Design Patterns for GDPR-Aware Process Modeling in BPMN

Abstract

In an increasingly digital world, collecting, processing, and exchanging personal data are considered critical drivers for enacting enterprise business processes (BPs). However, the long-term retention and access of personal data expose organizations to data breaches, in which sensitive and protected data are disclosed and exploited in an unauthorized fashion. To mitigate the damage that data breaches can cause, in the European Union (EU), the right to data privacy is enforced through the General Data Protection Regulation (GDPR), which defines how organizations must store and manage EU citizens' data. GDPR is highly influencing how organizations approach data privacy, forcing them to rethink and upgrade their BPs to become GDPR compliant, which can be daunting. In this paper, in line with the privacy-by-design principles of GDPR, we propose a methodology that shows how to capture the main privacy GDPR constraints in the form of design patterns and integrate them into BP models specified in BPMN (Business Process Model and Notation). This allows us to achieve full transparency of privacy constraints in BPs, making it possible to ensure their compliance with GDPR at design time. We adopt a design science research approach to present our methodology and make design decisions explicit.

Keywords: Data Privacy, GDPR, Design Patterns, Process Models, BPMN.

1. Introduction

Nowadays, the increase in both storage and processing power has made it possible to store and process virtually all the information that might be of interest for an organization to rapidly deliver digital and physical services to their customers (e.g., the creation of a bank account, the management of a purchase order, etc.). As [18] pointed out, the seemingly never-ending collection of customers' data by large corporations such as Google and Facebook has raised public awareness on *privacy* concerns.

Since 25 May 2018, General Data Protection Regulation (GDPR) tackled in the European Union (EU) the *right to privacy* for personal data, intending to protect EU citizens from privacy breaches. Since organizations that are not compliant with GDPR must face heavy fines, they must implement the GDPR data management policies correctly and take appropriate actions on data when requested by their customers. Among a list of technical and non-technical challenges to address [5], to achieve compliance with GDPR, the regulation enforces organizations to reshape the way they approach the management of personal data stored and exchanged during the execution of their business processes (BPs) [7]. A BP is a collection of activities required for delivering either a service or a product to a customer while accomplishing an organizational goal. BPs can be abstractly modeled via specific languages, such as ISO/IEC 19510:2013 BPMN (Business Process Modeling and Notation). Although [19] showed that BP modeling in BPMN is well-suited for expressing stakeholder collaboration and the data flow exchanged between BP activities, little has been done so far to tackle potential privacy breaches in BP models.

Conversely, the common practice to address privacy breaches in a BP is to implement ad-hoc countermeasures (e.g., in the form of scripts or business rules) during the automation stage of the BP life-cycle, when the BP model is configured by a system engineer (SE) for its execution with a dedicated BP Management System (BPMS). However, this approach requires that the SE knows precisely where potential privacy breaches can manifest in the BP. This information, if not explicitly documented in the BP model, may lead to a defective implementation of compensatory strategies from privacy breaches. As BPMN can explicitly mark and indicate data artifacts involved in the BP, we can directly pinpoint the privacy issues that BP might suffer without extending the modeling language. Despite works by Maines et al. [14], Maines et al. [15] and Chergui and Benslimane [9] explicitly extend BPMN with cyber-security requirements, such extensions are not readily perceived by business analysts, which are the customary users of BP models. Furthermore, our attempt is to handle generic BP descriptions that could be immediately implemented via customary BPMN technologies.

Based on the foregoing, we advocate that privacy should be considered a first-class citizen in BP models and should be introduced by design and not as an afterthought. Under the assumption that the BP has been correctly modeled in BPMN, in this paper, we present a methodology that shows how to capture the main privacy GDPR constraints in the form of design patterns and integrate them into BP models defined in BPMN. This paper emphasizes awareness of privacy concerns in BPs at design time when a proper analysis of the involved data allows a BP designer to prevent (possible) violations of privacy constraints by mitigating their impact. Our solution targets BP designers or Data Controllers as primary users for such BP models, as they will be in charge of (i) modeling

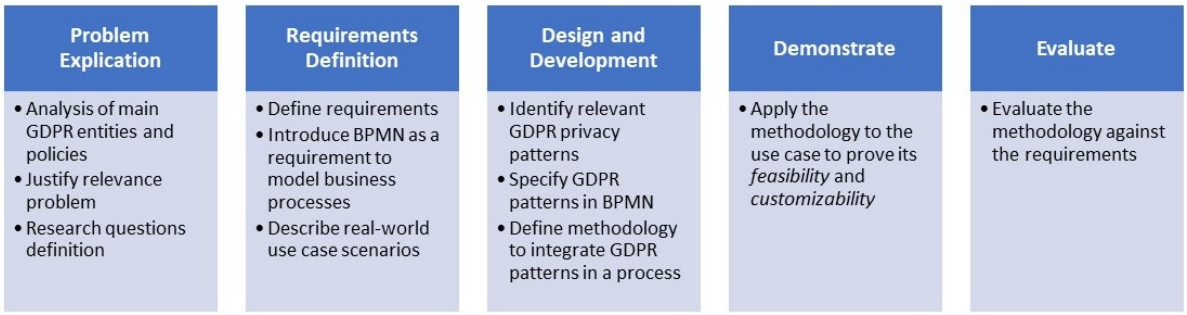


Figure 1: Research approach based on Design Science principles by [11]

in BPMN the activity sequences relevant to the BP, (ii) identifying its vulnerabilities, and (iii) adopting solutions to overcome those. Article 25 of GDPR compels the Data Controller of an organization handling personal data to carry out appropriate technical and organizational measures implementing data-protection principles, both at processing and at design time. When privacy issues are tackled at design time, part of the liability attributed to the Data Controller is shared, at least internally, with the BP designer responsible for designing BP models. It is worth noting that the present paper will not deal with techniques for detecting privacy issues but will provide a methodology to overcome such issues once they are detected.

We structure the paper[1] accordingly to the activities suggested by [11] for delivering a design science artifact (Section 2). After explicating the problem related to GDPR in more depth and administering a dedicated survey to 33 GDPR experts we define two research challenges to be tackled for achieving GDPR compliance at design time (Section 3). Then, we define the requirements for modeling our solution, thus including an introduction to the modeling language BPMN and a description of a motivating use case related to a phone company (Section 4). An additional use case of a hiring company is reported in Appendix A. In Section 5, we design and develop our methodology, which relies on a set of design patterns to integrate privacy-enhancing features in a BPMN model according to GDPR. We demonstrate its *feasibility* using the use cases (Section 6), while in Section 7 we evaluate the methodology with respect to usage (*comprehensibility*, *customizability*, and *learnability*) and structural (*modularity* and *conciseness*) qualities. Finally, in Section 8 we compare our methodology to state-of-the-art, and in Section 9 we draw conclusions and trace future work.

2. Research Approach

As mentioned in the introduction, our research approach is inspired by the Design Science principles described by [11], applied in five sequential phases as shown in Fig. 1: problem explication, requirements, and use cases definition, design and development, demonstration and evaluation. We briefly describe how we enacted such a research approach in the following.

Problem Explication. This phase, which is addressed in Section 3 has a twofold objective. First, *(i)* we outline the problem being tackled by analyzing the main GDPR features (i.e., the entities involved, the definition of personal data, and the obligations of the Data Controller, presented as a list of privacy constraints to be respected); and then, *(ii)* we conducted a survey with 33 GDPR experts with the purpose to derive the research challenges to be tackled for enabling the achievement of the main privacy GDPR constraints in BPs at design-time. The relevance of the problem is also justified by the amount of recent literature works dealing with this matter (cf. Section 8).

Requirements and Use Cases Definition. The second phase, which is discussed in Section 4, consists of defining the requirements to be addressed through our research solution, including *comprehensibility*, *customizability*, *learnability*, *modularity*, and *conciseness*. In addition, we provide an overview of the main BPMN concepts and modeling constructs, whose understanding is required to realize design patterns that capture GDPR constraints. Finally, we present two use cases handling personal data (the second use case is reported in Appendix A), which will be used both as a running example and to demonstrate the *feasibility* of the patterns.

Design and Development. The third phase concerns the creation of the design science artifact. To meet the requirements previously described, in Section 5 we show how we modeled our solution using BPMN, and motivate our design choices. The result consists of a list of nine privacy patterns for BPMN, which represent effective design-time solutions to tackle GDPR constraints in BP models. In addition, we provide a guide on when each pattern should be included in the BP model.

[1]This paper extends our previous work (blinded for peer review, according to the journal's policies, see https://www.elsevier.com/wps/find/journaldescription.cws_home/505553?generatepdf=true) in several directions, as explained in the cover letter.

Demonstration. The fourth phase, discussed in Section 6, requires a demonstration of the artifact in the context of full transparency of privacy constraints into BPs. This is achieved by applying the artifact, i.e., the privacy patterns, to the chosen use cases (related to BP models of a phone company and a hiring company), to show its feasibility when it is put into practice.

Evaluation. In the fifth phase, presented in Section 7, we observe and evaluate how effectively the artifact solves the explicated problem and fulfills the identified requirements.

3. Problem Explication

Here follows the theoretical background of the main GDPR features (cf. Section 3.1) which will aid in conceptualizing the design and development of our solution artifact, and the in-depth analysis of the survey (cf. Section 3.2) conducted with the purpose to derive research questions that will guide the paper.

3.1. Background on GDPR

The GDPR is applicable to all enterprises operating within the European Economic Area (EEA) as well as those outside the EEA that process the personal data of individuals within the EU, regardless of the location of the enterprise or the citizenship of the Data Subjects involved. Thus, in the Eurozone, compliance with GDPR is required whenever a BP deals with personal data. GDPR has introduced changes to privacy and data protection regulation, thus having significant consequences for those who need to design BPs. GDPR requires *privacy-by-design*, which means that data protection is not an addition to the BP, but rather an integral part of it, and the BP should comply with GDPR from the design stage. Therefore, already at this stage, a BP designer needs to take into consideration privacy and data protection issues. With the increase of systems able to collect data automatically, privacy has been at the center of many discussions between designers who want to use such data to provide services to users, while at the same time sharing minimal information while accessing those services.

Entities. To identify who is responsible for what in a BP where data is handled, GDPR defines 4 entities:

- *Data Subject*: is the person the data is about.
- *Data Controller*: s/he collects and stores data from the Data Subject and that determines the purposes of processing such data.
- *Data Processor*: s/he processes data from the Data Subject on behalf of the Data Controller.
- *Data Protection Officer (DPO)*: s/he performs monitoring on the Data Controller and Data Processor to ensure they comply with the GDPR constraints on the data collected from the Data Subject.

Personal Data. In the context of GDPR, *Personal data* is defined as any information related to a person (Data Subject), e.g., the identification number, location data, online identifiers including IP address and cookies, etc. GDPR distinguishes three types of personal data,[2] each with a different level of protection:

- *Personal Data*: any information that can identify a person.
- *Sensible Data*: is a special type of *Personal Data* that requires a higher level of security, i.e., health, genetic, physical, physiological, mental, economic, cultural, social identity, and biometric data.
- *Criminal Records*: is a subset of *Sensible Data* including information to identify past crimes committed by the Data Subject.

Obligations of the Data Controller. This paper focuses on designing modeling patterns in BPMN to explicitly specify the obligations of the Data Controller. This implies a list of constraints that must be fulfilled by the Data Controller to be compliant with GDPR. These obligations are:

- *Data Breach*: in case of a data breach, the Data Controller has to communicate it within 72 hours to the National Authority as well as to the Data Subject. This constraint is not subject to any *de minimis* standard, thus any data breach, regardless of its magnitude, needs to be always communicated along with the actions that will be performed to limit the damage. The only exception is the case in which the stolen data is not usable (e.g., encrypted). However, also in this case, the National Authority can force the Data Controller to communicate the breach to the Data Subject.

[2]The only exception is National Security Data that does not follow GDPR, but is left to the jurisdiction of each State.

- *Consent to Use the Data*: when retrieving personal data, the Data Controller needs to ask the Data Subject for consent and to provide the Data Subject with information about the intentions on how to use and/or process the data.
- *Right to Access and Rectify*: at any moment, the Data Subject can *access* and *rectify* the personal data associated to them. As a result, the Data Controller has the obligation to satisfy these requests.
- *Right of Portability*: at any moment, the Data Subject can ask for the portability of the data associated with her to third parties and the Data Controller has the obligation to satisfy this request.
- *Right to Object*: at any moment, the Data Subject has the right to object to certain types of data processing, such as direct marketing. The Data Controller shall no longer process the personal data unless she demonstrates compelling legitimate grounds for the processing which override the interests and rights of the Data Subject.
- *Right to Object to Automated Processing*: at any moment, the Data Subject has the right to object to a decision based solely on automated processing, and that may significantly affect the Data Subject's freedoms, such as profiling. The Data Controller should implement suitable measures to safeguard the data subject's rights and, if needed, stop the automated processing of personal data.
- *Right to Restrict Processing*: It gives a Data Subject the right to limit the way an organization uses her personal data, rather than requesting erasure.
- *Right to be Forgotten*: if the Data Subject wants their data to be deleted, the Data Controller has the obligation to satisfy this request.

