ARTICLE

Validation of GDPR Compliance in a Library Management System

A BISIS and TeMDA Case Study

Tijana Rajačić, Danijela Boberić-Krstićev, Danijela Tešendić, and Gordana Milosavljević

ABSTRACT

This paper explores the challenges of achieving General Data Protection Regulation (GDPR) compliance in library management systems (LMSes) by integrating our novel TeMDA framework into BISIS. BISIS is an LMS used in more than 60 libraries in Serbia. We employed a case study conducted in collaboration between the selected BISIS and TeMDA developers, all authors of this paper. We maintained and presented a detailed development diary to provide insights for other developers seeking GDPR compliance in LMSes. The study provides a practical solution for LMSes to ensure GDPR compliance with minimal effort. The description of the development process, accompanied by listings, tables, and diagrams, can assist LMS developers in determining whether the proposed approach is suitable for them. To the best of our knowledge, this paper presents the first detailed case study on integrating a GDPR compliance framework into an LMS.

INTRODUCTION

Digitization has become crucial in recent decades, enabling the storage of vast amounts of data in digital formats that are more efficient for processing and analysis. However, data privacy and security have emerged as critical concerns, particularly regarding electronic resources in libraries.[1] As information is primarily accessed and stored electronically, ensuring the confidentiality, integrity, and availability of library resources is essential. Library management systems (LMSes) often collect and store members' data, including search histories, usage patterns, and personal information. To safeguard data privacy and comply with regulations like the General Data Protection Regulation (GDPR) in the European Union (EU), libraries must implement robust privacy policies and security measures to protect user data and maintain trust.[2]

The GDPR aims to enhance data privacy for individuals in the EU, granting rights such as clear consent for data processing, easier access to personal data, rectification and erasure requests, the right to be forgotten, objection to profiling, and data portability.[3] It also addresses the transfer of personal data to non-EU countries and international organizations. In libraries, members have the right to know what data is collected, know how it is used, and maintain control over their data, including its access, correction, deletion, or restriction of processing.

About the Authors

Tijana Rajačić (tijana.lalosevic@uns.ac.rs) is PhD student, Faculty of Technical Sciences, University of Novi Sad, Serbia. **Danijela Boberić-Krstićev** (dboberic@uns.ac.rs) is Associate Professor, Faculty of Sciences, University of Novi Sad, Serbia. **Danijela Tešendić** (tesendic@uns.ac.rs) is Associate Professor, Faculty of Sciences, University of Novi Sad, Serbia. **Gordana Milosavljević** (grist@uns.ac.rs; corresponding author) is Professor, Faculty of Technical Sciences, University of Novi Sad, Serbia.

Submitted: 24 March 2025. Accepted for Publication: 23 July 2025. Published: 15 December 2025.

https://doi.org/10.5860/ital.v44i4.17407

During software development, security is often managed by engineers who may lack adequate expertise, leading to a limited understanding of security and privacy concepts.[4] The repeated failures of software systems to protect user privacy highlight the need to examine development practices and understand why these efforts frequently fall short.[5] Such flaws compromise user privacy and damage the reputation and market value of the organizations involved.[6] Violators of data protection regulations may face fines of up to €20 million (about $23 milion) or 4% of their annual global turnover.[7]

Misunderstandings of security concepts contribute to vulnerabilities in 78% of projects.[8] Engineers are required to swiftly acquire new knowledge and adapt to rapidly evolving technologies. This highlights the necessity for tools and frameworks that enable quick understanding of advanced concepts and their integration into software solutions.

Our novel TeMDA framework enables the dynamic validation of business processes for GDPR compliance using model-driven engineering (MDE). The name TeMDA combines Themis, the Greek goddess of justice, and model-driven approach (MDA). Currently, it supports Java projects.

In MDE, models at a high abstraction level play a central role in implementation, as they are automatically converted to code using custom code generators. The modeling process may utilize domain-specific languages (DSLs) tailored to specific business or technical domains.[9]

TeMDA's foundation is a DSL that incorporates GDPR concepts and Object Constraint Language (OCL) rules with embedded legal expert knowledge to automatically validate software project compliance with GDPR.[10] The DSL employs a minimal set of meta-classes to facilitate easy learning while enabling the validation of real scenarios involving personal data usage. During TeMDA's development, we collaborated with legal experts to clarify GDPR concepts, model legal terms as meta-model elements, and specify and validate OCL rules.

To simplify integration into software projects and eliminate the need to learn new tools or develop formal security policy models, the DSL's concrete syntax is based on Java annotations, allowing easy integration with the existing toolchain. The development team annotates their code at all points where data management occurs. Aspect-oriented programming (AOP) is used to dynamically generate a privacy model by using aspects to read annotations and annotated code elements (user-defined types of interest for GDPR validation and methods that create instances of these types).[11] AOP enables the ability to add behavior to existing projects without modifying code, making TeMDA applicable to both new and existing projects. The generated policy model is validated using OCL rules, allowing automatic tests to be executed with each project change. If errors occur, OCL rules prevent breaches, enabling developers to experiment and learn by doing.

To assess the challenges of achieving GDPR compliance in LMSes, we integrated TeMDA into BISIS, which is used in more than 60 public, faculty, and some specialized libraries in Serbia. BISIS includes cataloging, bibliography reports, circulation, online public access catalog (OPAC), bibliographic data interchange, and administration (https://www.bisis.rs/).

This paper provides a comprehensive overview of the development process, including code listings (https://data.mendeley.com/datasets/5ncspt2rg9/1), tables, and unified modeling language (UML) diagrams. It outlines the changes required to integrate a GDPR framework into BISIS and shares the lessons learned. To the best of our knowledge, there is currently no detailed case study or source code available on implementing GDPR frameworks in LMSes.

The paper is organized as follows: Section 2 discusses related work, Section 3 introduces TeMDA, Section 4 presents a detailed case study of the integration of BISIS with TeMDA, and the concluding section summarizes the findings and outlines directions for future work.

BACKGROUND AND RELATED WORK

Our research focuses on two areas: (1) data privacy requirements in LMSes and (2) tools and frameworks for ensuring GDPR compliance in software projects. These areas are discussed in the following subsections.

Data Privacy in Library Management Systems

Ethical principles guiding libraries, as outlined in the International Federation of Library Associations and Institutions' Code of Ethics, emphasize the importance of respecting privacy and protecting personal data exchanged between libraries and users.[12] Libraries must comply with GDPR regulations, prioritizing the safeguarding of sensitive user data and protecting intellectual property. Strong data governance frameworks, transparent data practices, and mechanisms for obtaining user consent are essential for compliance with evolving privacy laws.[13] Robust data privacy measures in an LMS not only protect user information but also build trust and ensure legal compliance.

A survey of academic and special libraries in the United Kingdom revealed that privacy was not a top priority for librarians; only 14% had privacy notices, and 64% had data protection policies. Most Data Protection Officers lacked training, with 52% reporting no organizational changes since the 1988 Data Protection Act and 40% unsure of the current status of their policies.[14]

Vavousis et al. interpreted GDPR articles and their impact on the LMS in the National Library of Greece.[15] They emphasized Article 4, which defines key terms; Article 6, which outlines principles for processing personal data; and Articles 7 and 8, which introduce the concepts of consent and children's consent, respectively. Finally, they focused on Article 12, underscoring the importance of transparent communication regarding data collection purposes.

Katulić et al. revealed that 60% of European national libraries have a privacy page, and 53% of EU Member States publish the required data protection information, compared to 47% of nonmembers.[16] Among the 27 libraries with privacy pages, 24 include notices detailing the purposes and legal bases for data processing, reflecting responsible efforts.

GDPR Tools and Frameworks

The complexity of GDPR lies in its comprehensive coverage of all data manipulation stages, including collection, storage, transformation, transport, processing, and suspension of processing. It emphasizes the roles and responsibilities of data owners. Implementing GDPR compliance in software systems is a significant undertaking that requires careful attention to detail. Neglecting specific data flows can lead to substantial penalties in the event of a data breach. Several solutions have been developed to provide tools and frameworks that streamline the process and mitigate the risk of oversight.

