On Purpose and by Necessity: Compliance under the GDPR*

David Basin[1], Søren Debois[2], and Thomas Hildebrandt[2]

[1] ETH Zürich
basin@inf.ethz.ch
[2] IT University of Copenhagen
{debois,hilde}@itu.dk

Abstract. The European General Data Protection Regulation (GDPR) gives primacy to *purpose*: Data may be collected and stored only when (i) end-users have consented, often explicitly, to the purposes for which that data is collected, and (ii) the collected data is actually necessary for achieving these purposes. This development in data protection regulations begets the question: how do we audit a computer system's adherence to a purpose?
We propose an approach that identifies a purpose with a business process, and show how formal models of interprocess communication can be used to audit or even derive privacy policies. Based on this insight, we propose a methodology for auditing GDPR compliance. Moreover, we show how given a simple interprocess dataflow model, aspects of GDPR compliance can be determined algorithmically.

1 Introduction

Data protection is now heavily anchored in national and international law. The prime example of this is the European General Data Protection Regulation (the GDPR) [9], which strengthens previous data protection directives to give individuals more rights on how their personal data is processed. A central principle of data protection in general, and the GDPR in particular, is that organisations collecting and processing personal data must be explicit about how the data will be used and *the data is actually used for the purposes for which it was collected.*

Contrast this situation with standard access control, which regulates who may carry out which operations in a system. With few exceptions (e.g., history-based access control or access decisions incorporating environmental attributes) access is independent of context: if Alice has the right to access Bob's bank account balance, then she can do this for *any* purpose. This includes both intended purposes, such as serving as his customer relations manager, or unintended ones, such as selling information on his financial status to credit agencies. Modern data protection calls for something more: *access control relative to a purpose.*

* Authors listed alphabetically. This work is supported in part by Innovation Fund Denmark, grant 7050-00034A, project "Effective, co-created & compliant adaptive case management for knowledge workers" (EcoKnow).

The key difficulty then is that mainstream programming technologies do not leave us any obvious representation of purpose, much less one that can be reasonably related to sites of data collection or data use. There is, however, one area of computer science where a notion of purpose takes a center stage, and where formal models abound: The study of Business Process Management, in particular Business Process Modelling. Here, the operations (both IT and human) of a business are modelled in one of a variety of formal languages including Statecharts [11], UML [21,22], BPMN [2], GSM [13], CMMN [23], DECLARE [1,25], or DCR graphs [12,19]. Such a model will include details about both data collection and data use. Crucially, most definitions of "business process" *also* either implicitly or explicitly include the purpose of the process, although sometimes expressed in terms of a product to be manufactured or a service to be rendered.

We propose exploiting the formal notion of a business process model as a bridge between a system implementation and the GDPR. In doing so, we exploit that a business process model by its very nature embodies a particular purpose, while at the same time it specifies at what points data is collected and used. For instance, an online-shop will have an order-fulfilment process where a customer's address is used to ship a product. Our proposal conflates a formal model of that *process* with the *purpose* of order-fulfilment; the model then describes both data collection and data use.

This idea poses a challenge to formal business process models. Under the GDPR, we must account for the data transferred between processes: data collected for one purpose and used for another. For example, a mailing address might be collected in a customer registration process that is subsequently used in an order-fulfilment process. Typical process models do not give detailed accounts of such inter-process interactions. We posit that an interprocess dataflow model is *necessary* to audit GDPR compliance.

We show that formal models of interprocess communication enable the algorithmic verification of parts of GDPR compliance. However the GDPR in certain cases goes *beyond* what we can automatically verify. For example, it can be difficult to determine whether a text message is an advertisement or a notification about an upcoming delivery. In these cases, the underlying business processes themselves must be augmented with human actions, for example explicit manager approval of the text message. Our approach therefore supports automated compliance checking complimented by human actions when necessary.

In summary, we make the following contributions:

1. We show how a mechanism for relating *purpose* to implementation artefacts is necessary to demonstrate compliance with the GDPR (Section 4.1).
2. We put forward the idea of *identifying* a business process and a purpose (Section 4.2).
3. We identify inter-process communication as key to GDPR compliance (Section 4.3).
4. We propose a *methodology* for auditing GDPR compliance by decomposing the audit into verifying the compliance of an implementation to a process

model, of a process model to a privacy policy, and of these latter two to the GDPR itself (Section 4.4).
5. We show how a formal process model allows us to verify compliance of certain aspects of the GDPR algorithmically (Definitions 5.1 and 5.4). In particular, we can *generate* compliant privacy policies.
6. Finally, we illustrate through examples that GDPR compliance cannot be fully achieved by algorithmic means, and that process models can fill the gap here by specifying needed human actions (Section 5.4).

2 The General Data Protection Regulation

The GDPR [9] was passed on April 14, 2016 and will come into force May 25, 2018. It embodies a major departure from current practices. It requires not only that data is only collected after obtaining consent from the user, but also that data is collected and used *only for specific purposes*, and *must be deleted when those purpose are no longer applicable*. The GDPR spells out these requirements in its notions of *purpose limitation* and *data minimisation*, its treatment of *consent*, and the *right to be forgotten*.