3.2. Survey on GDPR

The survey was administered to 33 GDPR experts with the purpose to investigate whether previous knowledge of the privacy constraints imposed by the GDPR legislation and their explicit representation in a BP model (design-time) can improve their management and resolution when they occur during the execution of the process itself (run-time). GDPR experts are selected from academic and business contexts working on Process Mining, Data Science, and BPM in general. The survey used a Likert scale, ranging from 1 to 4, to evaluate responses to Q1, Q2, Q6, Q7, Q8, and Q9, as shown in Table 1. Additionally, pre-defined answers related to privacy violations were used to evaluate responses to Q3 (cf. Table 2), Q4 (cf. Table 3), and Q5 (cf. Table 4). The complete list of questions is reported below:

- **Q1**: How important do you think it is to comply with the privacy restrictions imposed by the GDPR when executing a BP?
- **Q2**: How often do you witness privacy breaches during a BP execution?
- **Q3**: Which privacy breaches are most likely to be observed when executing a BP?
- **Q4**: In your experience, which are the most complex rights of the Data Subjects (according to the GDPR) to be guaranteed when executing a BP?
- **Q5**: What is the strategy adopted by your company to address GDPR violations?
- **Q6**: Do you think that managing GDPR violations during the design phase of the process (design-time) rather than during its execution (run-time) can reduce potential negative impacts on the process itself?
- **Q7**: Do you think the availability of well-defined procedures (to be integrated into the BP model) for identifying and handling GDPR violations assists the company in resolving them?
- **Q8**: Individual users may contact a company to exercise their rights under the GDPR (rights of access, rectification, erasure, restriction, objection, etc.). Do you think the availability of well-defined procedures for managing GDPR violations may positively affect user satisfaction (e.g., response time, etc.)?
- **Q9**: Do you think that the availability of a well-defined methodology that identifies GDPR violations at design time and integrates their management within BP can make their resolution more efficient during process execution?

Questions seek to understand the awareness of GDPR experts regarding privacy issues at the design stage and their perception of GDPR's importance. The answers provide valuable insights into the effectiveness of design-time privacy considerations and their impact on run-time BPs. Starting from the insights two distinct research questions are formulated:

Table 1: Answers related to Q1, Q2, Q6, Q7, Q8 and Q9

Likert scale	1: Not at all	2: Slightly	3: Moderately	4: Very
Q1	0%	0%	18,2%	81,8%
Q6	0%	9,1%	24,2%	66,7%
Q7	3%	6,1%	24,2%	66,7%
Q8	0%	9,1%	27,2%	63,6%
Q9	0%	9,1%	36,4%	54,5%
Likert scale	1: Never	2: Rarely	3: Occasionally	4: Often
Q2	12,1%	24,2%	54,5%	9,1%

Table 2: Answers related to Q3

Q3: Which privacy breaches are most likely to be observed when executing a BP?	
Access by an unauthorized third party	69,7%
Unauthorized disclosure of personal data	66,7%
Loss of availability of personal data due to accidental or/and deliberate causes	45,5%
Destruction or damage of personal data	36,3%
Computing devices containing personal data being lost or stolen	30,3%

Table 3: Answers related to Q4

Q4: In your experience, which are the most complex rights of the Data Subjects (according to the GDPR) to be guaranteed when executing a BP?	
Right to data portability, Right to be forgotten	42,4%
Right to restrict processing, They are all complex to manage	24,2%
Right to object, Rights to object to automated processing	21,2%
Right to rectify, Right of access, Right to be informed	9%
They are all simple to manage	6%

Table 4: Answers related to Q5

Q5: What is the strategy adopted by your company to address GDPR violations?	
Internal procedures enacted by the company	84,8%
Third-party support	9,1%
I don't know	6,1%

RQ1: How can BP models be developed to detect *privacy violations*, and what are the various types of violations that can be encountered?

RQ2: To what extent does the utilization of an approach that integrates various patterns of *privacy violations* into a BP model prove to be effective?

After defining the requirements for modeling our solution (cf. Section 4), in Section 5 we address RQ1 towards a new methodology that shows how to capture the main privacy GDPR constraints in BPMN models, thus achieving GDPR awareness at the design time. RQ2, on the other hand, is addressed both in Section 6 and Section 7 where the methodology has been first demonstrated to show its feasibility and secondly evaluated to fulfill the identified requirements. By answering these research questions, the paper spells out how organizations can effectively manage privacy violations and comply with GDPR regulations.

4. Requirements and Use Case Definition

In accordance with Art. 25 of GDPR, our methodology should be focused on the *privacy by design* principle, thus forcing organizations to address privacy issues at design time. In addition to this, our artifact must adhere to certain *usage qualities* that define how it should work and be perceived in various application scenarios. We also consider *structural qualities* concerning its overall structure [11].

Regarding the usage qualities, we evaluate several factors. Specifically, we assess *comprehensibility* (the ease with which an artifact can be understood by a user, also called understandability), *customizability* (the degree to

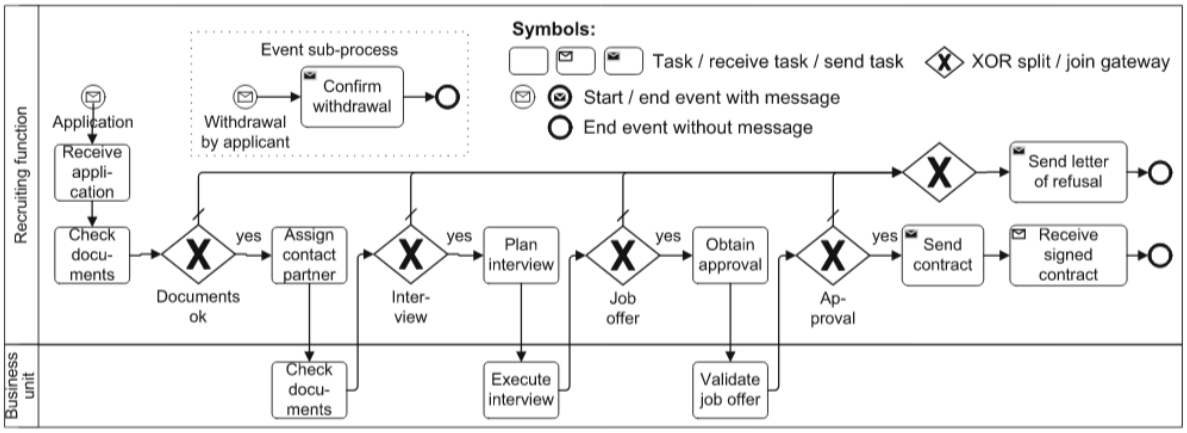


Figure 2: A BP Diagram by [13] specified in BPMN

which an artifact can be adapted to the specific needs of a local practice or user), and *learnability* (the ease with which a user can learn to use an artifact).

When it comes to the structural qualities, we focus on two main aspects. Firstly, we examine *modularity* (the degree to which an artifact is divided into components that can be separated and recombined as needed) and secondly the *conciseness* (the absence of redundant or unnecessary components within the artifact).

Both usage and structural qualities will be discussed in Section 7 to assess the degree to which the proposed methodology fulfills these qualities. Next, we discuss BPMN as a requirement for modeling BPs and present an actual scenario involving the management of personal data.

4.1. Business Process Modeling and Notation (BPMN)

BPMN (Business Processing Modeling Notation) is a standard language, proposed by the Object Management Group (OMG), to design BPs.[3] It equips with an intuitive semantics for BP models. BPMN defines a BP model that includes a set of graphical constructs divided into: (i) flow objects, (ii) data, (iii) connecting objects, and (iv) swimlanes. Flow objects define the behavior of a BP, as the one reported in Fig. 2. They can be classified into *events*, *activities*, and *gateways*. *Events* model the occurrence of states in the real world that are relevant for BPs and, more generally, anything that can happen instantaneously (e.g., an invoice has been received). Events in BPMN can be partitioned into three types, based on their position in the BP: *start events*, which are depicted as circles with a thin border and have, by definition, no incoming sequence flow edges, are used to trigger BPs and, from the simulation perspective, create tokens; *intermediate events*, which are represented as circles with a double border and have both an incoming and an outgoing sequence flow edge, can delay BPs or be triggered during BP executions; *end events*, which are modeled as circles with a thick border and have, by definition, no outgoing sequence flow edges, indicate the termination of BPs and the destruction of the tokens. *Activities* represent units of work performed during BPs that, differently from the events, have a certain duration (e.g., pay an invoice). Activities capturing units of work that are not further refined are called atomic activities or *tasks* (e.g., check stock availability). Activities might also have an internal structure, in which case they are called sub-processes. *Sub-processes* are composite activities that can be broken down into smaller units of work, generally sub-processes gather together groups of activities that achieve a particular goal, e.g., when a company has different suppliers, the activity of acquiring raw materials can be modeled as a sub-process, managing the raw material order from each supplier with a different activity. An *event sub-process*, a sub-process delimited by a dotted rectangle with rounded corners, is started by the event attached to an activity's boundary and encloses the procedure that would be triggered by the boundary event. Those sub-processes are used to handle both the reception of external messages as well as handling errors depicted with a lightning bolt: in the latter scenario, the BP is not aborted and the error is caught by the event sub-process providing the recovering activity guaranteeing a consistent state of execution. *Gateways* are used to represent the split and join behavior of the control flow when there is a need to model specific conditions like mutual exclusion or concurrence. A gateway determines the forking, merging, or joining of paths. The main gateway types are *exclusive*, *parallel*, *inclusive*, and *event-based*. In BPMN, each gateway acts as a join node or as a split node. *Split gateways* have exactly one incoming edge and at least two outgoing edges, representing the flow that is branched. By contrast, *join gateways* have at least two incoming edges and one outgoing edge, modeling the merging of the control flow.

[3]We refer here to the last release of BPMN, namely BPMN v2.0 – http://www.omg.org/spec/BPMN/2.0/

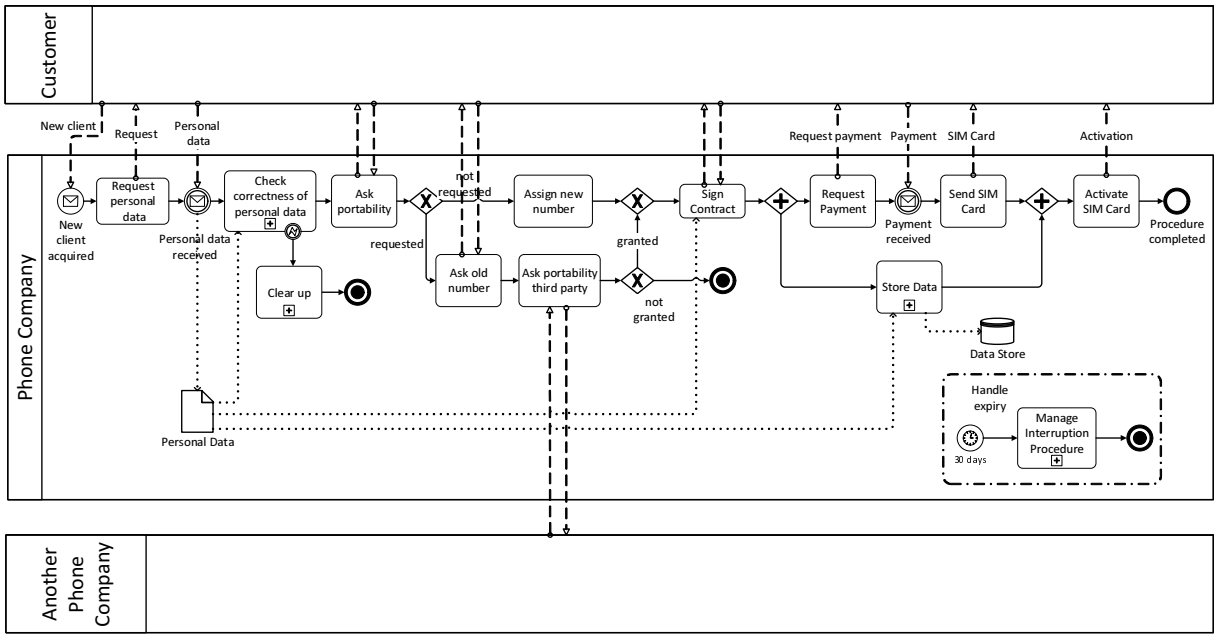


Figure 3: BPMN model for the case of the phone company

Exclusive gateways, also known as *XOR gateways*, model the relation between two or more alternative activities that are mutually exclusive, i.e., only one of them can be executed according to a condition. *Parallel gateways*, also known as *AND gateways*, model the relation between two or more activities that can be executed concurrently, i.e., they do not have any order dependencies on each other, but they do have to be executed. Specifically, the AND-split models the parallel execution of two or more branches, and the AND-join awaits all the tokens that have been created and merge them into one. *Inclusive gateways*, also known as *OR gateways*, model the relation between two or more alternative activities that are not mutually exclusive and each of them can be executed according to a condition. Unlike exclusive gateways, the true evaluation of one condition does not exclude the true evaluation of other conditions. *Event-driven gateways* model the *race* of two or more events against each other, i.e., the first event to occur determines the continuation of the BP. The event-driven gateway acts like a split gateway, having as many outgoing edges as the number of events participating in the race. When a token arrives at this gateway, the execution of the instance stops until one of the events on the outgoing branches occurs.