Ayala-Rivera and Pasquale introduced the GuideMe approach, a six-step process to derive solution requirements from GDPR obligations.[17] These steps include Data Audit, Gap Analysis, Planning and Preparation, Plan Review, Execution, and Post-Implementation Review. While practical, it only partially addresses GDPR requirements, focusing on Articles 5 through 25. Additionally, familiarity with GDPR terminology is required, which can be a limitation in real-world scenarios where

developers often lack such expertise. Our solution employs terminology that is accessible to non-legal professionals and covers the first 50 articles of the GDPR, providing a more comprehensive framework.

Building on earlier work, Ayala-Rivera et al. developed the So&Co solution, which helps engineers identify and integrate appropriate technical controls into sequence diagrams.[18] So&Co includes a catalog of planned solutions and templates, streamlining the design process. Its use of annotations and AOP aligns closely with our framework, further validating the effectiveness of these techniques for achieving GDPR compliance.

Several solutions leverage blockchain technology to implement GDPR concepts like consent, using smart contracts to securely manage and store consent information.[19] Daudén-Esmel et al. propose a blockchain-based solution with two contracts: Consent Smart and Purpose Smart Consent.[20] These contracts feature rich meta-models and offer various methods, enabling efficient management of consents and their purposes. However, our solution focuses on privacy by design and a DSL to describe data access compliance within the system.

Vanezi et al. present a graphical DSL for defining GDPR purposes, using a notation similar to Business Process Model and Notation (BPMN).[21] However, it focuses solely on purposes and lacks support for key concepts like consent, erasure, rectification, and complaints. Our model addresses these gaps by supporting all crucial GDPR concepts.

Michael et al. proposed a privacy meta-model with a graphical DSL and a code generator.[22] The core concept, PrivacyPolicyRule, allows users to define rules for data usage, collectors, time periods, and storage locations, with a notable focus on spatial context. While it introduces the purpose of data use, it lacks support for complaints, rectifications, and the suspension of data processing.

Caramujo et al. developed a flexible DSL for specifying privacy policies using Xtext (https://eclipse.dev/Xtext/).[23] Designed for cloud use, it introduces the concept of Enforcement, enabling rules such as restricting service usage for individuals under the age of 13. However, it lacks support for adding new legal documents such as complaints and consents and does not account for the time dimension critical for GDPR compliance. Our model includes these features.

Sánchez et al. evaluate application terms of use with machine learning, quantifying GDPR compliance levels.[24] Their approach helps users decide whether to accept terms based on compliance grades, classifying policies as “good” or “bad.”

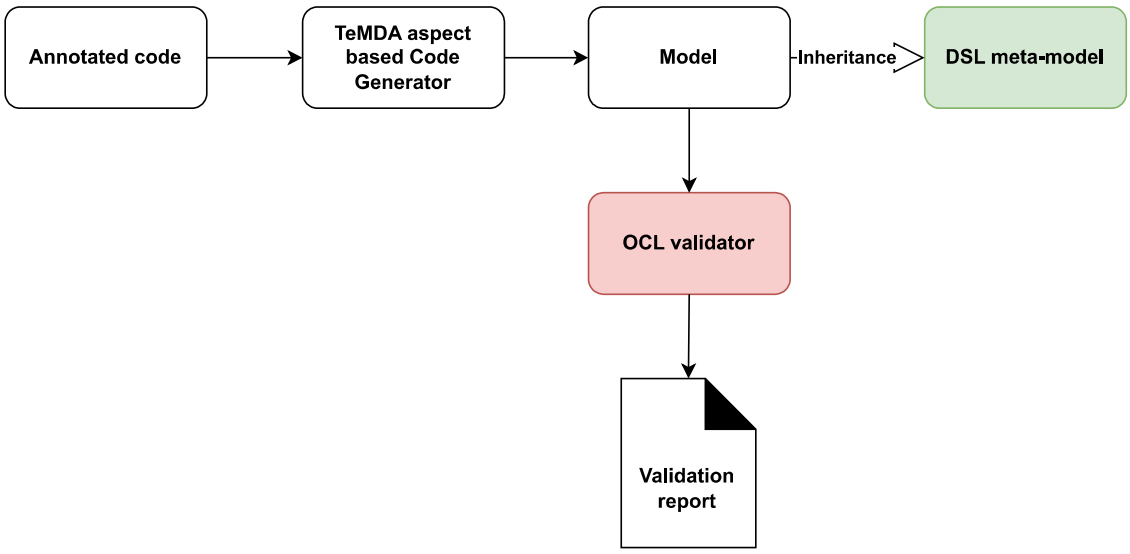
Torre et al. introduced a detailed GDPR meta-model developed with legal experts, addressing a wide range of GDPR acts.[25] Legal terms are modeled as meta-model instances validated with OCL. While the DSL supports property rights, supervisory authorities, liabilities, and penalties, its complexity makes it difficult to use. Additionally, it is unsuitable for defining security policies in cloud systems.

THE TEMDA FRAMEWORK

TeMDA consists of (1) a DSL meta-model definition with OCL rules and (2) a DSL concrete syntax based on Java annotations and aspects to handle annotated programming code. The first component defines the building blocks of the privacy model and is responsible for validation, serialization, and framework configuration. The second component dynamically generates a GDPR policy model by analyzing the annotated code. Their details are presented in the following

subsections. The architecture is illustrated in Figure 1. The TeMDA source is available at https://github.com/Tijana994/PrivacyDSLv3 (meta-model and validation rules) and https://github.com/Tijana994/TeMDA (aspects and annotations).

Figure 1. Overview of the TeMDA architecture.
DSL is domain-specific language, and OCL is object constraint language.



Our main objective was to integrate GDPR principles into software in a developer-friendly manner, enabling automatic validation of GDPR compliance. Many existing solutions require additional effort, such as learning complex languages for privacy model definition[26] and using auxiliary tools that are not standard in a development cycle, like those requiring XML or UML models as input.[27] Some tools also require a multi-step process.[28]

We aimed to design a framework that simplifies developers' roles by allowing easy annotation of their solutions, with validation procedures executed during runtime. This approach reduces cognitive load and operational demands on developers, streamlining the validation process and improving usability in real-world scenarios.

