAgeForDS. For example, the second constraint (L. 13-16) states that the age of data subjects should be greater than a certain dynamic threshold; the generic operation returns 16. However, when the context is known, the operation V_getMinimumAgeForDS should dynamically identify the value of the threshold based on the country of residence of the data subject and the locations of the involved data processing, controllers, and processors. We further discuss variability in Section IV-B.

To ease the understanding of the modeling artifacts for legal experts, we rely on a glossary of important terms. The glossary has 54 entries for the CM. This glossary further includes intuitive descriptions for all the OCL constraints. Table I presents an excerpt of the glossary that supports the CM fragment in Fig. 4 and the OCL constraints in Fig. 5. The first column points to the modeled concept, e.g., classes

TABLE I
GLOSSARY EXCERPT

Concept	Traceability	Intuitive Description
Personal Data	Arts. 4, 9, and 10	Means any information relating to an identified or identifiable natural person.
Data Processing	Art. 4	Is any operation performed on personal data, whether or not by automated means, including collection, recording, organization, structuring, storage, etc.
Data Controller	Arts. 4 and 24	A natural or legal person, public authority, agency or any other body which, alone or jointly with others, determines the purposes and means of the processing of personal data.
Data subject	Art. 4	A natural person whose personal data is processed.
Consent	Arts. 4, 7, and 8	Means any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she signifies agreement to the processing of him or her personal data.
Lawfulness	Arts. 6 and 9	A data processing is said to be lawful if its legal basis matches one of the possible circumstances under which GDPR permits the processing of personal data. Example of valid legal basis for data processing are consent and when processing is necessary to perform or prepare for a contract with the data subject.
ConsentProvider (OCL constraint)	Art. 8	In case that data processing is only based on consent, this constraint checks that all identifiable and concerned data subjects have agreed on the purposes of the processing. Consent must be provided by data subject themselves, unless for child data subjects, for whom consent should be provided by their legal parent.
VAR_DSAge (OCL constraint)	Art. 8	Ensures that data subjects under a certain age (default = 16) are classified as child data subjects.
VAR_isLegalParent (OCL constraint)	Art. 8	Checks that a given person is indeed the holder of parental responsibility over a given data subject.

and constraints. The second column lists the GDPR source articles of the elements in the first column. Here, traceability is meant to help legal experts during the validation sessions. In particular, it makes it easier to spot whether we have missed some important articles that might further consolidate the definition of a given concept. The last column presents an intuitive natural-language description of the element in the first column. For example, the constraints in Fig. 5 are described in the last three rows of Table I.

As illustrated by Table II, the next modeling artifact is a table that maps obligations and rules to their corresponding OCL constraints and variability elements. This table documents the major modeling decisions made and will be used during the subsequent tailoring activities. The first column provides a unique identifier for the obligations while the second column indicates the source articles from where the obligation comes. The third column provides a textual description of the obligation itself. For example, Table II contains the rule (O1) and obligation (O2) highlighted in Fig. 2. The fourth column lists the OCL constraints that are used to encode the obligation.

TABLE II
(EXAMPLE) OBLIGATION AND RULE ENCODINGS

ID	Art.	Obligation & rules	Constraints & Variability
O1	8	When the data subject age is below a certain threshold (by default 16), the controller must ensure that consent is given or authorized by the holder of parental responsibility over the child.	**Constraints:** consentProvider VAR_DSAge **Variability in the CM:** *Consent* class
O2	8	The controller shall make reasonable efforts to verify that consent is given by the holder of parental responsibility over the child.	**Constraints:** VAR_isLegalParent **Variability in the CM:** *Consent* class and *Document* class and its subclasses
...	...	...	...