BPMN allows explicitly model data using constructs denoting *artifacts*. *Artifacts* are used to show additional information that is not directly relevant to the sequence flow or the message flow of the BP. Artifacts serve only information purposes so that the execution semantics of a BP is not influenced by them. The main supported artifacts are data objects and data stores. *Data objects* provide information regarding activity requirements and specify data inputs and outputs of activities. *Data stores* are places containing data objects that need to persist beyond the duration of a BP instance, e.g., a database for electronic objects or a filling cabinet for physical ones. *Connecting objects* connect flow objects, lanes, or artifacts. The *Sequence flow* is used to specify the ordering of flow objects, while *message flow* describes the flow of messages between business partners represented by pools. *Association* is a specific type of connecting object that is used to link artifacts to elements in BP diagrams.

Swimlanes are used to specify who is responsible for the execution of a certain BP. When multiple actors need to be represented within the same swimlane, other modeling constructs, called *pools* and *lanes*, can be used. *Pools* are generally used to model resource classes, i.e., independent organizational entities that do not share any common system, but communicate with each other through messages (e.g., a business party playing the role of customer, supplier, or manufacturer). *Lanes* represent organizational entities such as departments or single resources in organizations. Lanes can be nested in multiple levels and sub-lanes can be used to define for instance organizational entities within departments.

4.2. Use Case

We now describe a real use case handling personal data before the enactment and implementation of the required BPMN patterns for GDPR compliance[4].

[4]In addition to the Phone Company use case in Appendix A we present an additional example related to a Hiring Company that deals with personal data and demonstrates how the required BPMN patterns can be implemented during the design phase to achieve GDPR compliance.

We take as an example a *phone company* in the process of acquiring a new customer. The phone company requests the new client's data (e.g., name, surname, address, etc.). Once the client has provided this data, the phone company goes through a verification process to determine if the data given by the new customer is correct. If not, a clear-up procedure starts and the BP ends. Otherwise, the future customer is asked if they want to *port* their old phone number into the new phone plan to be subscribed. If the answer is positive, then the phone company asks the new client for the old number and they engage with the *previous phone company* for carrying out the portability procedure. If either the procedure cannot be completed or the answer is negative, the BP is interrupted. Otherwise, the customer signs the contract describing how the phone company will provide the service: no detail on how the phone company will use the client's data is given. Then, the phone company both stores the data of the new client and requests payment from the client: once the payment is received, it sends the SIM card to the client. When these activities are completed, the company activates the SIM card and successfully concludes the procedure. If the procedure takes for some reason more than 30 days to complete, then the BP is interrupted. The BPMN model representing the scenario described above is shown in Fig. 3. It is worth noting that the procedure does not yet take into account the potential risk to get a data breach and does not provide mechanisms to protect the customer's privacy.

5. Design and Development

The aforementioned Article 25 of GDPR compels the Data Controller of a company handling personal data to carry out appropriate technical and organizational measures to implement data-protection principles, both at the time of processing and at design time. If a privacy issue occurs during the execution of the BP, the Data Controller can take a countermeasure that has only a local effect, indeed an adjustment introduced at run-time impacts only on the single execution that gave rise to the problem. On the contrary, when a modification is made at design time, it affects all the future executions of the BP. In other words, a problem solved when it appears is a problem that can reoccur, while a problem solved at the modeling stage is a problem handled over time. At design-time BPs are represented at a high level of specification through BP models. BP models describe the behavior of BPs, their activities, roles, and conditions, irrespective of their execution. To be compliant with GDPR at design time, it is fundamental to arrange BP models that take into account the obligations of the Data Controller. We now propose our methodology to tackle RQ1 and fulfilling the requirements previously described. We also produce descriptive knowledge by modeling our solution using BPMN, while motivating the design choices in words.

5.1. Pattern Development

We introduce a list of nine privacy patterns for BPMN, which represent effective design-time solutions to tackle GDPR constraints in BP models. We developed such patterns in a way that no additional BPMN symbol is required to integrate them into a non-GDPR compliant BP model: the patterns could be either triggered by some events or should be inserted as a sub-process before the acquisition of sensible information from the user. In the first case, triggers might occur once a message is received from the Data Subject, or when an exception (e.g., data breach) happens within the BP.

Consent to Use the Data. Before retrieving any kind of personal data from the Data Subject, the Data Controller has to ask the Data Subject for consent. In particular, the Data Controller needs to collect the following list of aspects the Data Subject should be aware of before giving her data to the Data Controller.

- in case the data has not been directly obtained from the Data Subject, from which source the personal data originates;
- the existence of the right to lodge a complaint to a supervisory authority;
- the existence of the right to withdraw the consent at any time;
- the existence of the right to data portability;
- the existence of the right to delete personal data;
- the existence of the right to access personal data;
- the existence of the right to rectify personal data;
- the period for which the personal data will be stored, or if this is not possible, the criteria used to determine this period;
- the existence of any profiling and meaningful information about the envisaged consequences of processing the personal data;

- if the personal data can be transferred internationally;
- who are the recipients or categories of recipients of the personal data;
- which are the interests pursued by the Data Controller or by third parties;
- the legal basis of the processing;
- the purposes for which the personal data will be processed;
- the identity and contact details of the Data Controller and the DPO.

Then, consent to use the data is requested from the Data Subject. If the consent is given, the data is collected. The design pattern in Fig. 4 implements the privacy constraint *Consent to Use the Data*. In the example of the phone company, this pattern can be added as a sub-process just before asking for the actual data to the potential new customer, at the start of the BP. This guarantees that the company is transparent with the customer and asks for explicit consent for any possible data usage.

Right to Access. When the Data Subject sends a request availing the right to access, the Data Controller has to *(i)* retrieve all the data associated with the Data Subject, and *(ii)* retrieve any processing on the data that has been made. Then, they are both sent to the Data Subject. The design pattern in Fig. 5 implements the privacy constraint *Right to Access*. In the example of the phone company, this pattern can be implemented as an asynchronous request from the Data Subject that can be received at any point in time after that any personal data has been retained. Note that the customer can request to access her personal data even before the BP is completed (potentially even before the customer signs the contract), and the phone company has to handle this request by providing any personal data it possesses.

Right of Portability. When the Data Subject sends a request availing the right of portability, she needs to specify the third party at hand. The third-party contacts the Data Controller which has to *(i)* retrieve all the data associated with the Data Subject, and *(ii)* retrieve any processing on the data that has been made. Then, they are both sent to a third party. Finally, the third party communicates to the Data Subject that the portability happened successfully. The design pattern in Fig. 6 implements the privacy constraint *Right of Portability*. In the example of the phone company, the company needs to have a procedure to handle portability when requested by a third-party company. However, in the process of acquiring a new client, even though the user requests the portability, *Another Phone Company* (and not the phone company of the case study) should be able to implement this pattern.

Right to Rectify. When the Data Subject sends a request availing the right to rectify, the Data Controller has to rectify the data as requested by the Data Subject and communicate back to the Data Subject that her data has been rectified. The design pattern in Fig. 7 implements the privacy constraint *Right to Rectify*. In the example of the phone company, the customer should be able to rectify the data at any time. For instance, if before signing the contract the customer changes address, or simply notices incorrect information, she should be able to rectify such information.

Right to Object. When the Data Subject sends a request availing the right to object, the Data Controller has to check if the data for which consent has been withdrawn is relevant for the execution of the BP. For this purpose, the BPMN model should start with a XOR-split preceded by a decision activity evaluating the condition according to which one of the alternative paths is chosen. Specifically, if the withdrawn data is relevant, the Data Subject should be informed that their objection to consent will mean the abortion of all running BPs. If, again, the user expresses their intention to continue the objection procedure, then all the BPs using the withdrawn data should be stopped; otherwise, the *Right to Object* sub-process is aborted. Whether withdrawn data are relevant or not, if the Data Controller receives a request for objection, it has to stop using the data associated with the Data Subject, and communicate back to the Data Subject that their data is not used anymore. Thus the BP terminates. The design pattern in Fig. 8 implements the privacy constraint *Right to Object*. In the example of the phone company, this asynchronous request from the client can happen at any time, thus the phone company might implement this pattern in BPMN as an event sub-process. If, at any time during the procedure of acquiring a new customer, the customer withdraws the consent to use the data, the phone company has to fulfill such request and implement a *Right to Object* procedure. Furthermore, if the data, for whose use consent has been withdrawn, is relevant, all BPs that need the concerned data to be executed, including the process of acquiring the new customer, must be stopped.

Right to Object to Automated Processing. When the Data Subject sends a request availing the right to object to automated processing on their personal data, the Data Controller should at least guarantee human intervention in all the concerned automated decision activities. In BPMN this can be done through compensation handlers. Any automated decision activity should dispose of a catching compensation event that enables an exception flow leading to a compensation activity. The compensation activity should perform the same task as the automated

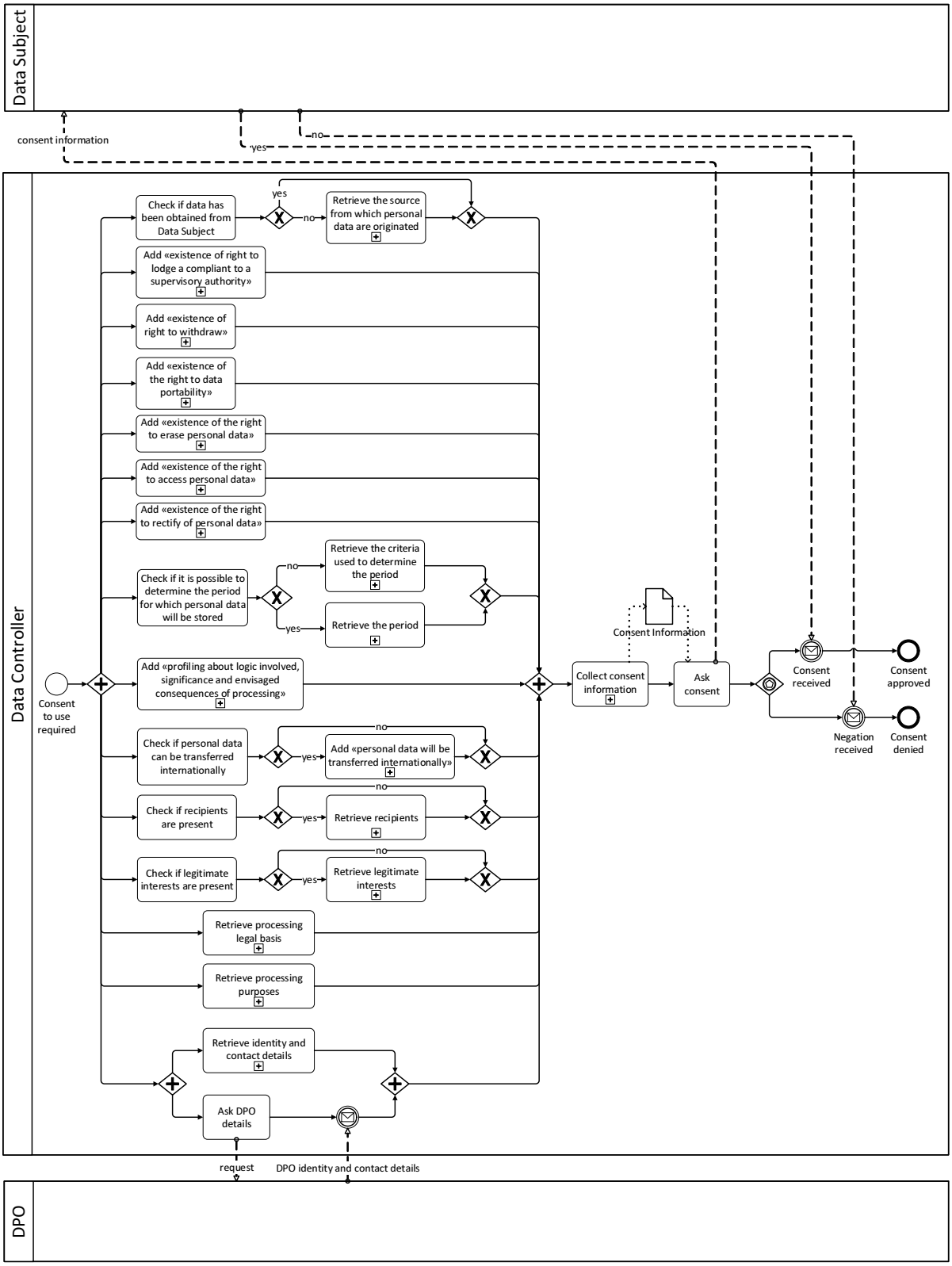


Figure 4: BPMN model for pattern *Consent to Use the Data*

activity, but it should be executed by a human. In a BPMN model manual activities are identified by a specific marker, thus the compensation activity should be marked with a manual marker.