TeMDA Meta-model Overview

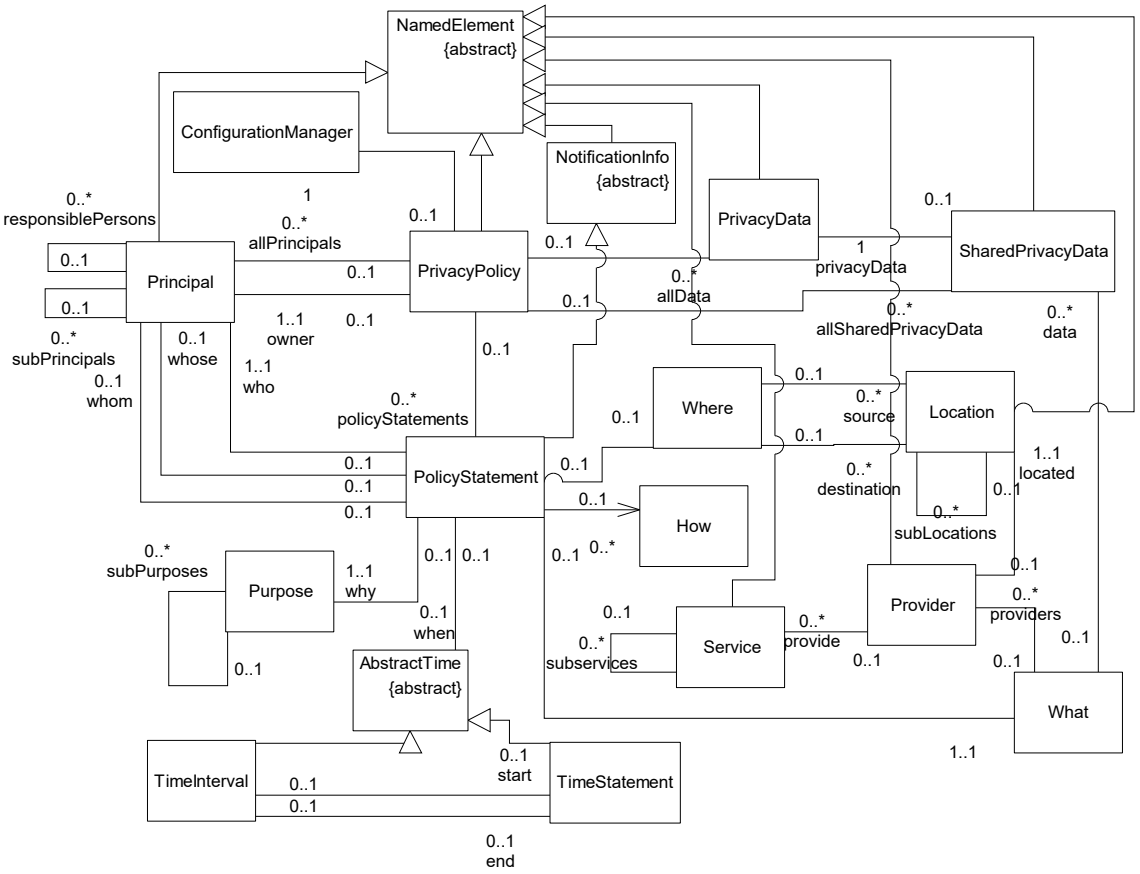
When developing a DSL, it's essential to consider the context, purpose, and intended audience. Our aim was to create an intuitive DSL for developers. The DSL vocabulary features terms commonly used in information systems, irrespective of business domain. We defined meta-classes for key GDPR concepts and specified OCL rules with embedded legal knowledge to ensure compliance. A comprehensive set of enumerated types allows developers with limited legal expertise to select from predefined values, enabling informed decisions.

The meta-model is developed using the Eclipse Modeling Framework, Ecore (https://eclipse.dev/modeling/emf/), and OCL (https://www.omg.org/spec/OCL). It consists of 29 meta-classes and 14 enumerated types. The meta-classes are presented in Figures 2 and 3. Examples of enumerated types can be found in Figure 4. An example of an OCL rule supporting GDPR Article 6 is given in Listing 1 (https://data.mendeley.com/datasets/5ncspt2rg9/1).

The *NamedElement* meta-class is an abstract meta-class that provides a unique name for all its descendants (see Figure 2).

The *Principal* meta-class models the user. Enumerated type *PrincipalScope* classifies the *Principal* meta-class as either an in-scope company member, an out-of-scope external person, or an undefined group (Figure 4).

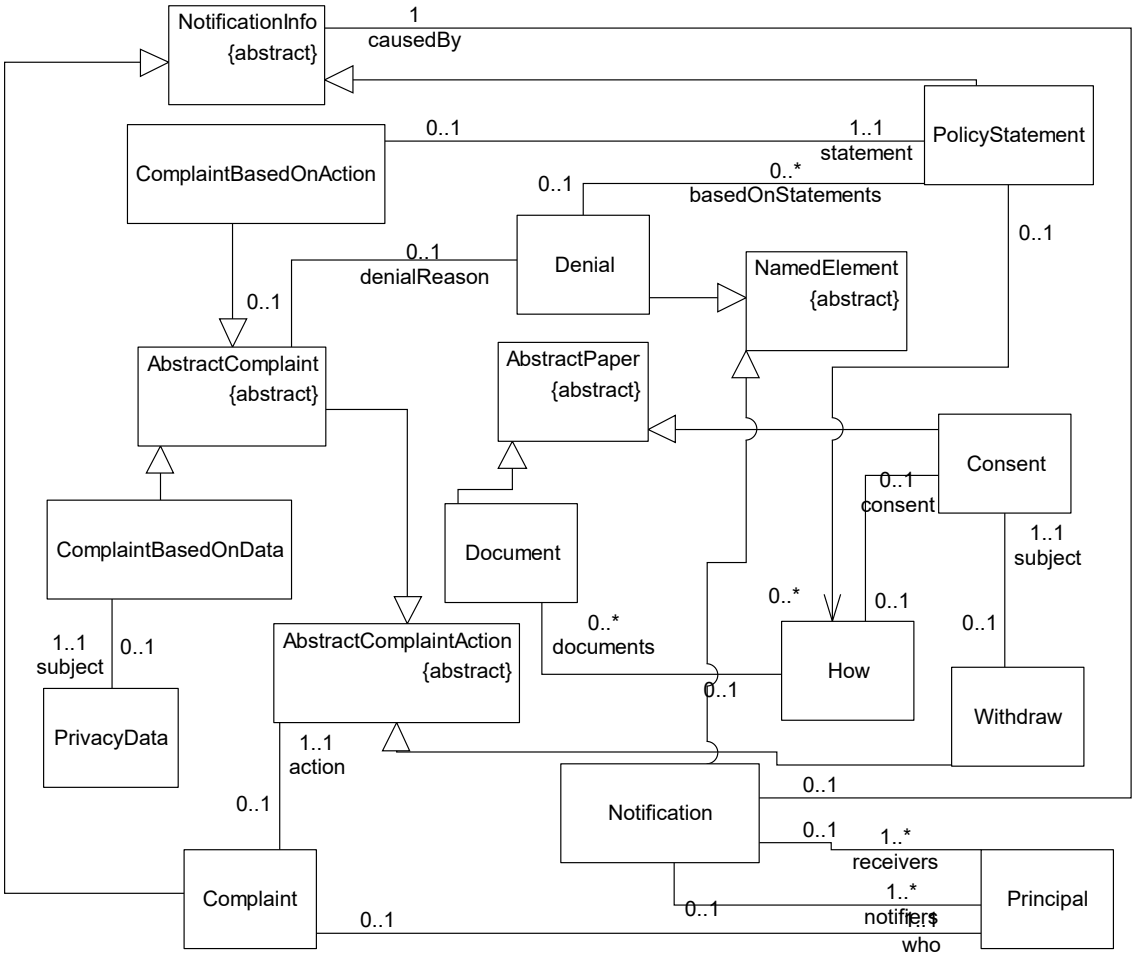
Figure 2. The first part of the TeMDA meta-model. The central meta-classes are *Principal* and *PrivacyPolicy*. The *PrivacyPolicy* meta-class includes a list of *PrivacyStatement* instances that define data access details.



The *PolicyStatement* meta-class describes user actions within a specific period. To enhance clarity in the DSL, its attribute names employ natural language prepositions and adverbs. The *PolicyStatement* meta-class contains three associations with the *Principal meta-class*: who performs the actions (data controller), to whom it belongs (data owner), and, in the case of data transfer, to whom the data is transferred (third-party data processor). Additionally, the *PolicyStatement* meta-class specifies what actions should be performed, as well as when, where, why, and how.

The *What* meta-class defines the actions performed within the system and supports its operation in a cloud infrastructure, allowing us to specify which providers offer which services. If the cloud infrastructure organizes services into subgroups, we can describe them using subservices. Compliance with GDPR also requires defining the provider's location, particularly in relation to data transfers outside the EU.

Figure 3. The second part of the TeMDA meta-model addresses complaints, notifications, and documents.