Purpose limitation [9, Article 5, §1(b)]:

> *"[Personal data shall be] collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes; [...]"*

Data minimisation [9, Article 5, §1(c)]:

> *"[Personal data shall be] adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed [...] "*

Consent (and its connection to purpose) [9, Recital (32), emphasis ours]:

> *"Consent should be given by a clear affirmative act establishing a freely given, specific, informed and unambiguous indication of the data subject's agreement to the processing of personal data relating to him or her [...]* ***Consent should cover all processing activities carried out for the same purpose or purposes. When the processing has multiple purposes, consent should be given for all of them."***

Right to be forgotten [9, Article 17, §1]:

> *"[...] the controller shall have the obligation to erase personal data without undue delay where one of the following grounds applies:*
> *(a) the personal data are no longer necessary in relation to the purposes for which they were collected or otherwise processed;*
> *(b) the data subject withdraws consent [...]"*

We remark that the GDPR mandates access control [9, Article 25, §1]:

> "*The controller shall implement appropriate technical and organisational measures for ensuring that, by default, only personal data which are necessary for each specific purpose of the processing are processed. [...] personal data are not made accessible without the individual's intervention to an indefinite number of natural persons.*"

Finally, the GDPR has teeth. It imposes two levels of fines, depending on which parts of the GDPR are violated. The highest level imposes fines of up to 20 million EUR or 4% of the organisation's world-wide turnover, whichever is higher [9, Article 83, §5]. Processing without consent is on the list of high-level infringements [9, Article 83 §5(a)].

For the purposes of the present paper, we shall emphasize the above concepts of purpose limitation, data minimisation, consent, and the right to be forgotten. However, the GDPR confers on citizens (data subjects) a remarkable range of additional rights, such as rights of "access" and "data portability" [9, Article 15 & 20] and the right to be notified of data breaches [9, Article 33].

In summary, for data collection to be GDPR compliant, the data must:

1. be collected for a purpose,
2. to which the user has consented, and
3. be necessary to achieve that purpose;
4. moreover the collected data must be deleted when it is no longer necessary for any purpose.

Contrast to extant privacy policies. Privacy policies are statements about how an organization collects, processes, and more generally manages, the personal data of its customers or other individuals. An informal survey of existing policies (including Facebook [8], Google [10], and IBM [14]) shows that their essence effectively consists of two types of declarations:

- **The kinds of data collected**, e.g., credentials, cookies, purchases, etc.
- **How collected data is used**, e.g., to process orders, personalize offerings and advertisements, etc.

These statements may be augmented with additional information, such as how *non*-personally identifiable information may be used, which usages one may opt out of, security measures taken when storing and processing data, and the like.

From the above, we conclude that current best-practice is to formulate *coarse grained* privacy policies. Their essence amounts to two sets, a set DC of the kinds of data collected and a set DU of data usages. In some cases (e.g., Google), a relation (a subset of $DC \times DU$) is given, where it is indicated how particular data items are used, e.g., "we use information collected from cookies to improve your user experience." However, in most cases (e.g., Facebook, IBM) the description of DC is non-specific with respect to which data is involved in which usages, e.g., "The information you provide may be used for marketing purposes."

The GDPR requires more. For example, Recital 39 specifies that the purposes that data will be used must be transparently laid out in a privacy policy. In particular, it should be clear that the personal data should be "adequate, relevant, and limited to what is necessary for the purposes for which they are processed." Indeed, "personal data should be processed only if the purpose of the processing could not reasonably be fulfilled by other means." This requires *fine grained* privacy policies that clearly elucidate purposes and the associated data required. Our thesis is that business process models provide the right basis for this elucidation, both supporting the creation of fine-grained natural language privacy policies (e.g., informing data owners) and supporting audit and compliance (e.g., informing technical specialists).

3 Running example

We provide an example from on-line retailing that we subsequently use to illustrate our methodology to audit compliance with the GDPR. An on-line retailer has customers who order goods using the retailer's homepage, pay with their credit cards, and expect to subsequently receive their orders by post. The retailer may engage in marketing, targeted or otherwise, using channels such as web-advertisements and e-mail.

We will focus just on the core processes of such a retailer, emphasizing what data is collected and used. These core processes are:

Register Customer: A prospective customer signs up with an on-line retailer. As part of this process, the customer provides his e-mail, his mailing addresses, and his credit card information.

Purchase: A registered customer selects a product on the retailer's homepage, pays using the recorded credit card number, and the retailer subsequently sends the product and invoice.

Mass Marketing: A customer's e-mail or physical address is used to send otherwise un-targeted advertisement.

Targeted Marketing: A customer's e-mail or physical address and past purchase history is used to send individually targeted advertisements.

In the following, we write processes in sans-serif, e.g., Mass Marketing, and descriptions of data classes in brackets, e.g., ⟨credit card number⟩.

In Figure 1 we show example models of these processes, written in the BPMN [2] notation. In brief, the diagram comprises four pools, one for each process; inside each pool is a number of activities in boxes, some human, some automated. Activities with white envelopes take incoming messages, typically user input; black envelopes produce outgoing messages, typically output to the user. Activities marked with a person icon are undertaken by humans. The sequencing of activities is indicated with solid arrows between them. Activities may both produce and consume data stored in databases, indicated by dashed arrows. Note that the databases allow data to be shared between processes.