Differently from the others, this right requires that any automated decision activity is easy to spot within the BP and that it disposes of a compensation event that catches the signal thrown by the compensation throwing event in the pattern. In other words, the pattern presented is not sufficient for the fulfillment of the right, but, in this case, the entire BP should be modified to be compliant with GDPR. The design pattern in Fig. 9 implements the

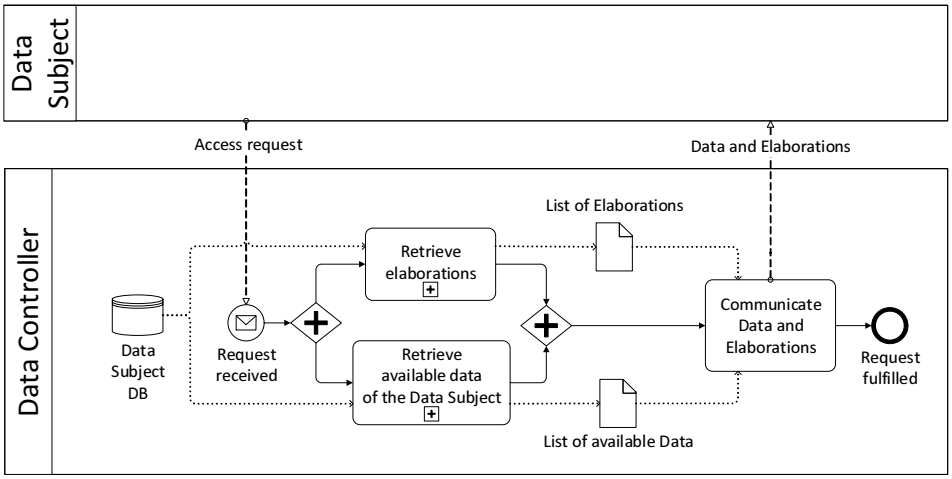


Figure 5: BPMN model for pattern *Right to Access*

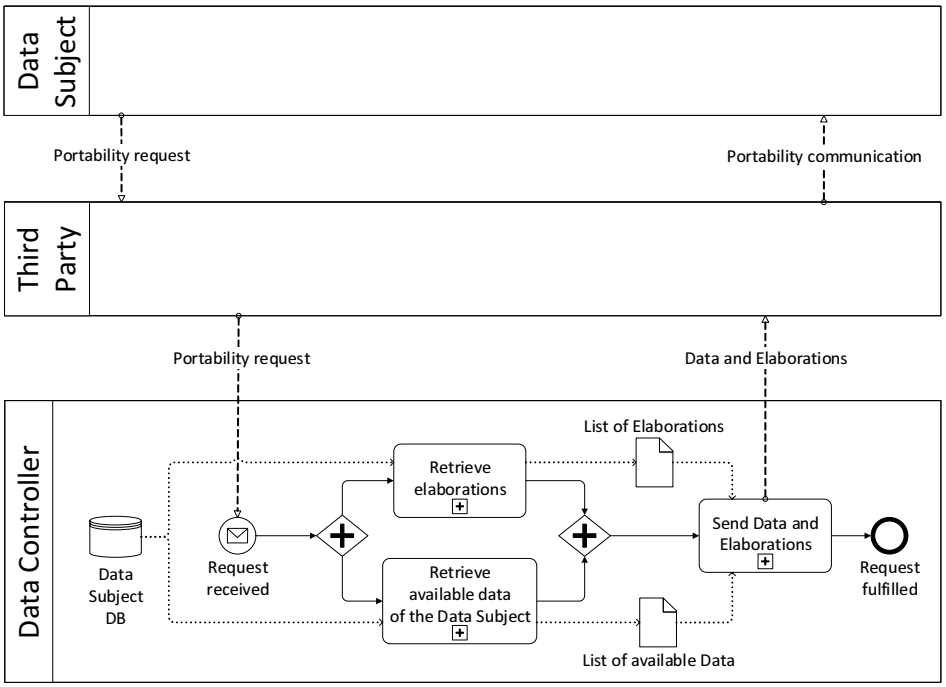


Figure 6: BPMN model for pattern *Right of Portability*

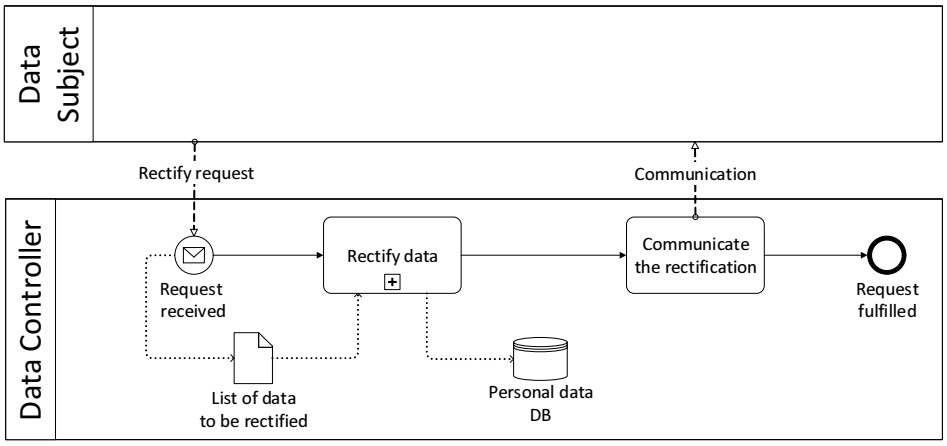


Figure 7: BPMN model for pattern *Right to Rectify*

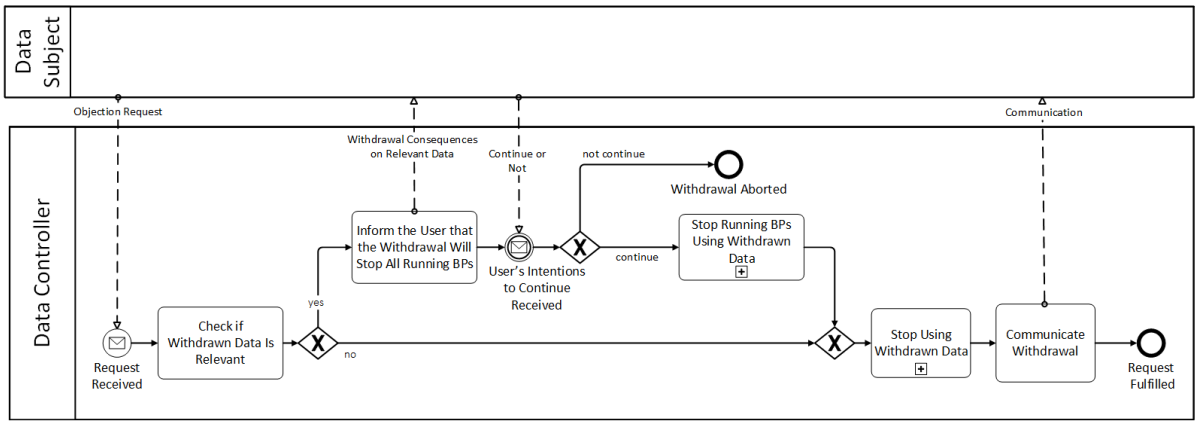


Figure 8: BPMN model for pattern *Right to Object*

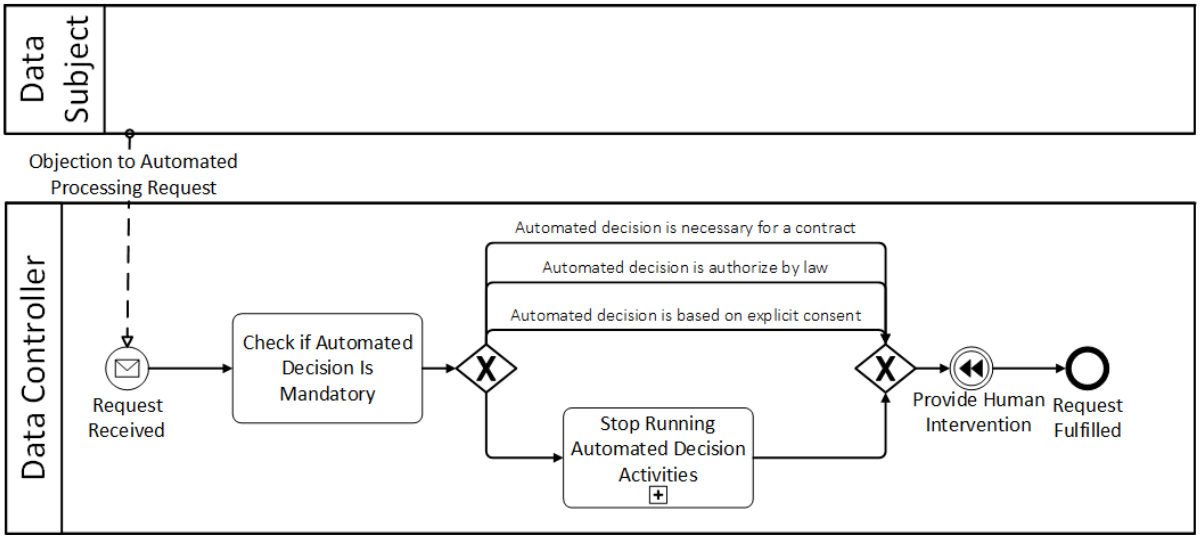


Figure 9: BPMN model for pattern *Right to Object to Automated Processing*

privacy constraint *Right to Object to Automated Processing*. In the case of the phone company, the decisions taken in the process of acquiring a new customer are related to the correctness of personal data and the approval of the portability request. While the latter is a decision that does not depend on the company, given that portability has to be granted by the old phone company, the personal data correctness check is, in all likelihood, an automated decision activity. Nevertheless, the correctness of personal data is fundamental for initiating the contract and there is no logical sense for asking the check on address, ID, or fiscal code to be executed manually. As a consequence, there is no need for the phone company to dispose of a *Right to Object to Automated Processing* pattern in this BP.

Differently, the hiring procedure is based on profiling: the activity "Analyse Candidate Profile" evaluates the CV and cover letter, provided by the applicant, and the profiling test results, furnished by the tests provider. Giving the consent to store and use their personal data the user accepts all the automated decisions taken during the hiring process. Notwithstanding, they can request at any moment human intervention to get an explanation about such decisions. A similar request is more likely to occur after a rejection, anyway, the company should be able to manage it whenever it occurs. For this purpose, not only the pattern should be added to the BP as an event sub-process (cf. Section 5.2), but also the "Analyse Candidate Profile" activity, which is in its turn a sub-process composed of automated and human decision activities, should provide a compensation handler for each automated decision activity it includes.

Right to Restrict Processing. When the Data Subject sends a request availing the right to restrict the processing of their personal data, the Data Controller has to *(i)* collect the data concerned by the request, *(ii)* temporarily move the involved data to another processing system that is not accessible by users, *(iii)* erase the data from the database and *(iv)* disseminate notification of data restriction to the Data Subject and to all the recipients of such data. So the request is fulfilled. In the BPMN model, these activities should be put in sequence apart from the notifications, which can be executed in any order. These same notifications could be modeled with message events or with send activities. Specifically, the notification to the Data Subject can be modeled with an intermediate throwing message

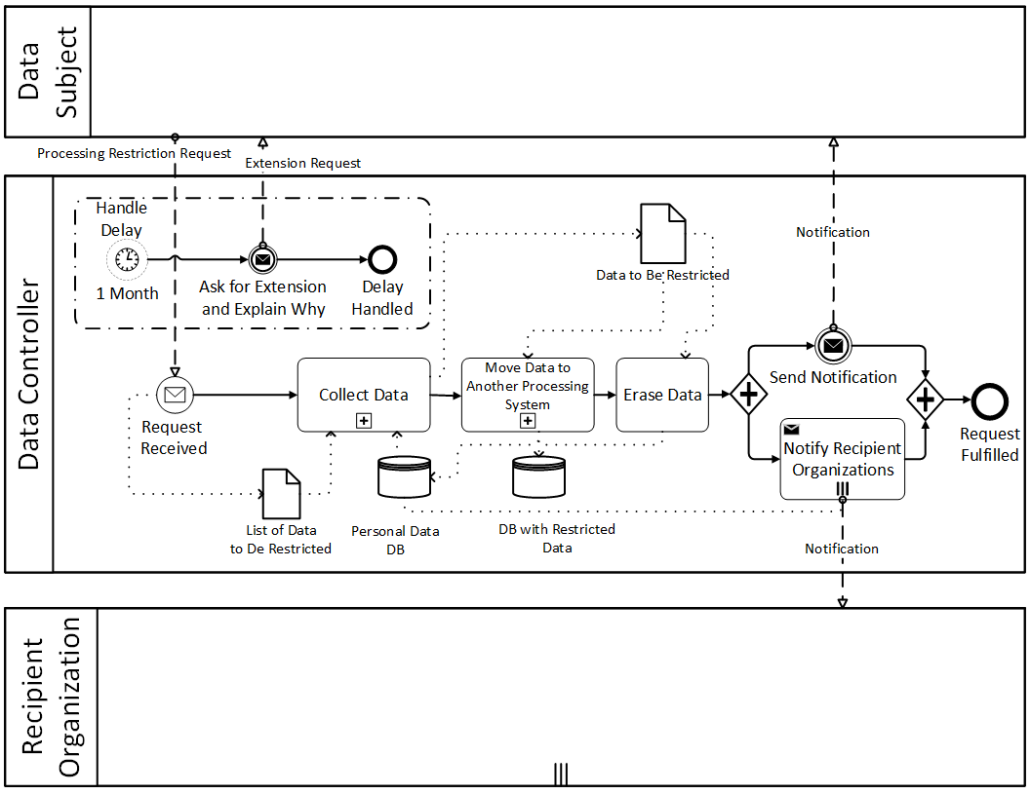


Figure 10: BPMN model for pattern *Right to Restrict Processing*

event, given the simplicity and the immediacy of the notification; instead, due to the multiplicity of recipient organizations and the complexity of the notification, a multi-instance send activity is required to notify the data restriction to all the data recipients. In other words, the Data Subject is the one who sent the restriction request, so they expect a notification such as "Data Restricted"; differently the recipients become aware of the restriction only when they receive a notification, thus such notification should be more detailed. It has to be noted that a restriction does not imply a loss of the data, but the concerned data is erased from the *Personal Data DB* only after it has been stored in a private database so that it could be easily recovered afterward. In addition, the procedure should complete before the 1-month timer expires, if it happens the Data Controller can ask for additional time to manage the request, but an explanation has to be provided to the Data Subject. An event sub-process starting with a timer event should be provided in the BPMN model. A request for restriction of personal data processing can occur under certain circumstances, however, every company handling personal data should dispose of a pattern managing a restriction request whenever it occurs, coming to the request directly from the Data Subject. In the case of the phone company, the new client can contest the correctness of their data and ask for restriction while verifications take place, or a data breach can have occurred and the company has lost some data. If the customer expresses their willingness to keep their data stored by the company, so that they could easily restart the procedure once data has been rectified or recovered, a restricted processing procedure has to be executed. As well as for the phone company, the hiring company has the same conditions to request personal data restriction, for example, the applicant that is not deemed suitable for the position they applied for decides to keep their data stored in the company's database for future applications, or it can happen that during the procedure the applicant changes their mind and withdraws the consent to treat their data, thereby the company is obliged to restrict processing while the necessary controls to approve the withdrawal are done.