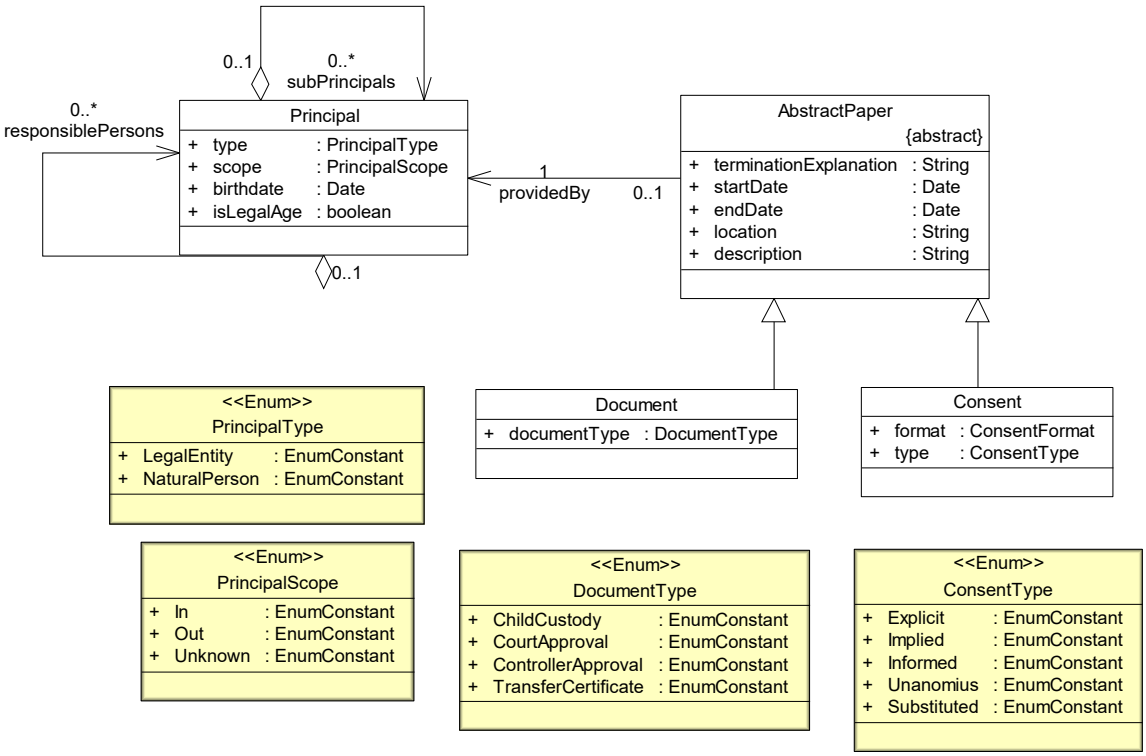


Time is defined through the abstract meta-class *AbstractTime* and its descendants, which represent either a specific moment (*TimeStatement*) or a time interval (*TimeInterval*; see Figure 2).

The *How* meta-class describes the documents that participants attach to perform a specified action. The *AbstractPaper* meta-class abstracts any *Document*. The *Document* meta-class inherits from *AbstractPaper* and provides information about the document type.

The *Where* meta-class provides information about the source and destination of the user's action location through the *Location* meta-class, which defines a specific geographical area. The *Purpose* meta-class contains information on why the user accesses privacy data.

Figure 4. Part of the detailed TeMDA meta-model for specifying types of documents for a user (*Principal*).



The *PrivacyData* meta-class represents the processed personal data within the business system and its type.

The *SharedPrivacyData* meta-class provides information about security mechanisms and supports Article 14, which states that every piece of data in the system must have a defined origin and include information about data sources.

The *Consent* meta-class inherits from the *AbstractPaper* meta-class. According to GDPR, we support written, verbal, and nonverbal types of consents.[29] They are further categorized as explicit, implicit, informed, unanimous, and substituted.

The *Complaint* meta-class describes types of user complaints as outlined in the GDPR, including their timing and reasons. To align with data owners' rights, we introduced an abstract meta-class, *AbstractComplaintAction*, as a parent for the *Withdraw* meta-class and the *AbstractComplaint* meta-class. The *AbstractComplaint* meta-class contains the complaint status. Its subclasses are *ComplaintBasedOnAction* (defining the right to restrict data processing) and *ComplaintBasedOnData* (specifying the rights to rectify and erase data) meta-classes.

The NotificationInfo meta-class is an abstract class that generalizes the actions used by the system to create user notifications, categorized as either data processing-related or complaint-related. The Notification meta-class includes the date, time, and type of the notification, as well as the recipients, senders, and the action that caused it (the *causedBy* reference). The *NotificationType* enumeration represents the notification types described in Articles 19 and 33. Both *PrivacyStatement* and *Complaint* meta-classes inherit from the *NotificationInfo* meta-class.

TeMDA Aspects and Annotations

AOP is a programming paradigm that enhances modularity by enabling the separation of cross-cutting concerns. It allows new functionalities, such as logging or security, to be introduced without modifying the original codebase. This is achieved by marking specific parts of the code with pointcuts, designated locations where additional functionalities should be integrated. Each functionality is implemented as advice within the aspect, specifying whether it executes before, after, or instead of the code marked by the pointcut.[30]

To simplify the creation of a privacy model for developers, we designed a solution that integrates into their existing toolchain and development environments without requiring additional artifacts or external tools. Java annotations are used to map the data model already present in their code to the TeMDA privacy model. These annotations are part of the concrete syntax of the TeMDA DSL and are based on the vocabulary introduced by the DSL's meta-model.

The aspects analyze the annotated code sections and dynamically generate the privacy model. They essentially perform a model-to-model transformation from the user-defined data model to the privacy model based on the TeMDA meta-model, using Java reflection to access objects at runtime. The privacy model is then utilized for OCL validation.

The DSL supports (1) concept and (2) creator annotations. The concept annotations define the mapping of user-defined data type properties onto TeMDA meta-class properties. The creator annotations mark class methods that create instances of user-defined data types for GDPR validation. The annotation name specifies the aspect that monitors the creation of the data type instance and its mapping to the TeMDA meta-class instance in the privacy model. Creator annotation attributes can be applied to a class property, a method parameter, a property of the method's return value, or a property of the method's parameter object. Each nonabstract TeMDA meta-class has its own aspect dedicated to creating its instance and setting its property values based on the specified annotation values. These annotations are presented in detail in the following sections.

CASE STUDY: INTEGRATION OF TEMDA INTO THE BISIS LMS

The BISIS LMS has been in development since 1993 at the University of Novi Sad (UNS), Serbia, and is implemented in more than 60 libraries in Serbia. Its primary modules include cataloguing, reporting, circulation, OPAC, bibliographic data interchange, and administration. BISIS supports cataloguing according to UNIMARC and MARC 21 formats using an XML editor for bibliographic material processing.[31] The circulation module manages member-related activities, including registering, charging, discharging, searching for members and publications, generating different types of reports, and sending user reminders.[32] The functionalities of this module are critical in the context of GDPR due to the handling of personal data, exemplified by the member registration process.

BISIS is implemented in Java and follows a client/server architecture, with the server side built using the Spring framework (https://spring.io/projects/spring-framework), MongoDB (https://www.mongodb.com), and ElasticSearch (https://www.elastic.co/elasticsearch). Client applications are developed as desktop, web, and mobile applications.

Our primary goal was to adapt BISIS to incorporate key GDPR concepts such as consent management and clear purposes for data collection and usage. Additionally, we aimed to enable BISIS to validate every type of data access, ensuring robust compliance and data protection.

Case Study Settings

To apply TeMDA to BISIS, two selected developers from BISIS and two from TeMDA were involved. The BISIS developers have experience in web development, software modeling, and design, along with an awareness of GDPR. The TeMDA developers have experience in GDPR, software modeling and design, and MDE.

To navigate the complexities of making the LMS GDPR compliant, the developers maintained a detailed development diary throughout the process. Integrating BISIS with TeMDA required two development phases and four meetings.

The first phase involved analyzing BISIS's compliance with GDPR. It included two one-hour meetings with BISIS and TeMDA developers. The first meeting featured participant introductions and a brief demonstration of TeMDA functions using a demo project. The second meeting focused on analyzing BISIS to identify areas needing compliance validation.

The second phase involved integrating BISIS with TeMDA over the next two meetings, lasting 5.5 hours and 2.5 hours, respectively.

In the third meeting, the BISIS developers began configuring and annotating the BISIS source code with assistance from the TeMDA developers. They encountered some challenges while using TeMDA and provided crucial feedback, which was promptly addressed. This feedback facilitated a more intuitive handling of annotations and offered flexibility to define certain concepts in alternative ways.

The fourth meeting enabled the efficient integration of BISIS with the improved TeMDA framework.