TABLE III
EXCERPT OF VARIABILITY TABLE

ID	Trac.	Actor	Description	When & how to resolve?
V1	Art. 8 (1), O1	EMS	The EMS law may provide for a lower age from which parental consent is no longer required, provided that such lower age is not below 13 years.	[When?] If there is at least one processing involving child data subjects. [How?] Override V_getMinimumAgeForDS based on the EMS laws.
...	...	...	...	
V16	Art. 45(3), O41	EC	The EC may decide that a third country, a territory or a specific sector within a third country ensures an adequate level of protection for data transfer.	[When?] If there is at least one cross-border data transfer. [How?] Update the CM and VAR_checkLeagalTransfer constraint based on the EC's adequacy decisions.
...	...	...	...	

For example, for O1 to be fulfilled, both constraints consentProvider and VAR_DSAge in Fig. 5 have to hold. In addition to the constraints, the variability elements in the CM that are related to the obligation row are also listed (fourth column). Constraints with variability are easy to identify thanks to the "VAR_", thus not requiring to be re-listed again.

The last modeling artifact is a table including all possible variation points extracted from the GDPR. We defer the discussion and presentation of this table to Section IV-B.

B. Specializing the Generic Model (RQ2)

In the second step of our approach (Fig. 1), analysts tailor the generic modeling artifacts to account for the specific context and activities of the organizations seeking compliance. This step addresses RQ2. Generally speaking, analysts have to resolve all the variations that are relevant to the context at hand. This might also introduce new obligations coming from other European and International laws. The output of this step is a specialized and augmented version of the modeling artifacts created in the first step of our approach (Section IV-A).

As mentioned in Section III, the variability in the GDPR comes from the fact that the interpretation or the enforcement of some provisions may be affected by additional acts and laws from the European Commission (EC) and the European Member States (EMS), relevant privacy authorities and courts.

In total, 12 alineas (paragraphs) belonging to 8 articles delegate some legislative power to the EC; whereas 24 alineas spread over 20 articles do the same for the EMS. Table III presents an example of the variability created during the first step of our approach. This table will guide analysts as to how and when they should resolve a given variability.

The first column of the table represents the identifier of the variation point. This identifier will be later used to record how a given variation was resolved. The second column traces the variability to (1) the legal text defining it, and (2) the obligations that were previously extracted (see Table III). For example, V1 is the variation shown in Fig. 2. The third column indicates the actor that should be consulted for resolving the variation, i.e., EMS or EC.

The fourth column of Table III provides an intuitive textual description of the variation. Note that the description also covers how the underlying actor (in the third column) is likely to influence the interpretation and enforcement of the original GDPR rules. For example, in V16, the EC might publish a list of the territories and sectors that are deemed to offer an adequate level of protection for data transfer. At the moment the paper was written, data can be transferred within the same international organization to Switzerland without additional obligations. However, unconditional data transfer to Canada is only limited to commercial organizations under Canadian's PIPEDA law (Personal Information Protection and Electronic Documents Act). Other sectors and domains involve additional obligations that need to be fulfilled such as the approval of the lead supervisory authority.

The last column of Table III provides guidelines to analysts for: (1) when they should intervene to resolve the variation given the application context, and (2) how they can resolve the variability. For example, handling V16 is only warranted when the organization seeking compliance performs internal cross-border data transfers. One way to do so, is by updating the CM and the relevant constraints based on the most recent requirements for data transfer published by the EC.

The strategy we employ for resolving variability is "clone and own" [23], where the generic artifacts are specialized for the organization and system(s) at hand. Examples of changes to the artifacts include updating the cloned CM, glossary, and the rule and obligation encodings table. Further, analysts can add new OCL constraints, and drop or override existing ones. The only artifact from the first modeling step of our approach that remains unchanged is the variability table (Table III). This is because the variability table incorporates all envisagable variations with regard to the GDPR and is used as a checklist for guiding the analysis during the tailoring step. Concretely, analysts skim through the variability table and resolve the variations that apply to the underlying context.