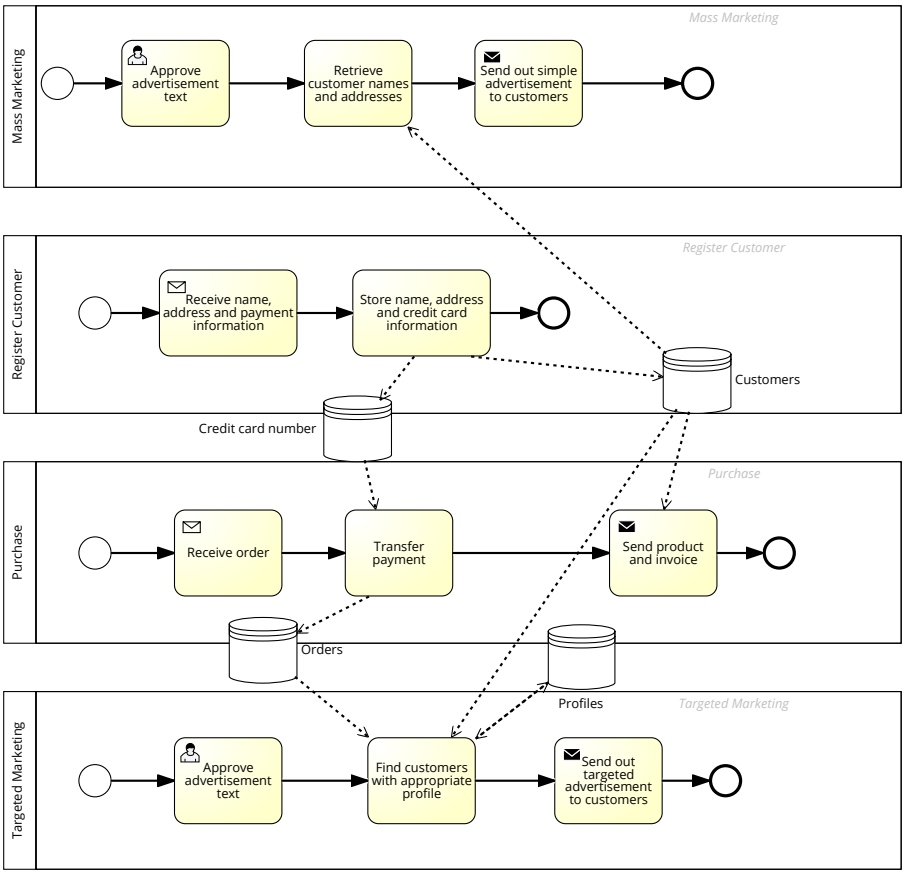

Fig. 1. BPMN model of the core on-line retailer processes.

4 Purposes

The GDPR's emphasis on *purpose* has interesting and subtle ramifications for both the design and audit of computer systems. In this section, we develop the idea that business process models encode purposes, and that this encoding can be used to analyse compliance with the GDPR.

4.1 Purpose and compliance

As with any regulation that applies to computer systems, we are faced with two key questions:

- How do we *build* a computer system in a manner guaranteeing compliance?
- How do we *analyse* or *audit* a computer system for compliance?

Reviewing the conditions (1)–(4) from Section 2, which are required for a data-collection to be compliant with the GDPR, we see an immediate problem: The notion of compliance revolves around the notion of purpose, but purpose tends *not* to have an explicit representation in contemporary computer system implementations. Whereas notions like "user authentication" or "order management" usually have an *explicit* representation as lines of code and tables in databases, we seldom see implementation artefacts representing "the purpose" of a particular dataset. But to answer the two questions above, we must not only be able to identify the points at which data is collected (which is presumably easy), we must *also* associate that data with a purpose.

For example, suppose our on-line retailer collects a user's e-mail address during registration. There are legitimate uses for such information: it may be used as a user-id, to send invoices, or for password resets; alternatively, it might also be used for illegitimate purposes such as unsolicited marketing. By examining the system's code, we will readily discover where data may be collected and processed. However, it is impossible to determine from the code alone whether data being collected at a particular point is personal data or not, for what purpose data is being processed, and if it is really necessary for that purpose.

Even if we can technically determine every place in the implementation that accesses this e-mail address, we still may not be able to determine the *purpose* for that access. For example, the Mass Marketing process of our e-shop could enable staff to send arbitrary messages to *every* registered customer. Obviously, we cannot statically determine what the *purpose* of these messages might be: A staff member might send important information about deliveries ("Due to strikes at our logistics partner, all deliveries will be delayed."); marketing messages ("You have ordered recently from us. How about also buying an electric cat food dispenser?"); or even political propaganda ("Vote for me for president!").

Finally, we must delete data when it has served its purposes. But it is difficult to know when this is the case, especially in large computer systems where the same data may be used in multiple subsystems, for multiple purposes.

4.2 Business processes as purposes

We propose using the business processes [5] that the computer system in question supports to identify purposes and to classify the types of data collected. The key insight is that business processes *explicitly represent one or more purposes.* Here is a standard definition of a business process [5, pp. 5–7] (emphasis ours):

> "*a structured, measured set of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how work is done within an organization, in contrast to a product focus's emphasis on what. A process is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for action. [...]* ***Processes are the structure by which an organization does what is necessary to produce value for its customers*** *[...]*"

The emphasized sentence highlights that a business process comprises both a purpose—the specifics of "to produce value for its customers"—as well as concrete steps—the specifics of "the structure by which." In practice, the purpose of a process will be most clearly represented by its title or perhaps a brief natural language statement. Through the description of "how work is done," a process description also gives us the necessary information about what data is collected, and where it is used. Moreover, when purposes are processes, we can determine when a purpose is served, e.g., when the corresponding process has terminated.