Analysis of BISIS Compliance with GDPR

A GDPR data action refers to any operation or request related to the handling of personal data under GDPR. Therefore, the first step is to recognize and document all types of personal data that BISIS collects, processes, and stores, along with the purposes for their use. Additionally, data flows within the organization and with external parties must be documented.

The results of mapping key GDPR data actions to functionalities in BISIS are presented in Table 1. Data collection in BISIS occurs only during the registration of a new member or when a member lends books. A librarian may need to access data for managing membership or loans, tracking overdue items, reserving books, sending reminders for returns, notifying members about overdue fines, sending notifications about upcoming events, and generating reports. The data transfer action is not applicable to BISIS, as there is no exchange of personal data with other parties. A librarian can delete or update data upon the member's request, which relates to the erasure and rectification actions.

Data should be collected and processed only for clearly defined and lawful purposes, which must be communicated to members. Table 2 shows how data in BISIS is mapped to the purposes defined by TeMDA. In BISIS, data is primarily accessed for operational functionalities, research and statistical purposes to generate reports, and marketing purposes when notifying members about upcoming events.

Table 1. Mapping the General Data Protection Regulation (GDPR) data actions to BISIS functionalities.

GDPR Data Action	Mapped Functionality in BISIS
Collecting	Registration of a library member (minor/adult); lending books
Access	Membership management; loan management; communication; reporting
Transfer	Not applicable
Erasure	Delete member's data
Rectification	Update member's details

Table 2. Mapping reason types defined in TeMDA to BISIS functionalities.

Purpose—Reason	BISIS Functionality
Research	Reporting
OutOfScope	Membership management; loan management
StatisticalPurposes	Reporting
ExercisingSpecificRights	Not applicable
Marketing	Sending notifications
Testing	Not applicable
Profiling	Not applicable
StopProcessing	Delete user data
PublicInterest	Access user data
ProtectTheVitalInterests OfTheDataSubject	Not applicable
LegitimateInterests	Not applicable
PublicHealth	Not applicable

Implementing GDPR in BISIS

This section describes the implementation of GDPR in BISIS, highlighting the member registration process as an example. Figure 5 illustrates the process of opening an account for a new library member. When an individual wants to join, they visit the library, where the librarian collects personal information (first name, last name, email address, ID number, username, and gender) to complete a registration form. This data is stored in BISIS for internal purposes: managing book lending, membership renewals, and marketing activities, including push notifications about book promotions and events.

This approach raises potential GDPR violations. Article 13 requires data controllers to inform data subjects about the purposes of data collection. Failing to notify members about the use of their data for marketing purposes risks noncompliance. Additionally, processing a minor's data without obtaining consent from a parent or guardian violates Article 8, which mandates explicit parental consent for data processing involving children.

Figure 5. Member registration diagram in BISIS before the General Data Protection Regulation compliance analysis.

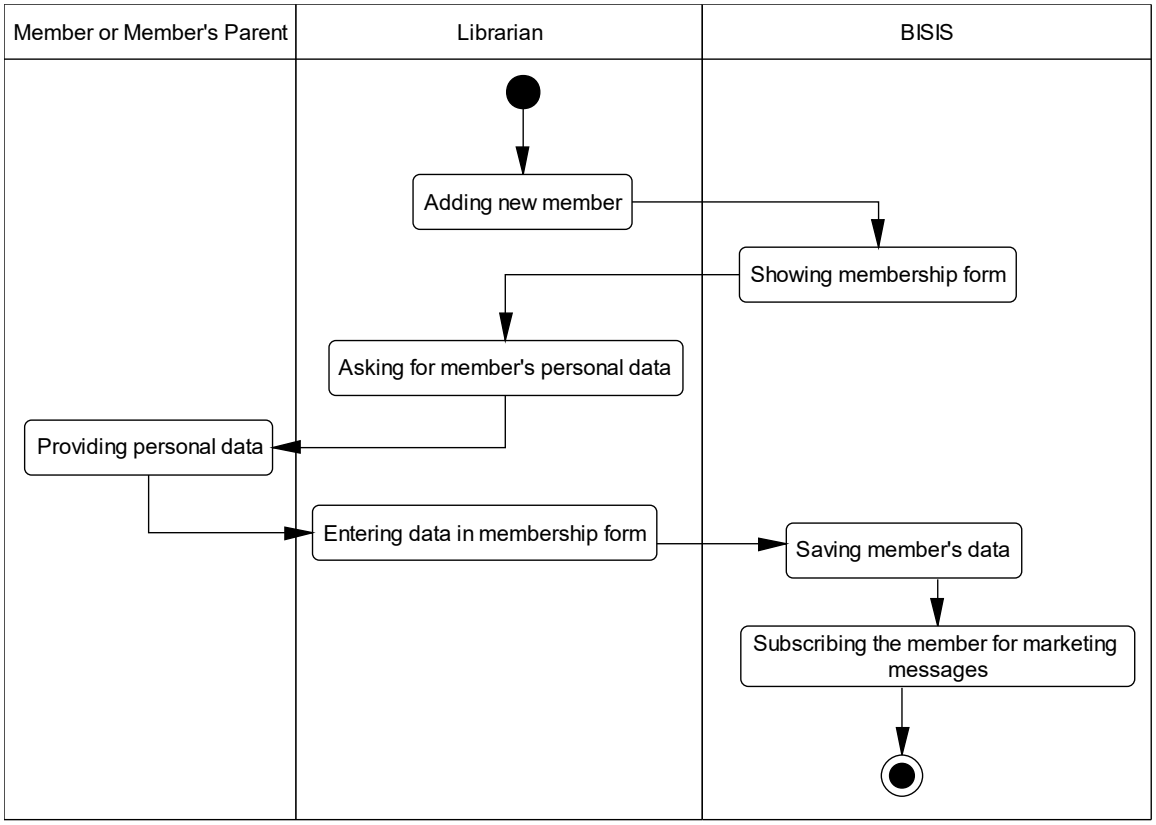


Figure 6 illustrates the correct process according to GDPR for opening an account for a new library member. After collecting the necessary data, the system determines if the individual is a minor. If not, they are asked to consent to the use of their personal data within BISIS and to agree to receive marketing communications. For minors, a parent or legal guardian must provide the required consents before the account can be created.