An important challenge here is keeping track of the changes made for specializing the modeling artifacts. To do so, analysts have to record the actions they have taken to tailor the generic modeling artifacts. To illustrate, let us suppose that an organization X is an international commerce company located in Europe and Canada. Table IV presents an example of how the variability in Table III would be handled for X. The first column of Table IV references a particular variation listed in Table III, whereas the second column of Table IV lists the cloned generic artifacts that were impacted during the resolution of the variation. The final column of Table III describes how the artifacts in the second column were updated based on the specific context of X. X must account only for V16 in Table III. V1 in Table III does not apply to X since X only trades with subjects aged over 18 years old (clearly stated in X's privacy policy and website).

TABLE IV
EXCERPT OF VARIABILITY RESOLUTION TABLE

Ref.	Artifact	Summary of actions
V1	-	[Not applicable]
...	...	...
V16	Obligations table	A new obligation **ON1** was added. When a cross-border data transfer is based on an adequacy decision from the EC, data controllers must also conduct a DPIA (Data Privacy Impact Assessment).
V16	Specialized Model	The enumerations covering the territories and specific sectors that can receive personal data based on the EC adequacy decision was updated according to the EC website.
V16	OCL constraints	One constraint was overridden to state that: (internal) cross-border data transfer to Canada is allowed when the underlying organization is: 1) under PIPEDA and 2) conducting a commerce activity. Otherwise, transfer cannot be based on adequacy decision. One constraint was added to encode the new obligation ON1.
V16	Glossary	The description of the overridden constraint was updated. An intuitive description of the added constraint was inserted in the glossary.

As shown by Table IV, only variation V16 is relevant for X. In particular, the cloned CM, OCL constraints, glossary, and the table of obligations were altered as described in the last column of the table. For example, the OCL constraint that checks the lawfulness of cross-border data transfer was updated according to the adequacy decisions published by the EC (fifth row of Table IV). We also note that a new OCL constraint was added to encode a new (non-generic) obligation imposed by one of the EMS laws that are relevant to X (third row of Table IV). In short, X is also requested to conduct a DPIA (Data Privacy Impact Assessment) to be able to perform cross-border data transfer to Canada. Once all the pertinent variability points are resolved, analysts will have developed the contextualized (tailored) models as well as the variability resolution table (Table IV).

Due to space, we cannot present all the practical details of the tailoring process. However, we make two additional remarks. First, regardless of the changes made, the OCL constraints should remain correct with respect to the cloned CM. For example, if the analysts decide to drop the class *Consent* and its associations, then all impacted constraints have to be either corrected or dropped. Second, analysts might unintentionally introduce inconsistencies in the set of OCL constraints, e.g., two contradicting constraints. To avoid this, one can employ existing constraint solvers, e.g., UML2CSP [24], Alloy [24] or PLEDGE [25], to spot UNSAT

sets of constraints.

C. Challenges Encountered during the Modeling (RQ3)

In this section, we address RQ3 by listing the main challenges encountered when modeling the GDPR. Later, in Section VI, we present our vision for how we intend to address these challenges.

Specification of Compliance Rules (C1): In this paper, we first use OCL constraints to embed compliance rules in the generic model. We then adapt and expand these constraints to create a specialized model. We have already taken care of defining OCL constraints over the generic model; no additional effort is thus foreseen for this task. Nevertheless, additional effort, including by legal experts, will be required for defining OCL constraints over the specialized model, noting that these constraints necessarily refer to legal material (e.g., EU national laws, EU and national case law, and domain adaptations) that is more complex and fragmented than the GDPR. Due to the scarce familiarity of legal experts with OCL, the creation of the latter group of constraints may be difficult and time-consuming.

Rationale for Model Specialization (C2): Although we keep track of all the actions performed during the tailoring step, we do not systematically express the rationale behind the actions; in other words, we do not document why analysts made the decisions they did [26]. In the context of our work, the rationale needs to cover the problems the analysts encountered, the options they investigated, the GDPR provisions they examined to evaluate the options, and, most importantly, the arguments that led them to make certain decisions.

Generation of the Instance Model (C3): The process of generating an instance of a specialized model is currently dealt with manually (recall step 3 in Fig. 1). This would mean that a legal expert would have to create, by using a model editor, the instance. This manual process is time-consuming and tedious.