Right to be Forgotten. When the Data Subject sends a request availing the right to be forgotten, the Data Controller has to retrieve the data related to the request and check if this data is relevant. If not, the Data Controller eliminates such data and communicates this to the Data Subject. Otherwise, the Data Controller communicates to the Data Subject why the data is relevant. The design pattern in Fig. 11 implements the privacy constraint *Right to be Forgotten*. In the example of the phone company, this pattern can be implemented during the process of acquiring a new customer even though the request will be for sure rejected (since all the data requested from the clients is necessary at this stage). This is needed to provide the customer with an understanding of why the data is relevant within the BP.

Right to Be Notified of Data Breaches. In case of a Data Breach, the Data Controller has to retrieve the breached data. From this data, the Data Controller needs to extract a list of Data Subjects who had their data breached.

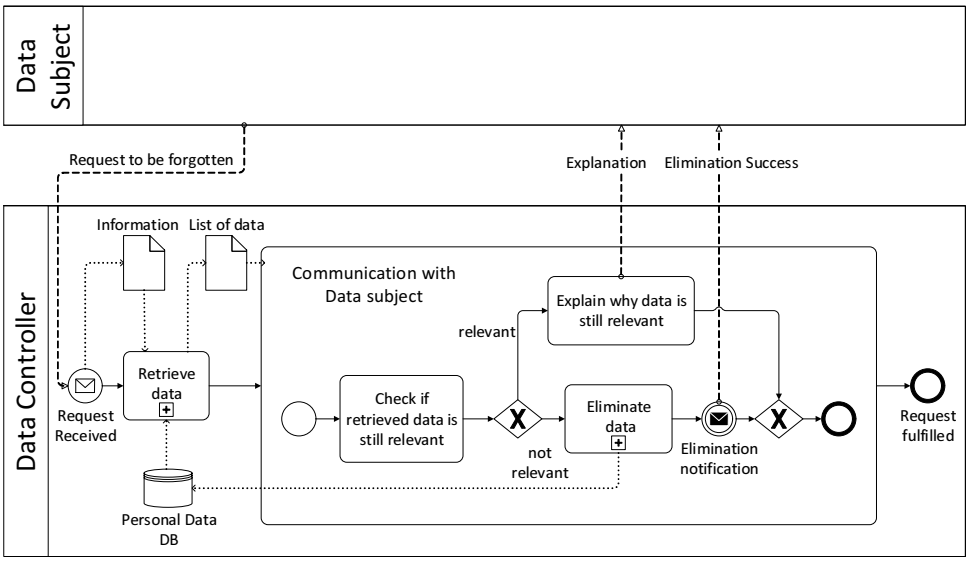


Figure 11: BPMN model for pattern *Right to be Forgotten*

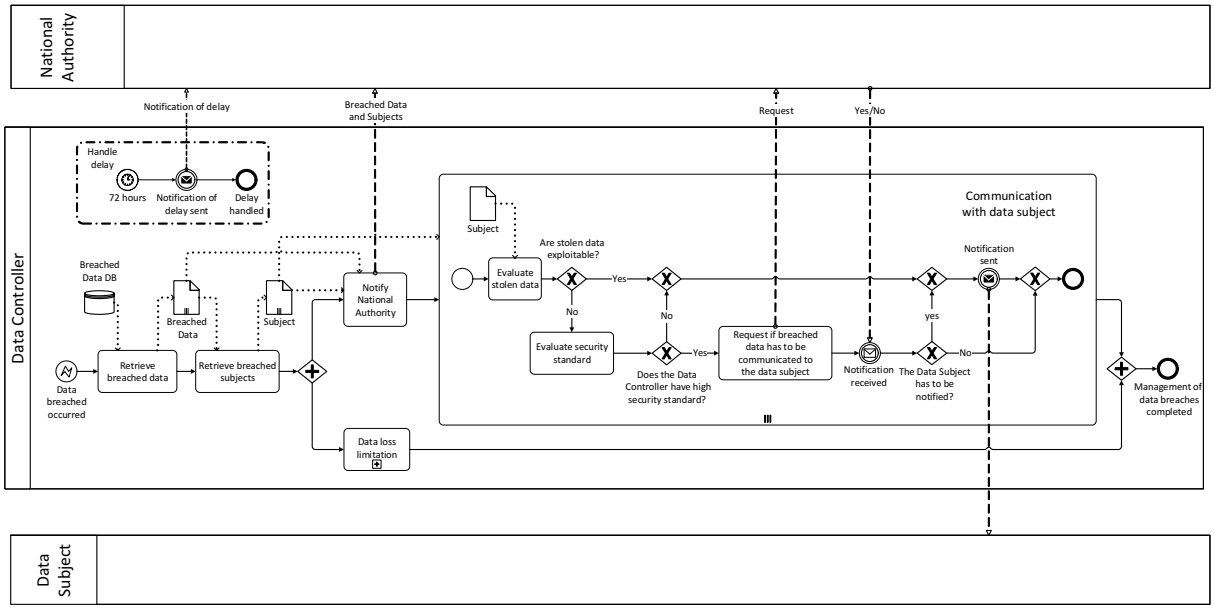


Figure 12: BPMN model for pattern *Data Breach.*

Then, in parallel, the Data Controller needs to limit the data loss and send a notification to the National Authority. For each breached Data Subject, the Data Controller evaluates if the stolen data is usable or not. If not, and if the Data Controller is proven to manage data using high-security standards, this is communicated to the National Authority which decides whether the breach should be communicated to the Data Subject or not. Otherwise, the Data Controller needs to notify the Data Subject directly. The design pattern in Fig. 12 implements the privacy constraint *Data Breach*. It is worthwhile noting that, during any BP involving personal data, a data breach can occur, and the Data Controller must promptly handle the problem within 72 hours. In the example of the phone company, a data breach can happen at any time after the personal data has been acquired. Thus, implementing the *Data Breach* pattern can help the BP to be reactive in case of a data breach, to properly provide a recovery procedure and communicate the data breach to both the Data Subject and the National Authority. Notice that if the 72 hours limit is not respected and the Data Controller is not able to provide a reasonable justification, the penalties amount to 20 million, or 4% of the company's global revenue.

5.2. Extending BPMN Processes with GDPR patterns

As the extension process is BP designer-centric, they should know in the first place **when** each pattern from the previous section should be included in the BP model. To do so, BP designers should detect fallacies in meeting the GDPR requirements and, whenever this happens, they should insert specific patterns. The diagram in Fig. 13

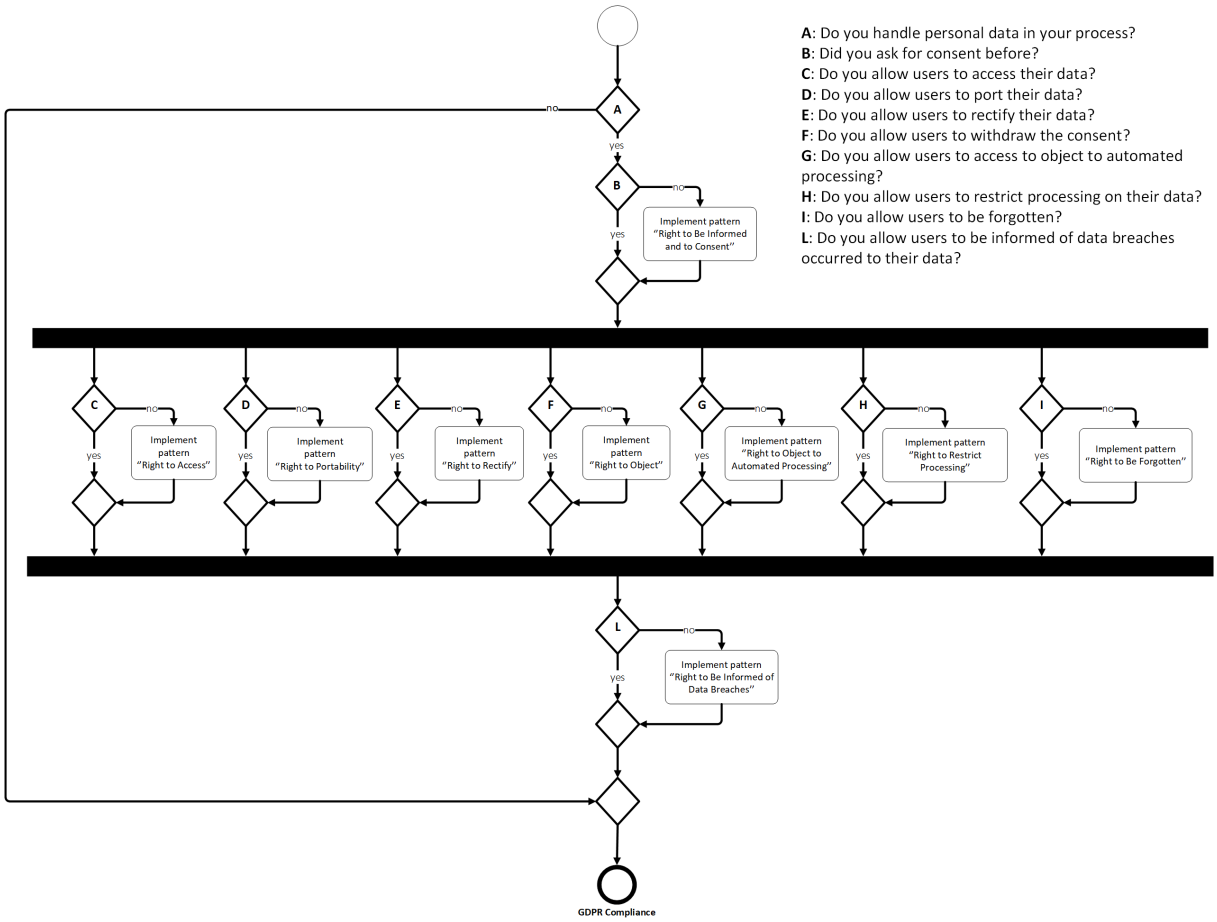


Figure 13: A methodology to decide when a pattern should be introduced

shows the logic the BP designer should follow to decide when a design pattern should be introduced in the BP in order to be compliant with GDPR. As we might observe, the insertion of such patterns is subject to the usage or storage of data coming from the Data Subject. If that happens, the questions addressed in Fig. 13 help to identify which patterns must be enacted. Thus, the BP designer needs to know where to apply these patterns to the existing BPMN model.

Most of the Data Controller's obligations, relative to the Data Subject's rights defined by GDPR, are associated with external requests from the Data Subject. Except for patterns *Right to Be Informed and to Consent* and *Right to Be Notified of Data Breaches*, the other patterns should be executable at any stage of the BP, since a request from the Data Subject can occur at any time. For example, a request for accessing, restricting, or even porting personal data to another company, can arrive even immediately after a contract has been signed or a new account created; consent to use personal data can be withdrawn soon after it has been given; and a request for restriction of data processing can occur whenever the Data Subject contests the accuracy of their personal data.

In all these cases, the Data Controller should be put in the condition to anticipate any possible request from the Data Subject, so that, when the specific request is received by means of a message, the related pattern can be immediately executed. Moreover, the Data Controller has to comply with the *Right to Be Informed and to Consent*, whose pattern must be executed once and for all when some personal data are asked for the first time and with the *Right to Be Notified of Data Breaches*, which is considered as an internal exception thrown by the system when unlawful access or processing is discovered. The exception has to be caught as an error inside any BP handling personal data and should provide a recovery procedure to limit the damage associated with the violation.