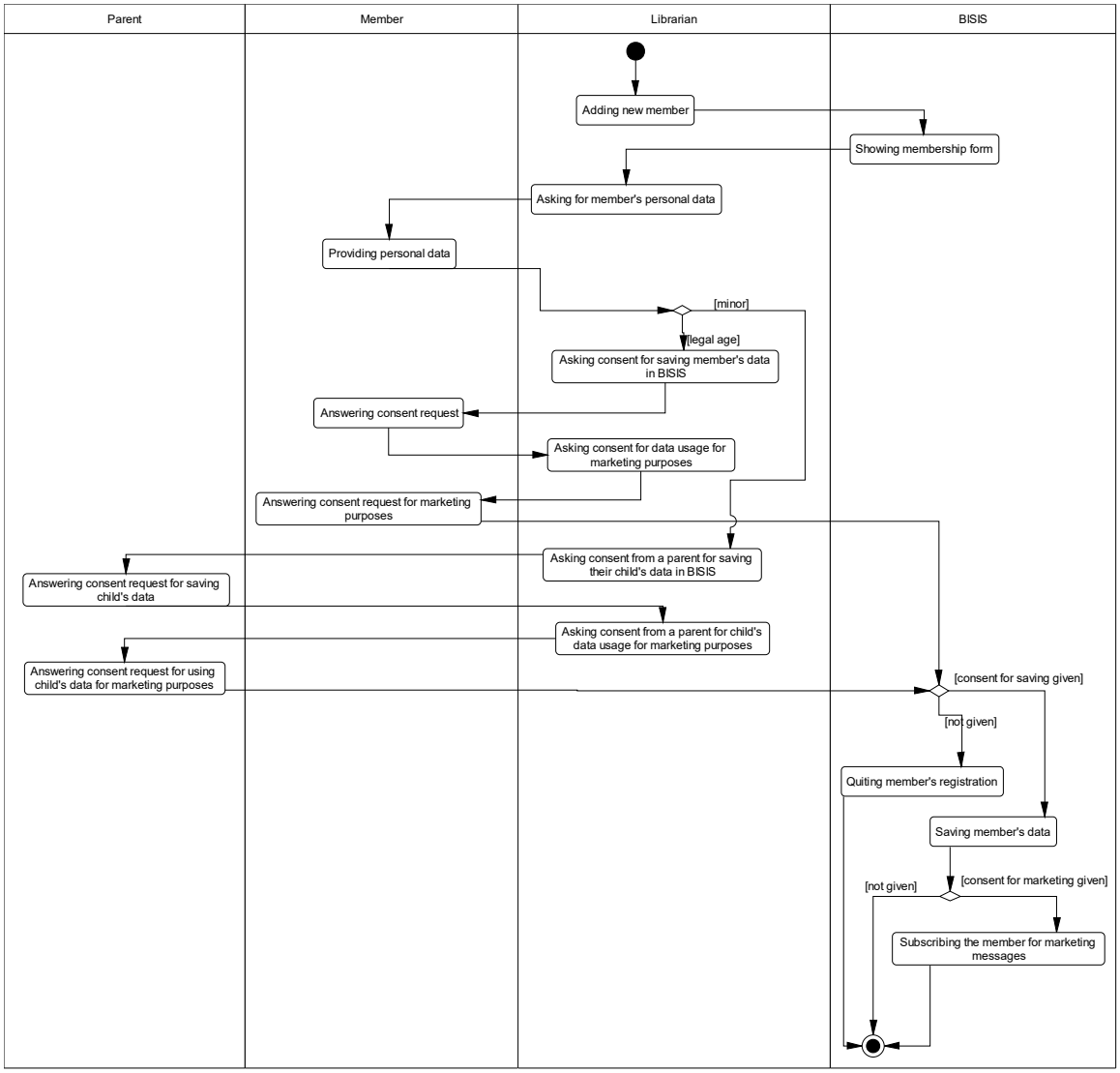
After obtaining consent for data storage, the system stores the data and verifies whether consent has been provided for marketing purposes. If marketing consent is granted, the system subscribes the member to the marketing service, enabling them to receive notifications.

To adjust BISIS to this proposed sequence of actions, we performed several modifications:

- Configured TeMDA within BISIS
- Adjusted and annotated the BISIS source code, followed by conducting GDPR validation
- Modified the BISIS user interface

We will illustrate our process using program code listings, all of which are available at https://data.mendeley.com/datasets/5ncspt2rg9/1.

Figure 6. Member registration diagram after the General Data Protection Regulation compliance analysis.



Configuring TeMDA within BISIS

In this phase, the developers integrated TeMDA references within BISIS, introducing the *TemdaConfiguration* class (Listing 5) as the central configuration point for key privacy settings. They defined the storage location for the privacy model generated by TeMDA aspects, configured the privacy policy name, and identified data sources, including publicly available sources, identity documents, and internal information. Then they specified protection control methods for the data, such as encryption, for each data type individually and set a default method to ensure a baseline level of security. This configuration approach enables customized data protection strategies that align with business needs and ensure compliance with data privacy regulations.

BISIS Source Code Annotation, Refactoring, and Validation

After configuring the application, TeMDA can be initialized by starting BISIS. Even with just the initial configuration and no annotations entered, TeMDA sends validation messages via a log file.

The validation phase identifies inconsistencies between the software and GDPR compliance requirements using TeMDA's 76 OCL rules, which are dynamically executed at runtime. They

detect compliance gaps and alert developers to obligations that must be fulfilled promptly to avoid potential compliance failures.

Validation errors indicated that the owner of the privacy policy (the company) and the geographic location of the policy were not defined, as shown in Listing 3. Geographic location is crucial for GDPR compliance, as it determines the territorial scope under Article 3 of the regulation. Analyzing the errors and the BISIS code revealed that interventions were needed in both BISIS and TeMDA to effectively perform the mapping.

To load data from the database, the developers initially planned to annotate the methods responsible for database loading and the corresponding data types. However, this would introduce significant complexity to BISIS, requiring a separate method for each entity. The developers identified a more optimal solution by introducing a helper class, *DataFactory*, to TeMDA to instantiate objects without using annotations.

TeMDA initially expected the existence of a legal entity representing the privacy policy owner, along with data on its employees. This required adding new entities to the BISIS model. Instead, TeMDA leveraged the existing BISIS *LibraryConfiguration* entity and mapped it with annotations to represent the Privacy Owner. The developers also connected employees who were not logically linked to the configuration in the existing BISIS model.

After these interventions, the project addressed the validation errors and annotated the code sections identified during the analysis phase (Section 4.2), along with the relevant parts of the data model. This was achieved using creator and concept annotations, as documented in the provided listings.

The concept annotations define the mapping of user-defined data type properties onto TeMDA meta-class properties. They were used to annotate parts of the BISIS data model relevant for GDPR. Figure 7 presents a segment of the data model extracted from the BISIS source code, specifying the members. Listing 4 illustrates the usage of concept annotations, which specify the mapping of the TeMDA meta-class *Principal* to the BISIS application data class *Member*. Detailed information about the *Principal* meta-class can be found in Figure 4.

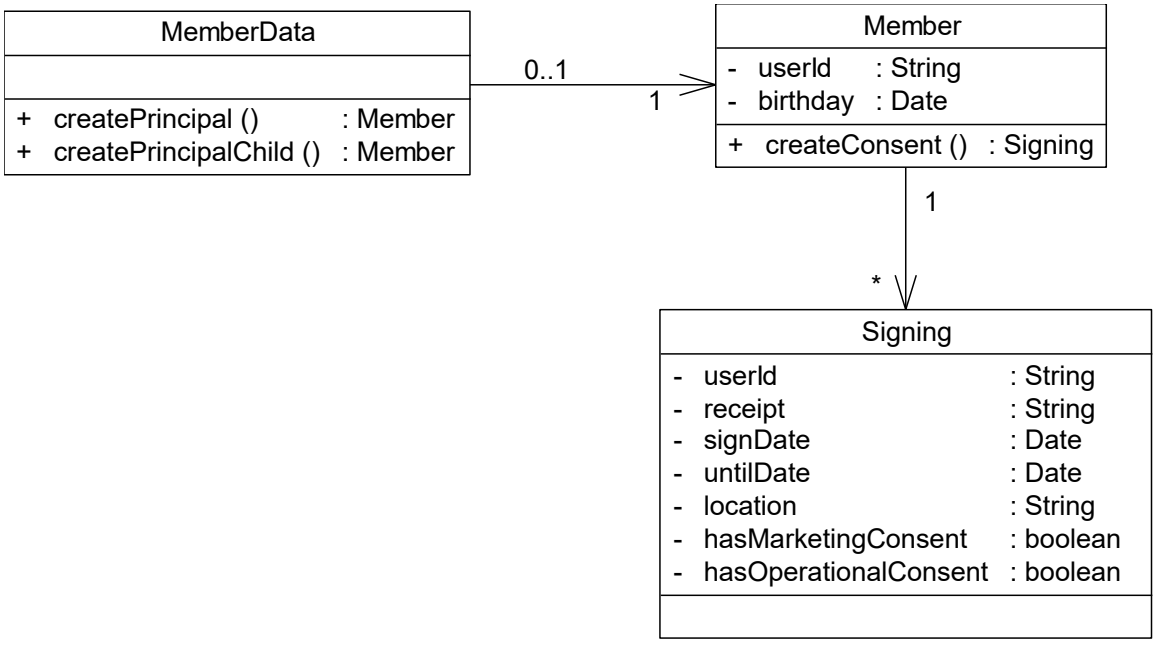
The creator annotations mark class methods that instantiate user-defined data types for GDPR validation. These annotations were used to annotate the BISIS API endpoint methods. Examples are provided in Listing 5, where two methods of *MemberData* are utilized to register library members.