V. Lessons Learned

In this section, we discuss the lessons we learned from modeling the GDPR.

Streamline the validation process. We observed that modeling the GDPR, whether at a generic or specialized level, necessitates substantial legal knowledge and expertise that may go beyond the GDPR itself, e.g., knowledge of the articles of the WP as illustrated in Section IV-B. Thus, putting in place an effective and efficient validation process with legal experts was paramount to ensure that the produced artifacts were as complete and precise as possible. To achieve this goal, we had to shield the legal experts from the complexity arising from the CM and its underlying OCL constraints.

As discussed in Section IV-A, legal experts were able to grasp the CM with relative ease. This was in large part thanks to the intuitiveness of UML class diagrams and the fact that non-software experts can be quickly trained to obtain a working understanding of the notation for validation purposes. In contrast, OCL, which we use to formally express the GDPR rules and obligations, was challenging and intimidating to legal experts, despite our attempts to explain the meaning of the constraints. Similar communication barriers were observed when we attempted to replace OCL with other logical notations, e.g., standard first-order logic. In general, we believe such barriers are to be expected when formal logic is used directly with professionals who do not have adequate mathematical background. We mitigated this issue by describing each OCL constraint via an intuitive but precise textual description in natural language (see the glossary in Table I). Nevertheless, the glossary per se was still not enough to ensure reliable validation of the OCL constraints. In particular, the same rule or obligation is often expressed over smaller and modular sub-constraints. For example, the rule in the article of Fig. 2 was encoded over two constraints, namely consentProvider and VAR_DSAge in Fig. 5. The former constraint encodes the common part of the rule, whereas the latter covers the variable part. With the rules getting fragmented, legal experts experienced difficulties because they could no longer relate to the original obligation or rule. One way to remedy this problem is by forcing one-to-one mappings, where any GDPR obligation or rule is expressed using a unique OCL constraint. However, such a solution will further complicate the tailoring step, since variant requirements will have to be mixed with the fixed ones. This prompted us to use the obligation and rule encodings table (e.g., Table II) which traces the GDPR obligations and rules to their corresponding constraints. Both the glossary and the obligation and rule encodings table facilitated the validation of the OCL constraints by legal experts.

Although the validation of the CM was conducted package by package, legal experts still found the models to be overwhelming in terms of their information content. For example, the stereotypes we use for modeling variability at the level of the CM, i.e., «Optional», «Variation», and «Variant», were applied to systematize and automate the tailoring step for analysts. Exposing the legal experts to these stereotypes brought accidental complexity. The variability table (e.g., Table III) was enough to enable the legal expert to verify that the list of extracted variation points was complete and precise. A simple but effective solution was to support several views for the same CM, where the level of detail to display is configured according to needs. To this end, we found out that, in many situations, hiding class operations, attribute types, and stereotypes would be helpful. Further, to ensure that enough time was given for validation, we alternated on-line and off-line validation as discussed in the modeling methodology of Section IV-A.

Maintain traceability. Another observation from our GDPR modeling experience is that both analysts and legal experts often needed to consult specific articles to refresh their memory. Being able to do so effectively required all our modeling artifacts to be traceable to their corresponding GDPR provisions. Examples of traceability links can be seen in the second columns of the glossary (Table I), the obligation and rule encodings table (Table II) and the variability table (Table III). Although not shown in Fig. 4, classes too are traceable to the specific GDPR provisions pertaining to them

at the level of the CM. For example, the class *Purpose* in Fig. 4 is mapped to Arts. 5, 13, 14 and 15. These links made it easy to go back and forth between the modeling artifacts and the GDPR. We anticipate the links to be useful for other purposes as well, e.g., performing impact analysis when the GDPR, the EMS or the EC laws change (evolve). The only classes that were not mapped to the GDPR are those we have created to better structure the CM, e.g., the *Document* class.