In general, it is reasonable to associate a "process" with a "purpose", e.g., the purchase process/purpose, the mass marketing process/purpose, the customer satisfaction evaluation process/purpose, or the warranty process/purpose.

4.3 Inter-process communication

In practice, a company may collect data about customers in one process and use that data in another. In our example, the Customer registration process collects ⟨credit card number⟩ and customer information (name, email, and physical address) that we simply refer to as ⟨customer⟩, but it does not itself use these. They are instead used by the Purchase, Targeted marketing, and Mass marketing processes. This disconnect mirrors a challenge faced by many companies: whereas the individual processes within the company are usually well-understood by the staff undertaking them, including the interfaces to other processes, the global picture of *all* processes in the company is rarely well-understood. But the GDPR requires such a global understanding: data collected in one process may migrate to other processes, and end-user consent is required for *all* involved processes.[3]

We propose that for contemporary process models to be truly useful for GDPR analysis, we must interpret *collections* of processes as data-flow graphs. We therefore introduce the following simple model of process collections.

Definition 4.1 (process collection). *A* process collection *PC is a tuple $PC = (P, D, DU, DC)$ comprising:*

1. *a set P of processes,*
2. *a set of data classes D,*
3. *a relation $DC \subseteq D \times P$ specifying what data is* collected *by which processes, and*
4. *a relation $DU \subseteq D \times P$ specifying what data is* used *by which processes.*

Note that the set D of data classes is not a set of data values per se, but rather a set of the possible kinds of data, e.g., addresses, credit card numbers, etc.

Example 4.2. We construct a process collection modelling the example from Section 3 by lifting the informal description of the example to the process collection QC given formally in Figure 2 and represented visually in Figure 3. In the

[3] Notice that without the anchor of processes-as-purposes, this problem is hardly solvable in practice. For example, what are the purposes the user consents to for the hundreds of computer systems running at a large corporation?

$$
\begin{aligned}
P &= \{\mathsf{Purchase}, \mathsf{Register\ Customer}, \mathsf{Targeted\ Marketing}, \mathsf{Mass\ Marketing}\} \\
D &= \{\langle\text{customer}\rangle, \langle\text{credit card number}\rangle, \langle\text{order}\rangle, \langle\text{profile}\rangle\} \\
DU &= [\mathsf{Purchase} \mapsto \{\langle\text{customer}\rangle, \langle\text{credit card number}\rangle\}; \\
&\quad\ \mathsf{Register\ Customer} \mapsto \emptyset; \\
&\quad\ \mathsf{Targeted\ Marketing} \mapsto \{\langle\text{customer}\rangle, \langle\text{order}\rangle, \langle\text{profile}\rangle\}; \\
&\quad\ \mathsf{Mass\ Marketing} \mapsto \{\langle\text{customer}\rangle\}] \\
DC &= [\mathsf{Purchase} \mapsto \{\langle\text{order}\rangle\}; \\
&\quad\ \mathsf{Register\ Customer} \mapsto \{\langle\text{customer}\rangle, \langle\text{credit card number}\rangle\}; \\
&\quad\ \mathsf{Targeted\ Marketing} \mapsto \{\langle\text{profile}\rangle\}; \\
&\quad\ \mathsf{Mass\ Marketing} \mapsto \emptyset]
\end{aligned}
$$

Fig. 2. The use of personal data in an online retailer: process collection $QC = (P, D, DU, DC)$ corresponding to the example of Section 3. To conserve space, the relations DU and DC have been represented as maps.

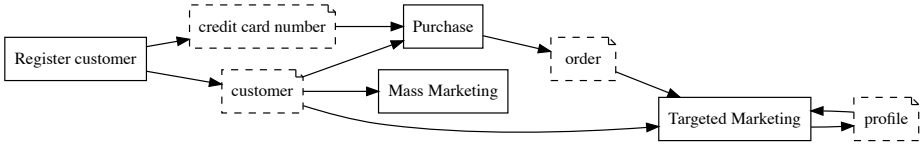

Fig. 3. Graphical representation of the process collection QC of Figure 2.

latter figure, the processes of the retailer are rendered as square boxes labelled Purchase, Register Customer, Targeted Marketing, and Mass Marketing. Data is written as dashed, dog-eared boxes, e.g., ⟨customer⟩, ⟨credit card number⟩, and ⟨order⟩. Data use and collection is indicated by arrows between process boxes and data boxes:

1. An arrow from a process to data indicates that the process collects and stores the given data, e.g., the Register Customer process records the new customer's contact information (name and address) in the data class ⟨customer⟩.
2. An arrow from data to a process indicates that the process uses the given data, e.g., the Purchase process uses the customer's contact information in ⟨customer⟩ and the payment information in ⟨credit card number⟩.

For example, the Targeted Marketing process uses the order data ⟨order⟩ and produces the personal ⟨profile⟩ of the customer. The Mass Marketing process similarly uses the customer data, but it does not use the orders.

In this example, we derived the process collection from the informal description of the on-line retailer. In general, process collections can be extracted (even

automatically) from formal process models such as the BPMN diagrams of Figure 1.

Example 4.3. Consider the BPMN models in Figure 1. The set P of processes is the set of labels of lanes. The set of data classes D is the set of labels of database access lines. The relations DC and DU are given by dashed lines from processes to databases (DC) and databases to processes (DU).

We can use the data production and usage relations to derive which user consents are needed. For example, the on-line retailer must acquire consent to use the customer data for future purchases and mass marketing. If we know when processes can no longer be started, we can also use the relations to infer when data must be deleted or made non-personal, for example by anonymising it. We shall pursue this idea further in Section 5.3.