The BP designer should include in the BP model the privacy-enhancing patterns **where** they can best fit the Data Controller's obligations. The recommended collocation for each of the privacy patterns introduced is explained hereunder:

- *Right to Be Informed and to Consent*: it is the first mandatory privacy constraint imposed by GDPR. It must be placed before the first activity that requires personal data. The BP designer should detect within the BP model the first activity asking for personal data and put the pattern right before it, as shown in Fig. 14. An activity called "Request Personal Data" might be what the designer is looking for but, on the other hand, it is not sure that in general data are requested with these exact words, on the contrary, it frequently happens

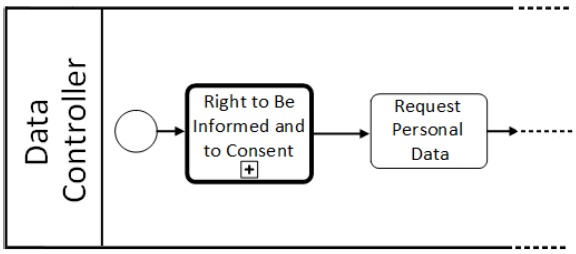
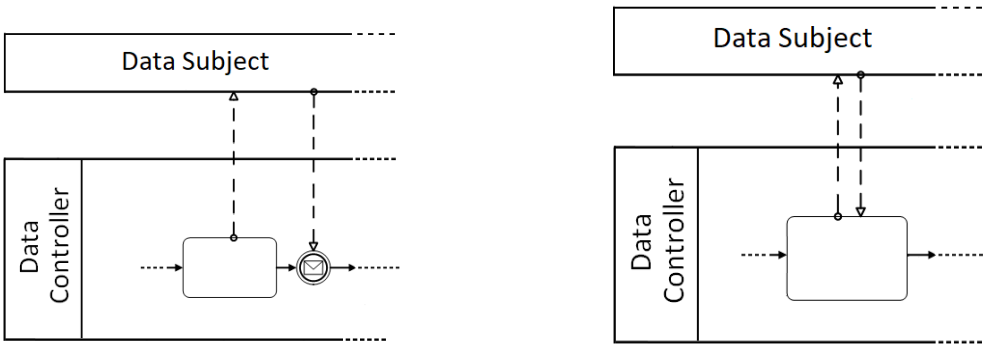


Figure 14: Correct placement for pattern *Right to Be Informed and to Consent*



(a) Activity & catching message event

(b) Send and receive activity

Figure 15: Requesting personal data in BPMN

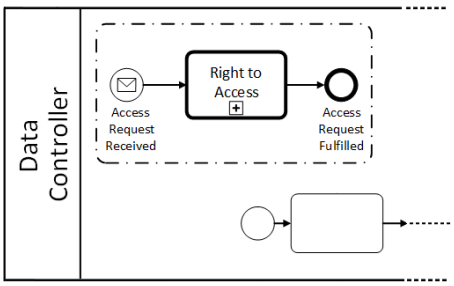
that personal data requests are made using different expressions, like for example "Ask for Personal Data", "Retrieve Personal Data", "Request Personal Information" or "Request Details". As a result, it is extremely difficult for the BP designer to identify, at the re-design stage, which are the activities that are effectively asking for personal data. Therefore, each activity should come with an explicit explanation of its purpose or, even better, a snippet code for its implementation.

Within the BPMN model, activities requesting personal data are likely to send something to the Data Subject and are followed by an intermediate catching message event receiving a message flow from the Data Subject. The outgoing message flow from the activity represents the request for personal data and the message flow coming into the catching event is the personal data requested. Such behavior is shown in Fig. 15a. Nevertheless, the same behavior can be obtained by using an activity with an outgoing and an incoming message flow to and from the Data Subject, as shown in Fig. 15b.

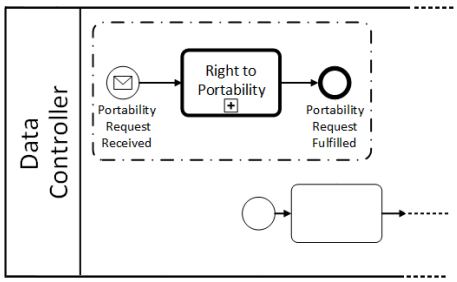
Still, the notation proposed in Fig. 15a and Fig. 15b might not necessarily involve personal data requests from the Data Controller toward the Data Subject. For example, in the case study of the phone company, payment is requested in the "activity + catching message event" form and portability through the "send and receive activity" notation. Thus, it results once again difficult to identify the first activity requesting personal data. In fact, the BP designer should check, for every notation of the proposed types detected in the BP model, whether or not it involves personal data and, if yes, find the first that may occur during the execution of the BP. This leads to an activity-by-activity check, which is proven to be highly time-consuming and hardly automatable. What is sure is that, if the BP handles personal data, there will be at least an activity requesting such data and an object flow from the Data Subject to the Data Controller related to personal data.

- *Right to Access*: it is an external request by the Data Subject, thus it can occur at any moment. The privacy pattern has to be placed in the BP model as an event sub-process[5], as shown in Fig. 16a.
- *Right of Portability*: it is an external request by a third party reflecting a Data Subject's request that can occur at any moment during the execution of the BP. Therefore, the privacy pattern has to be included in the BP model as an event sub-process, as shown in Fig. 16b.
- *Right to Rectify*: it is an external request by the Data Subject and it can occur at any moment. The privacy pattern has to be introduced in the BP model as an event sub-process, as shown in Fig. 16c.

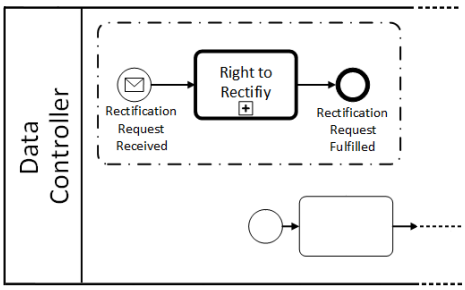
[5]Event sub-processes are used to capture exceptions (and define recovery procedures) that may affect an entire BP.



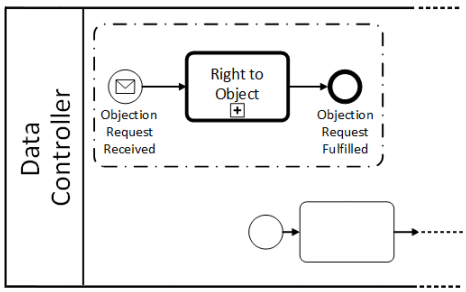
(a) *Right to Access*



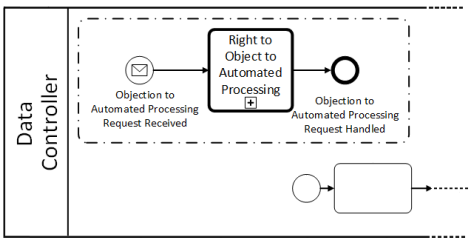
(b) *Right of Portability*



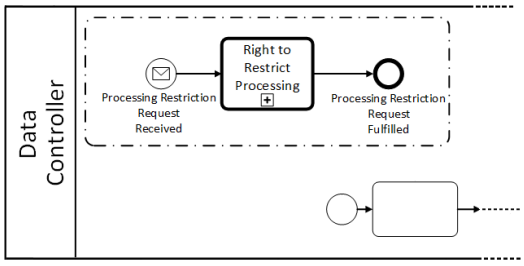
(c) *Right to Rectify*



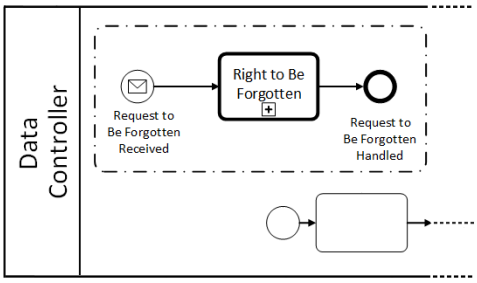
(d) *Right to Object*



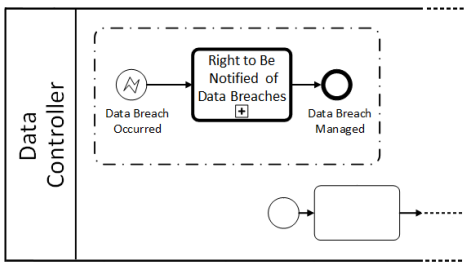
(e) *Right to Object to Automated Processing*



(f) *Right to Restrict Processing*



(g) *Right to Be Forgotten*



(h) *Right to Be Notified of Data Breaches*

Figure 16: Positioning the patterns within the BP model

- *Right to Object*: it is an external request by the Data Subject, thus it can occur at any moment. The privacy pattern has to be positioned in the BP model as an event sub-process, as shown in Fig. 16d.
- *Right to Object to Automated Processing*: it is an external request by the Data Subject involving decision activities. Since a similar request can occur at any moment, the privacy pattern should be included in the BP model as an event sub-process, as shown in Fig. 16e.
- *Right to Restrict Processing*: it is an external request by the Data Subject that is most likely to occur under certain circumstances, but it is not excluded that the restriction request might arrive at any moment. Hence, the privacy pattern has to be considered in the BP model as an event sub-process, as shown in Fig. 16f.
- *Right to Be Forgotten*: it is an external request by the Data Subject, thus it can occur at any moment. The privacy pattern has to be placed in the BP model as an event sub-process, as shown in Fig. 16g.

- *Right to Be Notified of Data Breaches*: it is an internal error that can be thrown at any moment. The privacy pattern has to be placed in the BP model as an event sub-process starting with a catching error event, as shown in Fig. 16h.

If the designer answers all the questions and integrates the patterns when necessary as previously described, then the BP in BPMN is GDPR compliant. The integration of the patterns as suggested enables the Data Controller to both manage every request by the Data Subject and handle data breaches that may occur during the execution of the BP. The incoming section will describe how to proactively apply the aforementioned methodology to real-world use case scenarios.

6. Demonstration

This demonstration section aims to show the *feasibility* of the proposed methodology when it is put into practice through a real use case. That is, starting from the BPMN process that does not conform with the GDPR, we are now able to obtain a GDPR-compliant version of the BP model by applying the methodology. In the following, we show how applying the methodology to the use case enables the transformation of the BP model that is not compliant with GDPR in its compliant counterpart.

The subscription of the contract requires personal data (i.e., name, surname, ID number, address, and fiscal code), the BP has to be modified to conform with GDPR. As observed in Section 5.2, the BP designer should detect which activities require personal data. To ease the task, we suggested in such subsection to model any personal data request in the form "activity + catching message event" and to name the activity requesting personal data with the word “Request”. The previous BPMN model of the Phone Company is conformant to such requirements. As per the aforementioned specifications, the remaining types of requests should be modeled as “send and receive activity”: the “Request Payment” activity name is also changed into “Demand Payment”. After that, the first activity requesting personal data is associated with a data object renamed as PII: it includes the client’s name, surname, ID number, address, and fiscal code. PII is an input to the “Sign Contract” activity, which then is the activity that most of all need personal data. The *Right to Be Informed and to Consent* pattern is collocated right before the activity “Request Personal Data”. As we assume that users will not object to automated processing, the pattern *Right to Object to Automated Processing* is excluded. The remaining privacy design patterns are introduced in a huge event sub-process. The event sub-process has different starting events, one for each privacy pattern required, and at the occurrence of a request by the customer the associated path is enabled and provides the fulfillment of the request. The same happens for the occurrence of a data breach. Fig. 17 shows the GDPR-compliant version of the BPMN model for the procedure of acquiring a new customer executed by a phone company.

Besides the Phone Company use case just described, in Appendix A we illustrate an additional use case involving a Hiring Company that handles personal information and showcases how the essential BPMN patterns can be incorporated during the design stage to ensure adherence with GDPR.

7. Evaluation

In this section, we provide the evaluation of our artifact by addressing RQ2, thus finally assessing the requirements listed in Section 4. According to the Design Science approach outlined by [11], we provided a complete characterization of the analyzed problem and methodology. Specifically, after assessing in the introduction (Section 1) that the BP designer should be overall responsible within an organization for being compliant with GDPR requirements (“*who is responsible for achieving GDPR Compliance?*”), the “Problem Explication” section (Section 3) explained *why achieving GDPR compliance is important* for entities handling sensible user information. Then, Section “Design and Development” (Section 5) discussed *what is necessary to achieve GDPR compliance*, as well as describing *when* and *where our design patterns should be introduced within the BPMN model*. We define below the evaluation *context*, *goals and strategies*, the *design* and its *execution* according to the structure proposed by [11]. We then provide interesting insights on the *privacy by design* principle, *usage*, and *structural* quality to test the effectiveness of the proposed methodology to achieve compliance with the GDPR, according to the identified requirements (as described in Section 4).

Context. The primary limitation in the evaluation of the methodology is the availability of resources, which in this case, are limited to the students who participated in the evaluation. Although their skills were considered adequate for executing the methodology, the effectiveness of the approach could not be fully determined due to the restricted number of participants.