Next, the actions of BISIS identified in Table 1 were addressed. These methods, located within controllers, were annotated with *CreatePolicyStatementAnnotation* (Listing 6).

Upon the next launch, a validation message indicated the absence of consent when validating the *CreatePolicyStatement* annotation (Listing 7). Another message stated that creating a policy statement was impossible due to the lack of the *PolicyStatementAnnotation* or the *systemActionId* parameter. This feedback enabled developers to address the missing GDPR concepts of consent and purpose of data usage. Although consent was added, the *CreatePolicyStatement* annotation still required the addition of a purpose parameter.

After analysis, it was determined that extending BISIS with a new concept was unnecessary. Instead, the existing *Signing* class (Figure 7) was extended and mapped to the TeMDA *Consent* meta-class (Figure 8).

Figure 7. A segment of the data model, extracted from the source code of the BISIS application specifying library members. Consents for data processing that members should provide are added to the *Signing* class.



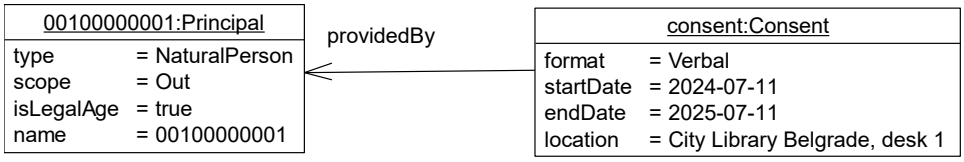
BISIS developers identified a method in the *Member* class that creates consent upon requests from library members, which was then annotated with *CreateConsentAnnotation* to trigger the execution of the consent aspect (Listing 5).

After resolving initial validation messages and running BISIS again, it was identified that a defined reason for accessing data, referred to as the Purpose, was missing. This was addressed using the *DataFactory* class.

Once the software was annotated and launched, the TeMDA aspects dynamically generated a privacy model. Real-time validation then ensured that the generated model adhered to GDPR requirements, verifying that data handling practices respected the rights of the data owner (Listing 8). If an error arises during data management, TeMDA automatically logs the issue, highlighting instances of data usage that deviate from GDPR compliance.

The part of the privacy model dynamically generated by TeMDA aspects, based on Listings 4, 5, and 9, is presented in Figure 8. The model is populated with annotation values specified in the code and the runtime property values of BISIS application class instances.

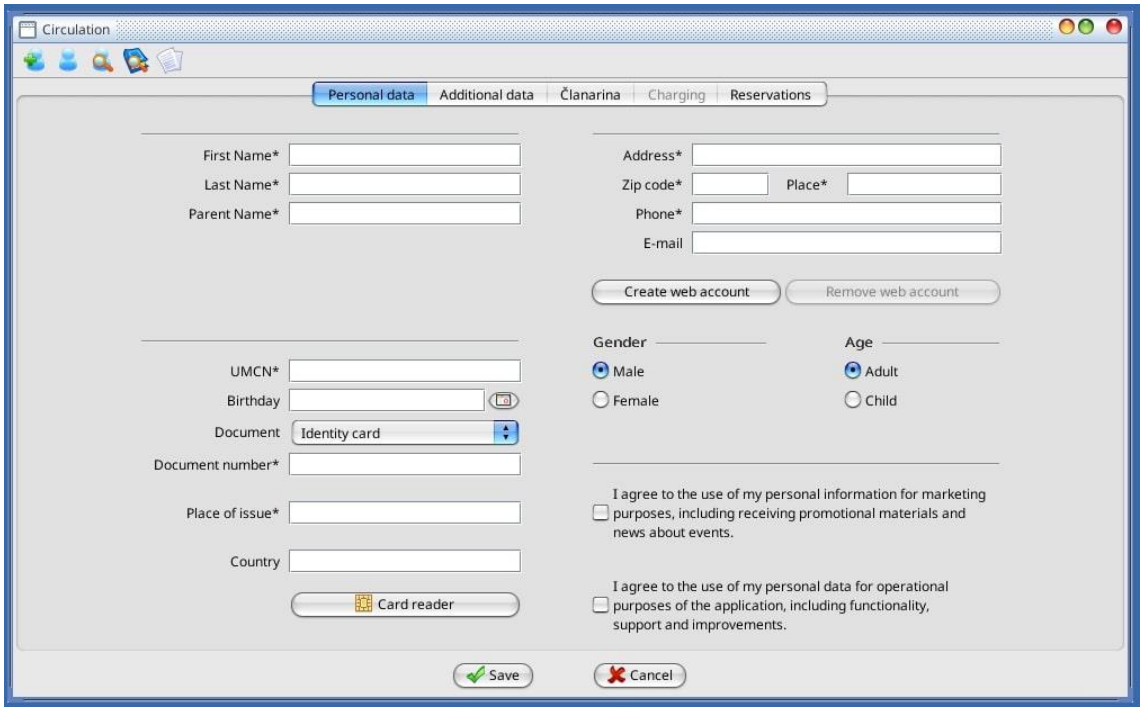
Figure 8. Part of a privacy model that is dynamically generated by TeMDA aspects. The elements in the model are instances of the TeMDA meta-classes. The model is populated with annotation values specified in the code, as well as with the runtime property values of BISIS application class instances. Object Constraint Language rules are used to validate its General Data Protection Regulation compliance.



BISIS User Interface Adjustment

Finally, the BISIS developers adjusted the member registration form to allow librarians to specify the purposes for using member data. Members can grant or deny consent for different purposes (Figure 9). Additionally, when "Child" is selected in the age field, the parent field becomes mandatory to capture consent from the parent on behalf of the child.

Figure 9. A new design for the BISIS member registration form now facilitates the entry of mandatory data to ensure compliance with the General Data Protection Regulation.



Realtime Validation

After integration, TeMDA can detect real-time noncompliance with GDPR. As data is accessed, the system checks for any violations related to the specific data owner. Two use cases were evaluated: registering a minor's profile without parental consent and accessing an adult's data for marketing purposes.

In the first use case, a minor who had previously registered without parental consent came to rent a book. The system generated a validation message indicating that the profile lacked valid consent for data collection, the minor's consent had expired, and parental consent was required (Listing 10). The first two messages were triggered by the expired consent, while the third clarified that the parent must provide the necessary consent.

In the second use case, access to an adult member's data for sending a notification about an ongoing book promotion was evaluated. Upon accessing the data, the system returned a validation message indicating that the reason for accessing the data was not well defined (Listing 11).

CONCLUSION

This paper presents a case study of integrating TeMDA framework within the BISIS LMS to ensure compliance with GDPR. The project aimed to adapt BISIS to meet GDPR requirements by incorporating concepts such as consent management and data access validation. To the best of our

knowledge, there is no detailed case study or source code available regarding the implementation of GDPR in LMSes.

BISIS is used in more than 60 libraries in Serbia. Its primary modules include cataloging, bibliographic reports, circulation, OPAC, bibliographic data interchange, and administration.

TeMDA automates GDPR validation with minimal effort required from developers. It is based on a DSL that supports GDPR concepts, accompanied by OCL rules incorporating legal expertise. Its vocabulary is designed to be comprehensible to developers by using terms commonly found in information systems. By utilizing Java annotations and AOP, TeMDA streamlines the integration of privacy measures into software without necessitating separate formal models.

The case study was conducted by two BISIS developers and two TeMDA developers. The integration required two development phases and four meetings. The first phase focused on analyzing BISIS's compliance with GDPR:

- Identifying actions requiring compliance validation, such as data collection (member registration, lending books), data access (membership management, notifications), and data erasure/rectification
- Mapping these actions to GDPR principles like consent management, purpose specification, and data access control

The second phase involved the following:

- Small adjustments to both BISIS and TeMDA to facilitate their integration
- Configuring and annotating the BISIS source code to define mappings to GDPR concepts
- Modifying the BISIS UI to allow the provision of consents for different purposes
- Performing real-time validation of selected use cases

The first phase included two one-hour meetings with developers from both BISIS and TeMDA. The second phase included two meetings lasting 5.5 hours and 2.5 hours, respectively.