4.4 A methodology for auditing for the GDPR

We now propose a methodology for auditing for the GDPR. Our methodology has the following inputs. First, at the lowest level, we require an *implementation*, say, written in Java, of the system under consideration. Second, we require *a collection of process models* describing the system's behaviour, from which we can produce a formal process collection (Definition 4.1). Finally, we require a user-facing *privacy policy*. Recall from Section 2 (transparency, consent) that this is required by the GDPR.

To establish GDPR compliance, we must show the following:

1. The implementation conforms to the process collection. That is, the implementation implements the processes described in the process collection.
2. The process collection conforms to the privacy policy. That is, the processes described actually treat data in the manner described by the privacy policy.
3. The process collection conforms to the GDPR. That is, the processes described follow the GDPR, for example they delete data as appropriate.
4. The privacy policy conforms to the GDPR. That is, the privacy policy does not make statements outside the GDPR, such as "we collect your personal information and use it for undisclosed purposes."

We illustrate the required conformance relations in Figure 4.

Following this methodology ensures that the purpose limitation is upheld because the implementation collects and uses data as specified by the process collection (1); the process collection uses data as specified in the privacy policy (2); and both the privacy policy and process collection conform to the GDPR (3,4).

The difficulty of these steps depend on the exact nature of the implementation. For example, for Step (1), if the implementation has a collection of BPMN models as its specification, the compliance of that collection to an implementation might be reasonably assumed or spot-checked by an auditor. If the implementation is based on a BPMN process engine or a Statechart interpreter [11], the process might be automated. Alternatively, a process collection might be

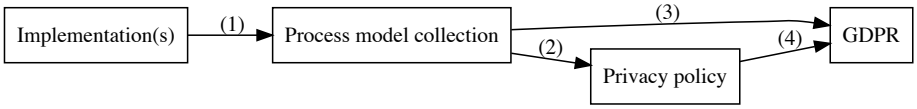

Fig. 4. Conformance requirements. An arrow $A \longrightarrow B$ means "A conforms to B".

obtained from an informal requirements specification (indicating processes) and a dataflow analysis [16] of a mainstream programming-language implementation (establishing collection and use).

In the latter case, where an informal requirements specification is the starting point, Step (1) involves establishing that a process collection is a dataflow model of a program. This problem is in general undecidable, so this step entails approximating dataflow. But what kind of approximation do we need? An under-approximation would leave DU smaller than it really is: Some pair (d, p) is not in DU even though the system uses data d for the purpose p, in violation of GDPR consent requirements. Conversely, an over-approximation would leave DU larger than it really is: Some pair (d, p) is in DU even though the system does not use data d for the purpose p, in violation of the purpose limitation.

These observations point to a curious problem at the intersection between computer science and law: if static analysis cannot determine whether data will or will not be used, is there a violation of the purpose limitation? Here, we take the pragmatic solution that the inclusion of a data usage in a process collection means the *possible* use of that data, and leave for the human audit to verify that this usage may indeed happen in the implemented process.

Finally, we note that if we omitted the process-model middleman, some other means would be required to relate purposes to implementation artefacts.

5 Algorithmic verification of compliance

We demonstrate in this section how parts of our methodology for demonstrating GDPR compliance can be supported or even achieved algorithmically.

5.1 Consent statements

We saw in Section 2 that the GDPR requires consent to the collection of data for specific purposes. We also saw how current privacy policies do not support this, primarily by failing to distinguish between different kinds of data and the purposes they are used for. For example, because Mass Marketing uses ⟨customer⟩, a privacy policy compliant with the GDPR must include words to the effect "we use your customer contact information (name and address) for mass marketing," indicating for what purpose the customers' contact information is used.

We now show how conformance of a process collection to a privacy policy (Part 2 of Figure 4) can be decided algorithmically. Recall that the DU component of a process collection PC comprises a set of pairs (d, p), where d is a

class of data and p is a process using that data. Assuming that PC adequately models the underlying processes, that is, DU comprises exactly the data used by the processes, we can automatically generate the corresponding privacy policy.

Definition 5.1 (Data purpose). *Let $PC = (P, D, DU, DC)$ be a process collection. A* data purpose *DP is a relation $DP \subseteq D \times P$. For a data purpose DP, the* privacy policy $\mathsf{pp}(DP)$ *is the set of statements*

"We use d for p,"

for each $(d, p) \in DP$.

That is, a data purpose associates data with processes. Note that the above definition may associate the same data with *multiple* uses, e.g., users would typically be presented with a consent statement like "we collect d_1, d_2 for purpose p_1 and we collect d_1, d_3, d_4 for purpose p_2."

d	p
⟨customer⟩	Purchase
⟨credit card number⟩	Purchase
⟨customer⟩	Mass Marketing
⟨profile⟩	Targeted Marketing
⟨order⟩	Targeted Marketing
⟨customer⟩	Targeted Marketing

Fig. 5. Privacy policy $\mathsf{pp}(DU)$ for the process collection $QC = (P, D, DU, DC)$ of Figure 2.

Example 5.2. Let $QC = (P, D, DU, DC)$ be the process model of Figure 2. Then DU comprises the pairs in Figure 5. Using Definition 5.1, and allowing meaning-preserving natural language transformations, the Targeted Marketing privacy policy reads: "We collect your customer information (name, address), order history, and profile, and use them to send you targeted advertising."