Goals and Strategy. The primary objective of this study is to determine the extent to which the proposed artifact effectively addresses RQ2. The evaluation strategy employed was naturalistic and ex-post. The ex-post evaluation approach was chosen because it involves assessing a completed artifact, reducing the likelihood of false positive results. However, it does require significant resources, time, and access to individuals or organizations involved

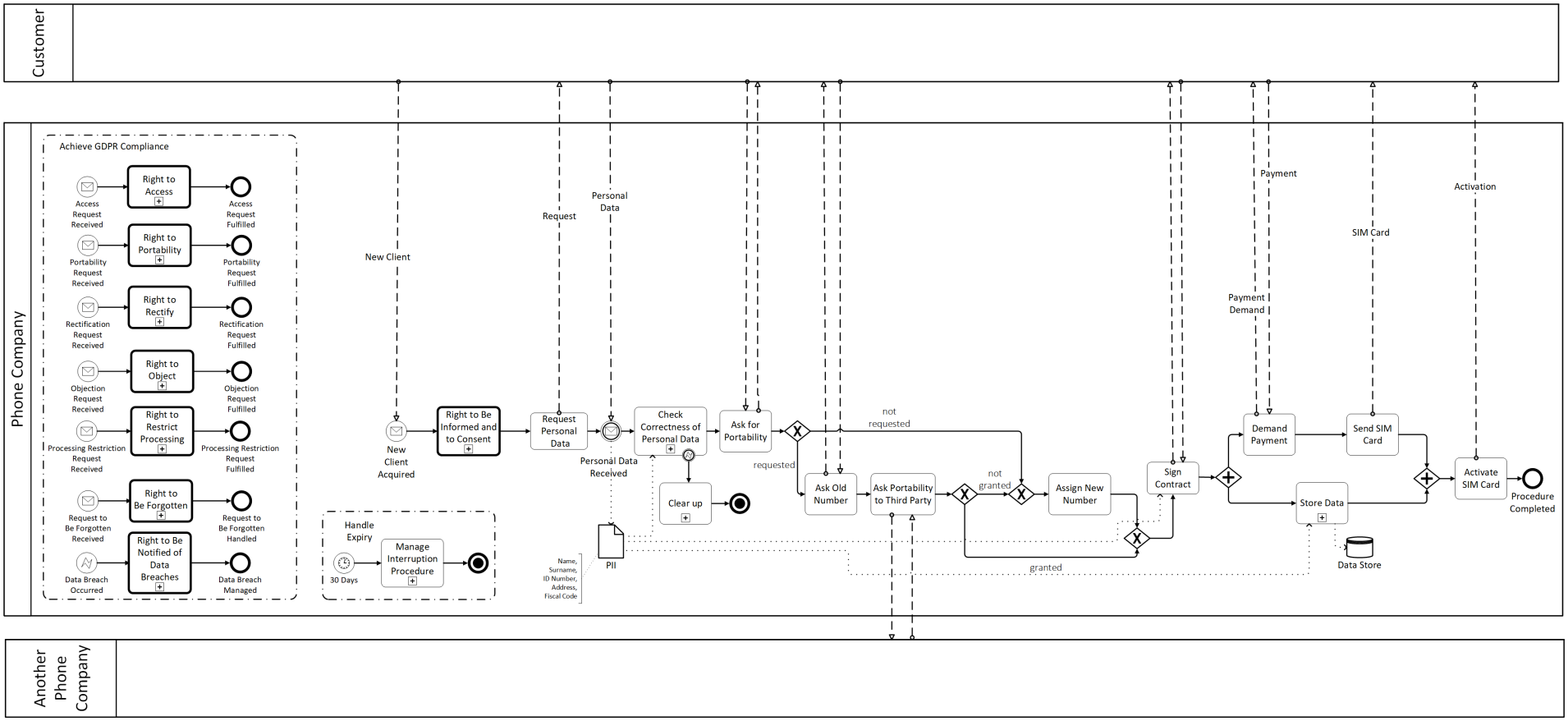


Figure 17: GDPR-compliant BPMN model for the case of the phone company

in the problem-solving process. Additionally, the evaluation follows a naturalistic approach that aims to study phenomena in their natural settings, without artificial manipulations or constraints. It involves observing and analyzing how people behave, interact, and make decisions in real-world contexts [25]. When conducting an ex-post and naturalistic evaluation, various approaches can be adopted, such as action research, case study, ethnography, phenomenology, survey, focus group, and participant observation. In this study, we opted for participant observation as the chosen method. This approach involves actively the participants as they interact with the artifact and solve the designated problems. By adopting participant observation, we were able to understand the users' experiences and behaviors about the artifact's functionality and effectiveness.

Design and Carry Out. In the evaluation process, students who had taken the courses of Enterprise Information Systems and Process Management and Mining at the XXX[6] University of Rome were selected as participants, as they were familiar with BPMN. The students were provided with a comprehensive list of potential privacy breaches that could occur during the phone company process and asked to redesign it (as described in Section 6) to meet privacy violations under the GDPR legislation. The test was conducted on white papers, and the participants were given one hour to complete it. The sample size consisted of 67 students, out of which 54 correctly inserted the patterns and understood both their position and type. The most common errors made by participants were inserting an inappropriate pattern (6 of 13 participants), inserting a pattern in the wrong position (5 of 13 participants), and losing one or more patterns (4 of 13 participants). Some participants (4 of 13) proposed alternative solutions to the provided pattern, which were ineffective in achieving the intended result. These findings suggest that some participants made more than one type of error.

Privacy by design. In Section 4, we first introduced the fundamental concepts of BPMN. Then, in Section 5 we proceeded to present a comprehensive list of design patterns that can be incorporated into the BP during the design or redesign phase of the BP life-cycle. By utilizing this strategy, the Data Controller gains access to the necessary tools to manage any potential request or breach related to personal data. Once these patterns are encoded within the BPMN associated with the BP model and placed in the appropriate locations, privacy by design can be achieved, as any required actions required at run time can then be promptly taken.

Usage Qualities. The results of the participants' observations indicate that most (81%) were able to correctly insert the patterns and understand their position and type. This indicates that most users were able to *comprehend* the artifact. The results do not directly indicate whether the artifact is customizable.

However, the fact that participants were asked to redesign the process so that it complies with GDPR legislation suggests that the artifact is *customizable* to meet specific privacy requirements. Indeed, GDPR patterns have been modeled employing the basic BPMN constructs that require no extensions of BPMN to capture them. This type of assessment helps to confirm actual learning capacity rather than perceived capacity (as in interviews and questionnaires). In this case, the fact that the participants were able to complete the task in an hour and without any particular problems while performing the test suggests that the task was relatively easy to *learn*. However, errors made in inserting patterns or proposing ineffective solutions suggest that certain aspects of the task could be clarified or made more intuitive to improve both learning and comprehensibility for those who struggled with it.

Structural Qualities. BPMN allows for the description of a BP model at various levels of abstraction by organizing the information within nested sub-processes. This crucial capability facilitates *modularity*, as security patterns can be separated into distinct sub-processes that are reusable across different use case scenarios. This approach results in a *concise* solution as message and exception handling within BPMN prevent the need to replicate patterns every time a security pattern is executed.

8. Related Work

With respect to work generally related to BPMN and security and privacy aspects, BPMN security extensions for healthcare processes are presented in [22] and [24]. [16] introduce security elements for BPMN to evaluate the trustworthiness of participants based on a rating of enterprise assets and to express security intentions such as confidentiality or integrity on an abstract level. In [8], BPMN is enriched with information assurance and security modeling capabilities. In [1], BPMN is aligned to the domain model of security risk management. In [19], Privacy Enhancing Technologies (PETs) are applied to enforce privacy requirements and support the analysis of private data leakage. In [23], the authors propose the SecBPMN-Q query language for representing security policies and a query engine that enables checking SecBPMN-Q policies against SecBPMN-ml specifications. Works by [9] and [14] are specifically related to the definition of extensions of BPMN to represent cyber security requirements. In [15], the authors investigate a new approach to modeling security and propose a solution to include all concepts potentially modelable in BPMN related to cyber security. In [2], the BPMN choreography models are used to detail

[6]The name of the company is not disclosed due to confidentiality constraints

message exchange and identity contract negotiation. In [6], BPMN is extended with access control, separation of duty, binding of duty, and need-to-know principles. Similarly to [6], in [12] privacy concerns are captured by annotating the BPMN model with ad hoc construct such as access control, separation of tasks, binding of tasks, user consent and the necessity to know novel icons. Contrary to these approaches, our methodology does not involve extending the BPMN language with extensions as such, thus maximizing the *learnability* and *comprehensibility* of the approach. In fact, the present paper shows that such extensions are not required to guarantee privacy by design as everything could be directly encoded in BPMN 2.0. This allows Data Controllers and BP designers to exploit already existing and customary BPMN modeling tools, which do not include the aforementioned extensions. As a result, our approach confirms that the design theory underlying BPMN formalism is adequate for solving GDPR model implementations.

Different from the above studies, our work is focused on GDPR. Specifically, we have provided an analysis of the main privacy constraints in GDPR and a set of design patterns for capturing and integrating such constraints in BP models represented in BPMN. Recent works concerning GDPR have been also presented in the BPM research literature. In [21], the authors propose a method to support the design of GDPR-compliant systems, based on a socio-technical approach composed of a modeling language and a reasoning framework. In [26], the authors present a model of GDPR that provides a visual overview of the associations between entities defined in the legislation and their constraints. In [17], the authors present an integrated framework for specifying legal knowledge employing well-known ontologies, such as LKIF (Legal Knowledge Interchange Format) ontology[7] and PrOnto for GDPR) and rule-based languages for modeling norms (e.g., LegalRuleML[8]), which are leveraged to detect or prevent violations of privacy rules using BPMN and the Regorous engine [10]. In [4], the authors propose to enrich BPMN with new annotations and connectors that express data protection requirements in a BP, and then to exploit at run-time a recommender system to supply auditors and supervisory authorities with a complete view of the BP and the procedures adopted for data protection. In [3], the authors describe how a blockchain-based automated tool can be used to support compliance checking and trust in BPs, enabling verification of GDPR obligations without the need for a trusted third party. In comparison with such approaches, the originality of our methodology lies in considering awareness of GDPR constraints at design-time, during BP modeling, and not as a run-time issue while automating the decision and implementation process of the patterns.

The latest work by [20] tackled the GDPR compliance problem by identifying a framework architecture where each component is delegated to ensure a specific GDPR requirement. They provided an extensive literary review showing that most of the GDPR literature does not implement all of the GDPR requirements. Still, the software architecture modeling language of choice cannot be used to effectively implement the GDPR requirements, which must be still expressed in natural language, as architectural patterns subdivide the software into different communicating components while no explicit behavioral requirements are explicit. On the other hand, the BPMN language allows us to subdivide the architecture into modular components (e.g., patterns) and proactively describes the activities and events occurring within the BP. Therefore, our modeling solution reveals to be more expressive than the aforementioned one.

9. Conclusion and Future Works

The enforcement of GDPR has revolutionized the way organizations address BPs, thus requiring them to understand how to cope with privacy issues and implement effective solutions to deal with such events. This paper provides a thorough explanation of how to build GDPR-aware BPs. Since it is evident that intervening at run-time on a BP instance when a breach occurs is adequate to tackle privacy issues only in the scope of the BP instance itself, in this paper, we have presented a methodology for making BP models GDPR-aware directly at design-time. We have adopted a systematic approach that supports a BP designer in inserting specific GDPR patterns in precise points of the BP model, thus anticipating and preventing (possible) violations of privacy constraints at run-time on any BP instance under execution. The novelty of our methodology is discussed in Section 5 with a dedicated related work analysis.

Despite the two proposed use cases showing that our methodology can be applied in different business contexts, the use cases could be further expanded with more examples of different contexts. In this direction, a first future work will consider an extensive validation of the patterns against a larger set of use cases, as this would be crucial to test the effectiveness of the overall methodology.

We proved the feasibility of designing a BP in BPMN that can cope with data violations as per the GDPR requirements by design. As second future work, we would like to implement the methodology into a tool supporting the work of BP designers. Such a tool should automatically identify points of the BP where a privacy violation can occur and propose where to place appropriate patterns inside the BP model. However, we would also need to

[7] https://github.com/RinkeHoekstra/lkif-core
[8] https://www.oasis-open.org/committees/legalruleml/

investigate whether a human-in-the-loop structure is required to detect the suitable patterns to be injected within the BP model. Furthermore, the development of such a tool will allow us to better quantify lawfulness, fairness, transparency, accuracy, and storage limitation embedded in the Data Subject's rights, respectively, to consent, be informed, rectify, and be forgotten. Purpose limitation, data minimization, and security could also be further assessed via this tool, thus achieving the accountability of the BP as a whole. Last but not least, we notice that even if our methodology specifically focuses on nine relevant constraints of GDPR, it can be easily extended to capture further constraints, thanks to its ability to be customizable, i.e., it can be applied to any BP modeled in BPMN handling personal data.

Acknowledgments

This work is supported by the H2020 project DataCloud (Grant number 101016835).

References

[1] O. Altuhhova, R. Matulevicius, and N. Ahmed. An Extension of Business Process Model and Notation for Security Risk Management. *Int. Journal of Inf. Syst. Modeling and Design*, 4(4), 2013.

[2] G. B. Ayed and S. Ghernaouti-Helie. Processes View Modeling of Identity-related Privacy Business Interoperability: Considering User-Supremacy Federated Identity Technical Model and Identity Contract Negotiation. In *Int. Conf. on Adv. in Social Net. Analysis and Mining*. IEEE, 2012.

[3] M. Barati and O. Rana. *Design and Verification of Privacy Patterns for Business Process Models*. Springer, 2021.

[4] C. Bartolini, A. Calabró, and E. Marchetti. GDPR and business processes: An effective solution. In *2nd Int. Conf. on Applications of Intelligent Systems (APPIS '19)*. ACM, 2019.

[5] D. Basin, S. Debois, and T. Hildebrandt. On purpose and by necessity: compliance under the GDPR. In *22th Int. Conf. on Financial Cryptography and Data Security*. Springer, 2018.