A further final about traceability concerns the importance of maintaining consistent relationships between the different modeling artifacts. In practice, one often needs to quickly navigate from one artifact to another, in particular during the tailoring step. For example, when resolving a given variability, it is often useful to view the list of obligations and rules whose fulfillment is likely to be impacted by the EMS and EC laws. Similarly, analysts need to navigate to the underlying OCL constraints that need to be updated. Examples of such links can be found in the third column of Table II and the second column of Table III. We received positive feedback from the legal experts about having navigable artifacts.

Make the tailoring step as systematic as possible. During the tailoring step, we observed that even experienced analysts could encounter difficulties in resolving the variation points. The root cause of this was the large number and size of the modeling artifacts. This prompted us to develop simple guidelines to systematize and better organize the tailoring step. First, analysts have to go through the variation points and tick those that are relevant to their working context. Then, analysts can focus only on the relevant variation points and apply our recommendations on how to resolve them. This was facilitated by the "When & how to resolve?" column of the variability table. For example, when V1 in Table III is relevant to the context, analysts will get to know that they have to update V_getMinimumAgeForDS to account for the minimum age of children as regulated by the relevant EMS laws. However, we do not recommend a sequential resolution of variation points, e.g., first resolving V1, then V2, then V3, and so on. In particular, analysts should postpone completing the specification of the OCL constraints until all the variability for the CM has been handled. This is because some changes in the CM might break other constraints for which variability was previously resolved. To help analysts follow these recommendations, we proposed to keep track of all the tailoring actions in the resolution table (see Table IV). This facilitates resolving the variabilities in an incremental and non-sequential manner. In line with the above, a recent work from Hajri et al. proposes a tool-supported approach that guides analysts in configuring product specific models from product line models [27], [28]. In the future, we envisage to operationalize our tailoring recommendations by customizing Hajri et al.'s work.

Finally, we found the resolution table to be very useful when we had to deal with several similar contexts. In such cases, we started the tailoring from specialized modeling artifacts produced for other similar contexts, rather than from the generic artifacts. This, in our experience, helps to expedite the tailoring step.

VI. Future Directions

In this section, we describe the most important future directions that, we believe, are necessary for addressing the challenges identified in Section IV-C.

Domain-Specific Rule Language (C1). Using OCL constraints is key to achieving automation in checking GDPR compliance. In our approach, this is done via the specialized set of OCL constraints that encode the rules and obligations applying to a given context. Nevertheless, some of the specialized constraints, in particular, the new ones originating from the EMS laws, have to be validated by legal experts. As discussed in Section V, OCL impedes understandability by legal experts. To tackle this limitation and improve the tailoring of the generic model (Step 2 in Fig. 1), it would be advantageous to develop a Domain-Specific Rule Language (DSRL). The DSRL should, on the one hand, be expressive enough to be useful for the precise specification of GDPR compliance checking rules, and on the other hand, understandable enough to be readily used by legal experts. For example, the OCL rule presented in Fig. 5, would be hardly understood by most legal experts. To ease understandability, restricted natural language (NL) could be used as the basis for the DSRL. While basing the DSRL on NL increases usability, there is still the risk that legal experts may find it difficult to articulate their rules in a proposed language. To mitigate this issue, one needs to closely interact with legal experts during the DSRL design, and iteratively validate the language constructs with them. In addition, providing training material for the DSRL would be essential to make the language more accessible to non-software experts. Finally, to support automated compliance checking, the rules specified in the DSRL should be automatically translatable into OCL so that the rules can be checked directly over instantiations of a specialized GDPR model.