The notion of data purpose (Definition 5.1) naturally orders the possible data purposes by set inclusion.

Lemma 5.3. *Let D be a universe of data and P a collection of processes. Then the possible data purposes form a lattice under the subset-relation, i.e., $DP \sqsubseteq DP'$ iff $DP \subseteq DP'$.*

Proof. Immediate from Definition 5.1

The lattice ordering provides a means of formalising privacy policies where users give consent to some, but not all, purposes supported by the system. For

a process collection $PC = (P, D, DU, DC)$, DU is the maximal data purpose; asking anything more is in effect asking users for permission to data that the underlying processes do not actually use, violating purpose limitation.

Returning to the present setting where users must consent to all purposes, we note that DU is in fact also the *least* admissible data purpose: If users consent to strictly less than DU, there must be a pair $(d, p) \in DU$ the user did not consent to. Hence the system violates the GDPR requirements on obtaining consent.

Besides the outright generation of privacy policies, we can also use this observation to check an existing privacy policy for correctness: Simply extract from the policy the appropriate set of pairs $\{(d_1, p_1), (d_2, p_2), (d_3, p_2)\}$ and compare it with DU.

5.2 Data minimisation

In the last section, we generated data purpose statements algorithmically from process models. However, these statements are GDPR compliant only if they mention all the data *used* by the process. As we saw in Section 2, the GDPR also requires that all of the data collected is *necessary* for the stated purpose.

We caution that necessity is a slippery concept. For example, one may ask whether an online merchant really *needs* my credit card number given that sending an invoice might satisfy the same purpose of collecting payment. We shall leave such fine distinctions for the auditors.

We can however determine algorithmically some classes of *un*necessary data: we can check whether data that has been collected is in fact also used. If not, that data is clearly unnecessary, violating data minimisation. This information will help a human auditor quickly judge the conformance arrow (3) in Figure 4.

Definition 5.4 (Used data). *Let $PC = (P, D, DU, DC)$ be a process collection and let $d \in D$ be data for PC. We say that d is* used *iff for some $p \in P$ we have $(d, p) \in DU$.*

In other words, d is "used" if some process uses it according DU.

Example 5.5. Returning to the process collection QC of Figures 2 and 3, it is straightforward to verify that no collected data is unused. However, if we did not have the Targeted Marketing process, then ⟨profile⟩ would not be present at all and ⟨order⟩ would not be "used" in the sense of Definition 5.4. Consequently, either the order data could not be legally stored in an order data base or the process collection is incomplete.

We remark that the data used is computable in time polynomial in the size of $\mathcal{L}(PC)$ under reasonable assumptions about representation:

Proposition 5.6. *Let $PC = (P, D, DU, DC)$ be a process such that P and DU are finite, and assume that P and DU are represented as sequences of their elements. Then computing whether any d of D is used is possible in time polynomial in the sizes of P and DU.*

Proof. Let $d \in D$ be given. Observe that d is used iff for some $(d', p) \in DU$ we have $d' = d$. Given a pair (d', p), we can determine in time $\mathcal{O}(|d|)$ whether $d' = d$. We can then compute whether d is used by iterating over the elements of DU in time $\mathcal{O}(|d| * |DU|)$.

5.3 Deletion

We saw in Section 2 that the GDPR, via the right to be forgotten, requires that data must be deleted on request, provided that the purposes for which consent has been given no longer apply. Since we have identified purposes and processes, this would be when either (i) no currently running process uses the data, and (ii) no process that may be started in the future uses the data.

For practical purposes, determining the set of (non-)applicable processes is often straightforward. In many web-services, the set of applicable processes is all or nothing: all the services are offered until the user deletes his account, at which point no services are offered.

Example 5.7. In our running example of an on-line retailer, we assume that consent is given before the customer inputs personal data, i.e. in the first activity in the Register customer process given in Figure 1. After a customer is registered, purchases and marketing may be started indefinitely. Implicit in our process collection model is that once the user deletes his account then no more processes can be started (equivalently, no more purposes can be activated) and the user's data must now be deleted.

5.4 Human verification of compliance

We have seen in the preceding subsections that some aspects of GDPR compliance can be verified algorithmically. However, in Section 4.1 we explained that other aspects cannot: We cannot distinguish algorithmically between, e.g., marketing messages and political propaganda. To enforce the purpose limitation in such cases it may be necessary to add *human* enforcement activities.

We saw an example already in the BPMN processes given in Figure 1: the Mass Marketing process includes a human activity "Approve advertisement text", whereby an authorised staff member confirms that the proposed advertisement text is in fact an advertisement.

This ability to model both automated and human activities is unique to business process models. This makes them particularly well-suited for the analysis of GDPR compliance: A model of only the computer systems cannot account for the necessary human activities.

6 Related work

Purpose-based access control [3,4] proposes an access control mechanism for databases where each data item has an associated intended purpose. To access the item, a user must state his access purpose. Both kinds of purposes are

arranged in hierarchies, and a notion of compatibility of purpose is defined. The access control mechanism itself is essentially role-based access control (RBAC). Similar ideas form the basis of a formal language for specifying purpose-based access control policies in [30] and for deriving formal invariants and proof obligations from a formal specification of such policies. The idea of matching a stated with an intended purpose is also pursued in [24]. Finally, similar to our work, [26] proposes identifying purposes and business processes. The authors use knowledge of the current task within a process for access control decisions. In contrast to all of these works, our focus is not on designing access control mechanisms, but rather audit and compliance.