[6] A. D. Brucker. Integrating Security Aspects into Business Process Models. *Inf. Tech.*, 55(6), 2013.

[7] A. Capodieci and L. Mainetti. Business Process Awareness to Support GDPR Compliance. In *9th Int. Conf. on Information Systems and Technologies (ICIST'19)*. ACM, 2019.

[8] Y. Cherdantseva, J. Hilton, and O. F. Rana. Towards SecureBPMN - Aligning BPMN with the Information Assurance and Security Domain. In *4th Int. Workshop on BPMN*. Springer, 2012.

[9] M. E. Chergui and S. M. Benslimane. A Valid BPMN Extension for Supporting Security Requirements Based on Cyber Security Ontology. In *8th Int. Conf. on Model and Data Eng.* Springer, 2018.

[10] G. Governatori. The Regorous approach to process compliance. In *19th Int. Enterprise Distributed Object Computing Workshop (EDOC'15)*. IEEE, 2015.

[11] P. Johannesson and E. Perjons. *An Introduction to Design Science*. Springer, 2014. ISBN 978-3-319-10632-8.

[12] W. Labda, N. Mehandjiev, and P. Sampaio. Modeling of privacy-aware business processes in BPMN to protect personal data. In *Symposium on Applied Computing (SAC'14)*. ACM, 2014.

[13] M. Lohrmann and M. Reichert. Effective application of process improvement patterns to business processes. *Software & Systems Modeling*, 15(2), 2016.

[14] C. L. Maines, D. Llewellyn-Jones, S. Tang, and B. Zhou. A Cyber Security Ontology for BPMN-Security Extensions. In *15th Int. Conf. on Computer and Information Technology*. IEEE, 2015.

[15] C. L. Maines, B. Zhou, S. Tang, and Q. Shi. Adding a Third Dimension to BPMN as a Means of Representing Cyber Security Requirements. In *9th Int. Conf. on Developments in eSystems Eng.*, 2016.

[16] M. Menzel, I. Thomas, and C. Meinel. Security Requirements Specification in Service-Oriented Business Process Management. In *4th Int. Conf. on Availability, Reliability and Security*. IEEE, 2009.

[17] M. Palmirani and G. Governatori. Modelling Legal Knowledge for GDPR Compliance Checking. In *The Thirty-first Annual Conf. on Legal Knowledge and Information Systems*. IOS Press, 2018.

[18] S. A. Petersen, F. Mannhardt, M. Oliveira, and H. Torvatn. A Framework to Navigate the Privacy Trade-offs for Human-Centred Manufacturing. In *19th IFIP Conf. on Virt. Enterprises*. Springer, 2018.

[19] P. Pullonen, R. Matulevicius, and D. Bogdanov. PE-BPMN: Privacy-Enhanced Business Process Model and Notation. In *15th Int. Conf. on Business Process Management (BPM'17)*. Springer, 2017.

[20] M. Rhahla, S. Allegue, and T. Abdellatif. A Framework for GDPR Compliance in Big Data Systems. In *14th Int. Conf. on Risks and Security of Internet and Systems*. Springer, 2019.

[21] M. Robol, M. Salnitri, and P. Giorgini. Toward GDPR-Compliant Socio-Technical Systems: Modeling Language and Reasoning Framework. In *10th Conf. on Pract. of Ent. Mod.* Springer, 2017.

[22] A. Rodríguez, E. Fernández-Medina, and M. Piattini. A BPMN Extension for the Modeling of Security Requirements in Business Processes. *Trans. Inf. Syst. (IEICE)*, 90-D(4), 2007.

[23] M. Salnitri, F. Dalpiaz, and P. Giorgini. Designing secure business processes with SecBPMN. *Software and System Modeling*, 16(3), 2017.

[24] K. S. Sang and B. Zhou. BPMN Security Extensions for Healthcare Process. In *15th Int. Conf. on Computer and Information Technology*. IEEE, 2015.

[25] Y. Sun and P. B. Kantor. Cross-evaluation: A new model for information system evaluation. *J. Assoc. Inf. Sci. Technol.*, 57(5):614–628, 2006.

[26] J. Tom, E. Sing, and R. Matulevicius. Conceptual Representation of the GDPR: Model and Application Directions. In *17th Int. Conf. on Perspectives in Business Informatics Research*. Springer, 2018.

Appendix A.

Here is a description of the second real use case handling personal data before enacting and implementing the required BPMN patterns for GDPR compliance.

The BPMN model in Fig. A.18 represents a generic *hiring company* receiving an application request for a job position. The BP starts when someone applies for the position. If the applicant has an active account, the company asks them to log in with their credentials. Otherwise, the company requests the user's data (e.g. name, surname, email address, phone number, etc.) and, after having checked the correctness of such data (if it is not correct the company should ask it again), both proceeds with the creation of an account and sends a confirmation email to the user. After that, the applicant can log in with their new credentials. Once the user is logged in, the hiring company retrieves the applicant's data and asks them to send a CV and Cover Letter. CV and Cover Letter are then stored in a database that the recruiter will access later on in the BP. In the meantime, the company sends the applicant some personality, behavioral, and ability tests to take when they wish. Such tests will be then analyzed by a profiler and the results will support the recruiter in the evaluation. A phone interview is scheduled if the applicant is deemed suitable for the job position. Otherwise, a rejection email is sent. In either case, the hiring procedure is complete. If, for some reason, the procedure takes more than 3 months to complete, then the BP is interrupted and the application will not be taken into consideration. Finally, please note that the exchange of personal data does not imply proper handling of personal data in accordance with the GDPR regulation.

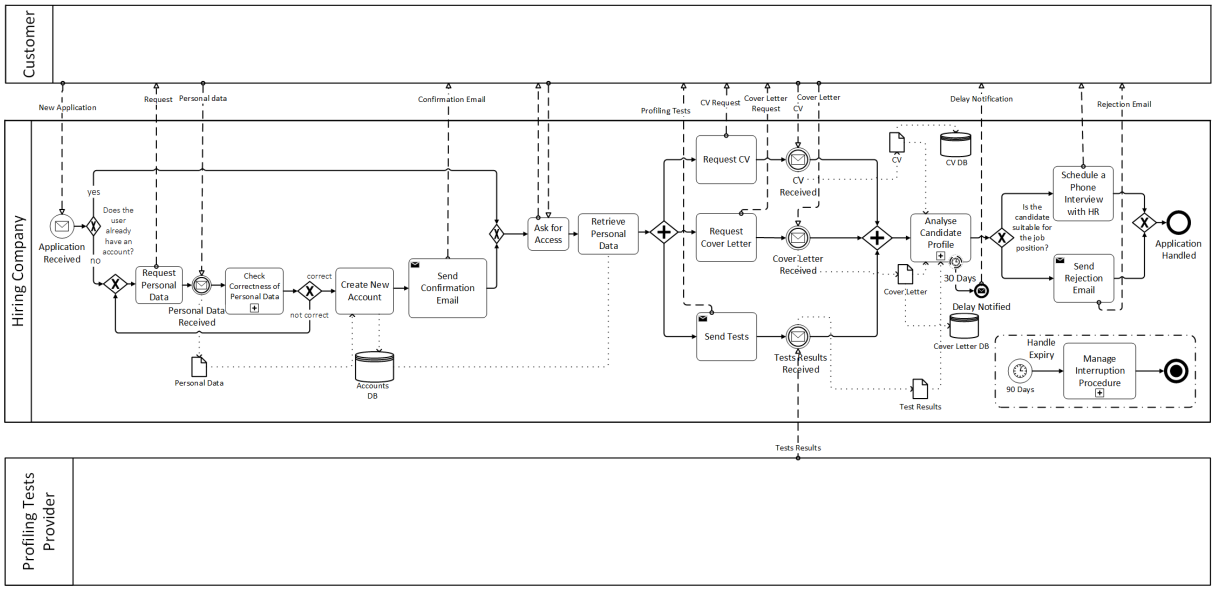


Figure A.18: BPMN model for the case of the hiring company

In the following, we show how applying the methodology to the use case enables the transformation of the BP model that is not compliant with GDPR in its compliant counterpart.

The BPMN model provided above does not take into account privacy concerns yet, thus, after the coming into effect of GDPR, on 25 May 2018, if the company has to evaluate the profile of a candidate located in the EU, the whole BP model has to be modified to become GDPR compliant. Since personal data are processed and have to be retrieved, it is mandatory to obtain consent from the subject to have their data processed. Therefore, the BP designer of the company has to introduce in the model the request for consent, given that it has not been contemplated yet. Such a request has to be provided to the user together with a list of privacy information so that the applicant is informed about how their data will be processed, by whom, and for what purpose. Consent and privacy information are included in the pattern *Right to Be Informed and to Consent*, which has to be placed before the first request for personal data occurs. In this case, the three PIIs identified in the model are the personal data such as name, surname, birth date, address, etc., the CV, and the test results provided to the hiring company by a tests provider in the form of a profile of the applicant. During the execution of the BP, the first activity that requests personal data is the one associated with the filling module for personal data during the creation of a new account. At this moment, the pattern should be introduced and it should include consent for all future requests for personal data. In fact, it makes sense that consent is requested when a new account is created, thus for any future job application, consent has already been provided and data can be retrieved more easily. Nevertheless, consent could be withdrawn at any moment, thus the pattern *Right to Object* has to be provided in the event sub-process. In the same big event sub-process, all patterns, and exceptions made for *Right of Portability*, have to be included. Most and foremost pattern *Right to Object to Automated Processing* should be executable at any moment, because,

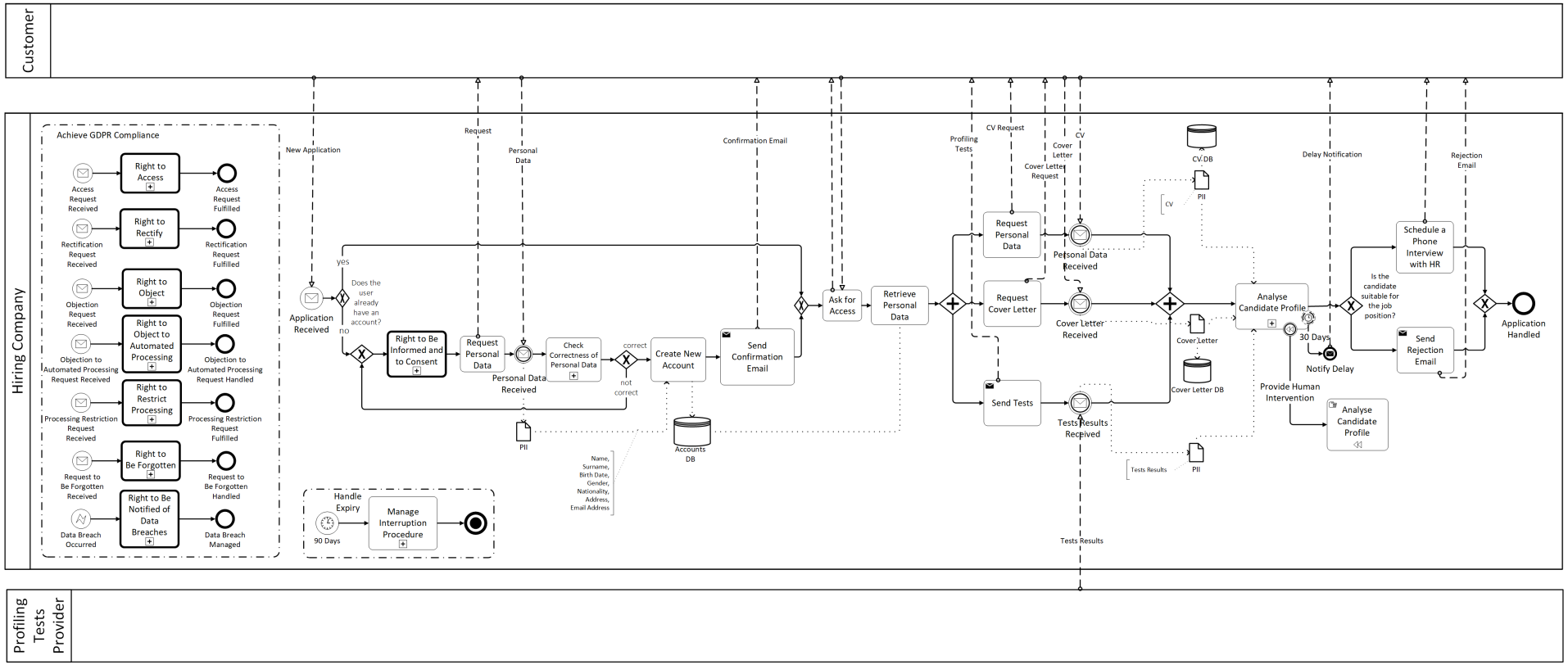


Figure A.19: GDPR-compliant BPMN model for the case of the hiring company

being the hiring procedure based on profiling, a request for human intervention instead of automated decisions is very likely to occur. To ensure the execution of the pattern, every activity taking a decision based on automated mechanisms must dispose of a compensation handler, enabling human intervention in the decision. This is what has been done with the activity "Analyse Candidate Profile", whose compensation activity is the manual activity named "Analyse Candidate Profile" as well. The GDPR-compliant BPMN model for the hiring procedure is illustrated in Fig. A.19.