Goal Models (C2). Using goal models can help to deal with capturing and reasoning about the rationale for model specialization. Each goal is a prescriptive statement of intent that a system should satisfy [29]. Here, the term "system" refers to a combination of IT applications, organizations, workflows and people that together perform certain functions. A goal model is characterized by a collection of goals, the relationships (e.g., hierarchical decomposition) between the goals, and the obstacles that could hinder the satisfaction of the goals. Goal models provide a flexible instrument for arguing about model specialization. A key task related to a goal-oriented analysis of the GDPR would be to decide how the application context discussed in Section III (Step 2 in Fig. 1) should be decomposed and analyzed in order to tailor the specialized model. This decomposition necessarily involves breaking down the GDPR's core tenets (e.g., data minimization) into more tangible sub-goals. Additionally, one may need to decompose the goals of a given system (e.g., a specific organization), and examine how the system goals map

onto the goals stipulated by the GDPR. A main criterion to fulfill regarding goal decomposition would be to ensure that the decomposition process makes progress towards a set of concrete claims for which meaningful evidence about satisfaction (in term of model specialization) could be collected. Meeting this criterion necessitates that the developed goal models should provide a blueprint for the justification that is needed in order to argue about the adequacy and effectiveness of a proposed model specialization.

AI-enabled Automation Support (C3). Legal documents typically come in the form of NL descriptions. Mining these descriptions to identify the appropriate metadata to build the instance model is a prerequisite for automated compliance checking. Metadata items relevant to GDPR are numerous. Examples of such metadata include: "purpose" to mark the purposes of the processing for which personal data is being collected, "basis" to mark the legal basis for the processing of personal data, and "right to access" to mark the clause(s) giving an individual the right to request from the controller access to their personal data. These metadata items have to be identified in legal and technical documents such as privacy policies, consent statements, records of processing activities and exemptions, and data protection impact assessments. Natural Language Processing (NLP) [30] and Machine Learning (ML) [31] provide a useful technical platform for metadata extraction [21]. The metadata identified will be the basis for the model-based representation of the legal and technical documents to be checked. In other words, an automatic instantiation process will convert the metadata extracted with NLP and ML for a given document into a model-based representation, i.e., the instance of a specialized model. The elements of this instance model will be both fully traceable to the content of the source document as well as unambiguously mappable onto the underlying generic and specialized models.

VII. Related Work

In this section, we present related work on (1) modeling the GDPR, and (2) checking compliance.

Modeling the GDPR. Ayala-Rivera and Pasquale [18] propose a model-based approach to help organizations understand the data protection obligations imposed by the GDPR. Caramujo et al. [19] target privacy policies from the web and mobile applications, and propose a domain-specific language along with model transformations for specifying privacy-policy models. Pullonen and Matulevicius [20] present a multi-level model to be used as an extension of the Business Process Model and Notation (BPMN) to enable the visualization, analysis, and communication of the privacy-policy characteristics of business processes. Tom et al. [32] present a preliminary GDPR model aimed at providing a simple, visual overview so that process implementers can better understand the associations between different entities in the GDPR. The authors describe an approach for using their proposed model as a tool to develop an organizational privacy policy along with an illustration of compliance-rule extraction. These existing strands of work either address narrow analytical use cases (e.g., only the compliance analysis of privacy policies) or focus on providing guidelines for the (manual) application of the GDPR. We go beyond the existing work by modeling the GDPR in a more holistic way and providing a systematic tailoring mechanism to support GDPR compliance automation in different contexts.

Checking Compliance. To the best of our knowledge, no automated approach for checking GDPR compliance has been published so far. However, there are a few threads of work that describe methodologies for assessing system compliance.

Chung et al. [33] identify non-compliance issues in user-defined process models by matching these models against a standard model during both process specification and process execution. Panesar-Walawege et al. [5] propose a model-based approach to aid the suppliers of safety-critical systems in defining the evidence information necessary for certification according to standards and automatically detecting non-compliance issues in the collected evidence.

Ranise and Siswantoro [34] devise an SMT-based tool for checking compliance of security policies at design time. Guarda et al. [35] propose a logic-based framework to support the specification of information system designs, purpose-aware access control policies, and legal requirements.

While being a useful source of inspiration, none of the above approaches can be directly adapted to the GDPR due to their main focus being different than data protection and privacy.