Privacy-aware role based access control [20] proposes extending RBAC with a hierarchical notion of purpose to model privacy policies, emphasizing conflict detection in the resulting formal models. Similarly, *Purpose-Aware Role-Based Access Control* [18] extends RBAC with an explicit notion of purpose, in part to alleviate technical difficulties in expressing privacy policies in RBAC. In [7], the authors propose an access control mechanism supporting conditional purpose, allowed purpose, and prohibited purpose, each of which is defined through dynamic roles, with the actual intended purpose computed dynamically. The paper emphasizes balancing privacy concerns against data mining opportunities. In [17], the authors use information-flow labels to specify and enforce purpose-based access control policies. They argue that information-flow diagrams are well-suited to express and reason about purpose-based privacy policies. This is very much akin to the use of process collections in the present paper.

All the above works associate data with purpose, an idea we have taken up in our notion of process collections. However, the previous works proceed to give methods for access control under various circumstances. In contrast, our work is concerned with the questions: what are the appropriate purposes in the first place, and is (data) access required for these purposes? Moreover, these other methods are invariably *automated*, whereas we have emphasized that the GDPR also requires human activities to ensure compliance (see Section 5.4).

Finally, [15] gives a semantic model of purposes themselves, stipulating that "the purpose of an action is determined by its situation among other interrelated actions." The authors model actions and their relations in an *action graph*, and develop a modal logic and model-checking algorithm for verifying purpose-based policies. This work is akin to the present one in that it addresses the question "how do we find purposes?" However, the present paper does not attempt a semantic analysis of purpose, and leverages instead the observation that practitioners have already defined purposes via business process modelling.

7 Discussion and Conclusion

We investigated the GDPR and we showed how a mechanism for relating *purpose* to implementation artefacts is necessary to demonstrate compliance. To remedy the problem that purpose is usually not represented explicitly in implementations of computer systems, we put forward the idea of *identifying* a business process

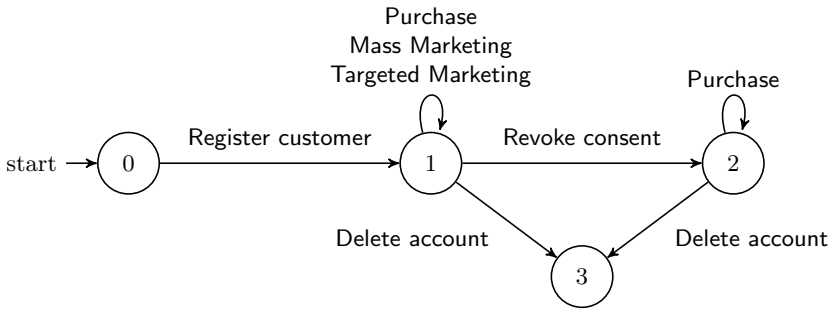

Fig. 6. Lifecycle model for the on-line retailer processes.

and a purpose. We proposed a methodology where GDPR compliance is decomposed into the compliance of an implementation to an interprocess dataflow model, of the dataflow model to a privacy policy, and of these latter two models to the GDPR. We demonstrated that given a model of interprocess dataflow, we may verify compliance of certain aspects of the GDPR algorithmically. In particular, we can generate privacy policies automatically from the model and detect violations of data minimisation. Finally, we explained why GDPR compliance cannot be entirely automated and the role of humans in enforcement.

Discussion. We return now to the question of data deletion. Recall from Section 5.3 that data should be deleted once the purposes for which it is used can no longer apply, that is, when the corresponding processes can no longer be started. Providing such a fine-grained account of deletion requires modelling when processes start using a *process lifecycle model*. We provide an illustrative example here, leaving the full development of this idea to future work.

Our example is the model in Figure 6, which models the lifecycle of the processes in our running example. We have added processes here for deleting an account and revoking consent. This model is a finite state machine where states distinguish what processes can be started and transitions are processes started. Some processes, such as Purchase, do not change the current state. Other processes lead to state changes, such as the new processes Revoke consent and Delete account. In a given state, the set of processes that may yet start is the set of reachable transitions. For example, in state 0, it is all the processes; in state 1 it is all but Register customer; in state 2 it is only Purchase and Delete account; and in state 3, we may not start any processes. From this information, we can compute what data we must delete. For example, in the transition from state 1 to 2 we lose the ability to start Targeted marketing, which is the only purpose for storing the ⟨profile⟩ data. It follows that immediately after this transition, we must delete that data.

Future work. Another important area for future work concerns data transfers to third parties. The GDPR has precise rules about who may transfer data to

other parties, when these transfers can occur, and under what circumstances other parties can or must delete, produce, or store data. Naturally, this opens up questions about audits and compliance similar to the ones addressed in this paper. Moreover, enforcement in this setting is closely related to research on distributed usage control [27,28] and on executable process models [6,29]. Other relevant future work includes how to distinguish between personally identifiable information and other information, and handling systems that allow users to consent to some, but not all, purposes.