BISIS's GDPR compliance was achieved relatively quickly, in just four meetings, due to direct communication between the BISIS and TeMDA developers, who possess a deep understanding of their systems and have the authority to modify them. Close collaboration helped navigate the complexities of ensuring BISIS's compliance with data protection regulations, documenting our experiences for future reference. This collaboration also proved beneficial for TeMDA by increasing its flexibility.

The results benefit not only BISIS but may also serve as a valuable case study for similar systems seeking GDPR compliance. This highlights the importance of cooperation and ongoing dialogue between software developers and GDPR tool developers. Additionally, selecting the right tool or framework for enforcing GDPR policies is crucial. A framework like TeMDA, which includes embedded legal knowledge and supports easy integration with existing projects, enables GDPR validation without requiring developers to write complex legal rules and with minimal changes to the core BISIS logic. While many GDPR compliance tools focus on static analysis, TeMDA enables runtime validation, helping to prevent unintentional violations.

The availability of annotated BISIS code examples contributes to the body of knowledge on implementing GDPR frameworks in LMSes, filling a gap in the existing literature.

Currently, TeMDA supports the development of GDPR policies for software projects built in Java, with future plans to expand support to other programming languages, such as C#.

ENDNOTES

[1] Rudrani Saha, "Data Privacy and Cyber Security in Digital Library Perspective: Safe Guarding User Information," *International Journal of Scientific Research in Engineering and Management* 8, no. 4 (2024): 1–6, https://doi.org/10.55041/IJSREM30761.

[2] Saha, "Data Privacy."

[3] "The General Data Protection Regulation," European Council, accessed June 6, 2024, https://www.consilium.europa.eu/en/policies/data-protection/data-protection-regulation/#application.

[4] Irit Hadar et al., "Privacy by Designers: Software Developers' Privacy Mindset," *Empirical Software Engineering: An International Journal* 23, no. 1 (2018): 259–89, https://doi.org/10.1007/s10664-017-9517-1.

[5] Oshrat Ayalon, Eran Toch, Irit Hadar, and Michael Birnhack, "How Developers Make Design Decisions about Users' Privacy: The Place of Professional Communities and Organizational Climate," in *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (New York: ACM, 2017), 135–38, https://doi.org/10.1145/3022198.3026326.

[6] Senarath Awanthika and Nalin A. G. Arachchilage, "Why Developers Cannot Embed Privacy into Software Systems? An Empirical Investigation," in *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018* (New York: ACM, 2018), 211–16, https://doi.org/10.1145/3210459.3210484.

[7] "The General Data Protection Regulation."

[8] D. Votipka et al., "Understanding Security Mistakes Developers Make: Qualitative Analysis from Build It, Break It, Fix It," in *29th USENIX Security Symposium* (USENIX Security, 2020), 109–126, https://www.usenix.org/conference/usenixsecurity20/presentation/votipka-understanding.

[9] Thomas Stahl et al., *Model-driven Software Development: Technology, Engineering, Management* (Wiley, 2006).

[10] "Object Constraint Language (OCL)," Object Management Group, accessed May 3, 2024, https://www.omg.org/spec/OCL.

[11] Gregor Kiczales et al., "Aspect-Oriented Programming," in *ECOOP'97—Object-Oriented Programming*, *11th European Conference, Jyväskylä, Finland, June 9–13, 1997, Proceedings*, ed. Mehmet Aksit and Satoshi Matsuoka (Berlin: Springer, 1997), 220–42.

[12] "IFLA Statement on Privacy in the Library Environment," International Federation of Library Associations and Institutions, accessed November 21, 2024, https://www.ifla.org/wp-content/uploads/2019/05/assets/hq/news/documents/ifla-statement-on-privacy-in-the-library-environment.pdf.

[13] Saha, "Data Privacy."

[14] Paul Sturges et al., "User Privacy in the Digital Library Environment: An Investigation of Policies and Preparedness," *Library Management* 24, nos. 1–2 (2003): 44–50, https://doi.org/10.1108/01435120310454502.

[15] Konstantinos Vavousis et al., "Text and Data Mining for the National Library of Greece in Consideration of Internet Security and GDPR," *Qualitative and Quantitative Methods in Libraries* 9, no. 3 (2020): 441–60.

[16] Anita Katulić, Tihomir Katulić, and Ivana Hebrang Grgić, "Application of the Principle of Transparency in Processing of European National Libraries Patrons' Personal Data," *Digital Library Perspectives* 38, no. 4 (2022): 399–411, https://doi.org/10.1108/DLP-11-2021-0097.

[17] Vanessa Ayala-Rivera and Liliana Pasquale, "The Grace Period Has Ended: An Approach to Operationalize GDPR Requirements," in *2018 IEEE 26th International Requirements Engineering Conference (RE)* (IEEE, 2018), 136–46, https://doi.org/10.1109/RE.2018.00023.

[18] Vanessa Ayala-Rivera, A. Omar Portillo-Domínguez, and Liliana Pasquale, "GDPR Compliance via Software Evolution: Weaving Security Controls in Software Design," *Journal of Systems and Software* 216 (2024): 112144, https://doi.org/10.1016/j.jss.2024.112144.

[19] Masoud Barati and Omer Rana, "Tracking GDPR Compliance in Cloud-Based Service Delivery," *IEEE Transactions on Services Computing* 15, no. 3 (2020): 1498–511, https://doi.org/10.1109/TSC.2020.2999559; Yangheran Piao, Kai Ye, and Xiaohui Cui, "A Data Sharing Scheme for GDPR-Compliance Based on Consortium Blockchain," *Future Internet* 13, no. 8 (2021): 217, https://doi.org/10.3390/fi13080217.

[20] Cristòfol Daudén-Esmel et al., "Lightweight Blockchain-Based Platform for GDPR-Compliant Personal Data Management," in *2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP)* (IEEE, 2021), 68–73, https://doi.org/10.1109/CSP51677.2021.9357602.

[21] Evangelia Vanezi et al., "DiálogoP—A Language and a Graphical Tool for Formally Defining GDPR Purposes," in *Research Challenges in Information Science* (Springer, 2020), 569–75, https://doi.org/10.1007/978-3-030-50316-1_40.

[22] Judith Michael et al., "Towards Privacy-Preserving IoT Systems Using Model Driven Engineering," in *Proceedings of MDE4IoT Workshop @ MODELS 2019* (dblp, 2019).

[23] João Caramujo et al., "RSL-IL4Privacy: A Domain-Specific Language for the Rigorous Specification of Privacy Policies," *Requirements Engineering* 24 (2019): 1–26, https://doi.org/10.1007/s00766-018-0305-2.

[24] David Sánchez, Alexandre Viejo, and Montserrat Batet, "Automatic Assessment of Privacy Policies under the GDPR," *Applied Sciences* 11, no. 4 (2021): 1762, https://doi.org/10.3390/app11041762.

[25] Damiano Torre et al., "Using Models to Enable Compliance Checking against the GDPR: An Experience Report," in *2019 ACM/IEEE 22nd International Conference on Model Driven*

Engineering Languages and Systems (MODELS) (IEEE, 2019), 1–11, https://doi.org/10.1109/MODELS.2019.00-20.

[26] Tore, "Using Models."

[27] Michael, "Towards Privacy."

[28] Piao, "A Data Sharing Scheme."

[29] "General Data Protection Regulation, 2018," EU Legislative, Intersoft Consulting, accessed May 27, 2024, https://gdpr-info.eu.

[30] Kiczales, "Aspect-Oriented Programming."

[31] Bojana Dimić and Dušan Surla, "XML Editor for UNIMARC and MARC 21 Cataloguing," *The Electronic Library* 27, no. 3 (2009): 509–28, https://doi.org/10.1108/02640470910966934.

[32] Danijela Tešendić, Branko Milosavljević, and Dušan Surla, "A Library Circulation System for City and Special Libraries," *The Electronic Library* 27, no. 1 (2009): 162–86, https://doi.org/10.1108/02640470910934669.