VIII. Conclusion

In this paper, we reported on our experience modeling the General Data Protection Regulation (GDPR) using UML and OCL. The main motivation behind our work is to pave the way for developing automated, model-based GDPR compliance analysis solutions. Our work resulted in a generic GDPR model alongside a precise and documented strategy for specializing this model according to the application context and to suit the requirements of different types of GDPR-related analysis. Drawing on the experience gained from our modeling endeavor, we discussed a number of important lessons learned. We further proposed future directions to address the challenges we observed in our work, and to support a longer-term research agenda on model-based analysis of GDPR compliance.

In the future, we plan to work on the directions presented in Section VI in order to enable a full realization of the approach outlined in Section III. In addition, we will be working closely with legal experts on implementing a number of compliance analysis use cases, e.g., checking the compliance of privacy-policy statements. Doing so will not only enable us to identify and address high-priority automation needs, but will also help bridge the gap between software engineers and legal experts by developing more effective communication methods.

Acknowledgment

This project has received funding from Linklaters Luxembourg LLP.

REFERENCES

[1] European Union, "General data protection regulation," *Official Journal of the European Union*, 2018. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

[2] EU-GDPR. (2019) EU GDPR portal. [Online]. Available: https://eugdpr.org

[3] C. Tankard, "What the GDPR means for businesses," *Network Security*, vol. 6, pp. 5–8, 2016.

[4] European Union, "The GDPR: New opportunities, new obligations," *Justice and Consumers*, 2018.

[5] R. K. Panesar-Walawege, M. Sabetzadeh, and L. C. Briand, "Supporting the verification of compliance to safety standards via model-driven engineering: Approach, tool-support and empirical validation," *Information and Software Technology*, vol. 55, no. 5, pp. 836–864, 2013.

[6] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*, 2nd ed. Morgan & Claypool Publishers, 2016.

[7] R. France and B. Rumpe, "Model-driven development of complex software: A research roadmap," in *Proceedings of 2007 Workshop on the Future of Software Engineering (FOSE '07)*, 2007, pp. 37–54.

[8] OMG, "Unified Modeling Language - Superstructure Version 2.5.1," 2017. [Online]. Available: https://www.omg.org/spec/UML/2.5.1/PDF

[9] ——, "Object Constraint Language - Version 2.4," 2017. [Online]. Available: https://www.omg.org/spec/OCL/2.4/PDF

[10] S. Ghanavati, A. Rifaut, E. Dubois, and D. Amyot, "Goal-oriented compliance with multiple regulations," in *Proceedings of 22nd IEEE International Conference on Requirements Engineering (RE'14)*, 2014, pp. 73–82.

[11] S. Ingolfo, A. Siena, and J. Mylopoulos, "Nòmos 3: Reasoning about regulatory compliance of requirements," in *Proceedings of 22nd IEEE International Requirements Engineering Conference (RE'14)*, 2014, pp. 313–314.

[12] N. Zeni, N. Kiyavitskaya, L. Mich, J. R. Cordy, and J. Mylopoulos, "GaiusT: Supporting the extraction of rights and obligations for regulatory compliance," *Requirements Engineering*, vol. 20, no. 1, pp. 1–22, 2015.

[13] G. Soltana, N. Sannier, M. Sabetzadeh, and L. C. Briand, "Model-based simulation of legal policies: Framework, tool support, and validation," *Software & Systems Modeling*, vol. 17, no. 3, pp. 851–883, 2018.

[14] C. Arora, M. Sabetzadeh, L. C. Briand, and F. Zimmer, "Extracting domain models from natural-language requirements: Approach and industrial evaluation," in *Proceedings of the 19th IEEE/ACM International Conference on Model Driven Engineering Languages and Systems (MoDELS'16)*, 2016, pp. 250–260.

[15] W. Emmerich, A. Finkelstein, C. Montangero, S. Antonelli, S. Armitage, and R. Stevens, "Managing standards compliance," *IEEE Transactions on Software Engineering*, vol. 25, no. 6, pp. 836–851, 1999.