References

1. Wil M. P. van der Aalst and Maja Pesic. DecSerFlow: Towards a Truly Declarative Service Flow Language. In *Proceedings of Web Services and Formal Methods (WS-FM 2006)*, volume 4184 of *LNCS*, pages 1–23. Springer, 2006.
2. BPMN Technical Committee. Business Process Model and Notation (BPMN). Technical Report formal/2011-01-03, Object Management Group, January 2011. Version 2.0.
3. Ji-Won Byun, Elisa Bertino, and Ninghui Li. Purpose based access control of complex data for privacy protection. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 102–110. ACM, 2005.
4. Ji-Won Byun and Ninghui Li. Purpose based access control for privacy protection in relational database systems. *The VLDB Journal*, 17(4):603–619, 2008.
5. Thomas H. Davenport. *Process innovation: reengineering work through information technology*. Harvard Business Press, 1993.
6. Søren Debois, Thomas T. Hildebrandt, and Tijs Slaats. Concurrency and Asynchrony in Declarative Workflows. In *Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings*, pages 72–89, 2015.
7. Md. Enamul Kabir, Hua Wang, and Elisa Bertino. A conditional purpose-based access control model with dynamic roles. *Expert Systems with Applications*, 38(3):1482–1489, March 2011.
8. Facebook Data Policy. `https://www.facebook.com/policy.php`. Accessed 2017-08-09.
9. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, L119:1–88, April 2016.
10. Google Privacy Policy. `https://www.google.com/policies/privacy/`. Accessed 2017-08-09.
11. David Harel and Michal Politi. *Modeling Reactive Systems with Statecharts: The Statemate Approach*. McGraw-Hill, Inc., New York, NY, USA, 1st edition, 1998.
12. Thomas Hildebrandt and Raghava Rao Mukkamala. Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs. In *Post-proceedings of PLACES 2010*, volume 69 of *EPTCS*, pages 59–73, 2010.
13. Richard Hull, Elio Damaggio, Fabiana Fournier, Manmohan Gupta, Fenno Terry Heath, III, Stacy Hobson, Mark Linehan, Sridhar Maradugu, Anil Nigam, Piyawadee Sukaviriya, and Roman Vaculin. Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles. In *WS-FM 2010*, pages 1–24, Berlin, Heidelberg, 2011. Springer-Verlag.

14. IBM Privacy Policy. `https://www.ibm.com/privacy/us/en/`. Accessed 2017-08-09.
15. Mohammad Jafari, Philip W.L. Fong, Reihaneh Safavi-Naini, Ken Barker, and Nicholas Paul Sheppard. Towards Defining Semantic Foundations for Purpose-based Privacy Policies. In *Proceedings of the First ACM Conference on Data and Application Security and Privacy*, CODASPY '11, pages 213–224, New York, NY, USA, 2011. ACM.
16. Jens Knoop, Oliver Rüthing, and Bernhard Steffen. Towards a tool kit for the automatic generation of interprocedural data flow analyses. *J. Prog. Lang.*, 4(4):211–246, 1996.
17. N. V. N. Kumar and R. K. Shyamasundar. Realizing Purpose-Based Privacy Policies Succinctly via Information-Flow Labels. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing*, pages 753–760, December 2014.
18. Amirreza Masoumzadeh and James B. D. Joshi. PuRBAC: Purpose-Aware Role-Based Access Control. In *On the Move to Meaningful Internet Systems: OTM 2008*, Lecture Notes in Computer Science, pages 1104–1121. Springer, Berlin, Heidelberg, November 2008.
19. Raghava Rao Mukkamala. *A Formal Model For Declarative Workflows: Dynamic Condition Response Graphs*. PhD thesis, IT University of Copenhagen, 2012.
20. Qun Ni, Elisa Bertino, Jorge Lobo, Carolyn Brodie, Clare-Marie Karat, John Karat, and Alberto Trombeta. Privacy-aware Role-based Access Control. *ACM Trans. Inf. Syst. Secur.*, 13(3):24:1–24:31, July 2010.
21. Object Management Group. Unified Modeling Language: Superstructure. Technical Report formal/05-07-04, Object Management Group, August 2005. Version 2.0.
22. Object Management Group. Unified Modeling Language: Infrastructure. Technical Report formal/05-07-05, Object Management Group, March 2006. Version 2.0.
23. Object Management Group. Case Management Model and Notation. Technical Report formal/2014-05-05, Object Management Group, May 2014. Version 1.0.
24. H. Peng, J. Gu, and X. Ye. Dynamic Purpose-Based Access Control. In *2008 IEEE International Symposium on Parallel and Distributed Processing with Applications*, pages 695–700, December 2008.
25. Maja Pesic, Helen Schonenberg, and Wil M. P. van der Aalst. DECLARE: Full Support for Loosely-Structured Processes. In *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference*, pages 287–300. IEEE, 2007.
26. Milan Petković, Davide Prandi, and Nicola Zannone. Purpose Control: Did You Process the Data for the Intended Purpose? In *Secure Data Management*, Lecture Notes in Computer Science, pages 145–168. Springer, Berlin, Heidelberg, September 2011.
27. Alexander Pretschner, Manuel Hilty, and David Basin. Distributed usage control. *Commun. ACM*, 49(9):39–44, September 2006.
28. Alexander Pretschner, Manuel Hilty, David Basin, Christian Schaefer, and Thomas Walter. Mechanisms for usage control. In *ASIACCS '08: Proceedings of the 2008 ACM symposium on Information, computer and communications security*, pages 240–244. ACM, 2008.
29. Ingo Weber. Untrusted Business Process Monitoring and Execution Using Blockchain. pages 329–347, 2016.
30. Naikuo Yang, Howard Barringer, and Ning Zhang. A Purpose-Based Access Control Model. In *Third International Symposium on Information Assurance and Security*, pages 143–148, August 2007.