[16] T. Breaux, "Exercising due diligence in legal requirements acquisition: A tool-supported, frame-based approach," in *Proceedings of 17th IEEE International Conference on Requirements Engineering (RE'09)*, 2009, pp. 225–230.

[17] N. Sannier, M. Adedjouma, M. Sabetzadeh, and L. C. Briand, "An automated framework for detection and resolution of cross references in legal texts," *Requirements Engineering*, vol. 22, no. 2, pp. 215–237, 2017.

[18] V. Ayala-Rivera and L. Pasquale, "The grace period has ended: An approach to operationalize GDPR requirements," in *Proceedings of 31st IEEE International Conference on Requirements Engineering (RE'18)*, 2018, pp. 136–146.

[19] J. Caramujo, A. Rodrigues da Silva, S. Monfared, A. Ribeiro, P. Calado, and T. Breaux, "RSL-IL4Privacy: A domain-specific language for the rigorous specification of privacy policies," *Requirements Engineering*, vol. 24, no. 1, pp. 1–26, 2019.

[20] P. Pullonen, J. Tom, R. Matulevicius, and A. Toots, "Privacy-enhanced BPMN: Enabling data privacy analysis in business processes models," *Software & Systems Modeling*, pp. 1–30, 2019.

[21] A. Sleimi, N. Sannier, M. Sabetzadeh, L. C. Briand, and J. Dann, "Automated extraction of semantic legal metadata using natural language processing," in *Proceedings of 26th IEEE International Requirements Engineering Conference (RE'18)*, 2018, pp. 124–135.

[22] T. Ziadi and J.-M. Jezequel, "Product line engineering with the UML: Deriving products," in *Software Product Lines*. Springer, 2006.

[23] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*. Addison-Wesley, 2001.

[24] J. Cabot, R. Clarisó, and D. Riera, "UMLtoCSP: A tool for the formal verification of UML/OCL models using constraint programming," in *Proceedings of the 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE'07)*, 2007, pp. 547–548.

[25] G. Soltana, M. Sabetzadeh, and L. C. Briand, "Practical model-driven data generation for system testing," *arXiv preprint (arXiv:1902.00397)*, 2019. [Online]. Available: https://arxiv.org/pdf/1902.00397.pdf

[26] S. B. Shum and N. Hammond, "Argumentation-based design rationale: What use at what cost?" *International Journal of Human-Computer Studies*, vol. 40, no. 4, pp. 603–652, 1994.

[27] I. Hajri, A. Göknil, L. C. Briand, and T. Stephany, "Configuring use case models in product families," *Software & Systems Modeling*, vol. 17, no. 3, pp. 939–971, 2018.

[28] I. Hajri, A. Goknil, L. C. Briand, and T. Stephany, "PUMConf: A tool to configure product specific use case and domain models in a product line," in *Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE'16)*, 2016, pp. 1008–1012.

[29] A. van Lamsweerde, *Requirements Engineering - From System Goals to UML Models to Software Specifications*. Wiley, 2009.

[30] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT Press, 2001.

[31] E. Alpaydin, *Machine Learning: The New AI*. MIT Press, 2016.

[32] J. Tom, E. Sing, and R. Matulevičius, "Conceptual representation of the GDPR: Model and application directions," in *Perspectives in Business Informatics Research*, 2018, pp. 18–28.

[33] P. W. Chung, L. Y. Cheung, and C. H. Machin, "Compliance flow – Managing the compliance of dynamic and complex processes," *Knowledge-Based Systems*, vol. 21, no. 4, pp. 332–354, 2008.

[34] S. Ranise and H. Siswantoro, "Automated legal compliance checking by security policy analysis," in *Computer Safety, Reliability, and Security (SAFECOMP'17 Workshops)*, 2017, pp. 361–372.

[35] P. Guarda, S. Ranise, and H. Siswantoro, "Security analysis and legal compliance checking for the design of privacy-friendly information systems," in *Proceedings o 22nd ACM on Symposium on Access Control Models and Technologies (SACMAT'17)*, 2017, pp. 247–254.