Integrated Privacy Modeling and Validation for Business Process Models

Sören Witt
Christian-Albrechts-University of Kiel, Germany
swi@informatik.uni-kiel.de

Sven Feja
Christian-Albrechts-University of Kiel, Germany
svfe@informatik.uni-kiel.de

Andreas Speck
Christian-Albrechts-University of Kiel, Germany
aspe@informatik.uni-kiel.de

Christian Prietz
Unabhängiges Landeszentrum für Datenschutz (ULD)
Kiel, Germany
uld78@datenschutzzentrum.de

ABSTRACT
Privacy is an important issue, inducing a strong interest in correct holistic treatment of data in processes and systems of enterprises. Beside avoiding infringements, trust in the correct holistic treatment of data increases the overall trust in an enterprise, gaining a competitive advantage. More and more, enterprises utilize business process models (BPMs) to specify, document or optimize (existing) processes and systems. Hence, such BPMs also offer the chance to analyze and validate specifications or existing systems with respect to privacy requirements.

In this contribution we present the concept of Integrated Privacy Modeling and Validation and its implementation in our BPM validation and verification framework Business Application Modeler (BAM). BAM enables the automatic validation of BPMs against graphically specified, formal privacy requirements, which can reduce error-prone and expensive manual checking. Furthermore, BAM provides the MultiView concept which allows the definition of concentrated and reduced, privacy related views on BPMs.

General Terms
Legal Aspects, Verification

Keywords
privacy, compliance, business process modeling, verification, validation, graphical validation rules, temporal logic, views

1. INTRODUCTION
Privacy has become an important and at the same time infamous issue for enterprises and public authorities as well as for their customers. The German electronic health card project, or different incidents like the transfer of personal data to third parties by a payment service [8], the spying scandal of Deutsch Bahn AG [1] have unveiled that the handling of personal data is often too careless or questionable.

In order to strengthen the trust in the enterprises or public authorities and their processes and systems a correct holistic treatment of data is required. This affects much more than the—nevertheless important—aspect of data security. Use of data beyond declared permissions, intransparency or the linking of explicitly disjunctive data are examples for (potentially) negative treatment of data. A primary goal of organizational practices of enterprises and public authorities should be the provision of services with compliant processes. The compliance of processes with respect to privacy could be supplemented by certificates like the EuroPriSe certificate of the ULD. To establish a holistic treatment of data organizational practices as well as technical procedures of the organizations are affected. Although, the integration of the awareness of privacy into organizational practices is an important factor, the focus of this contribution concentrates on issues like the integration of privacy aspects into the documentation and specification of processes.

An importance gaining documentation and specification technique are BPMs. Basically, these graphical models are used to describe the flow of activities in processes. However, BPMs are a central information storage. Therefore, all relevant information (e.g. privacy aspects) should be included or at least linked into the models. Hence, practices and techniques are needed to integrate these information into the processes and systems. In the context of our contribution this deserves the integration of privacy aspects into business process models. On the one hand this requires methods to integrate requirements for data handling into the modeling of BPMs. On the other hand the manual checking of the increasingly complex BPMs is time-consuming, error-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. PAIS 2012, March 30, 2012, Berlin, Germany. Copyright 2012 ACM 978-1-4503-1143-4/12/03 …$10.00.

1Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein; federal state data protection authority.
prone and uneconomical and it should be least supplemented by automatic checking techniques [16].

Basically, integration of privacy aspects into process models can be realized by annotating the model and its elements. However, the extensive use of annotations can make the process models at least graphically confusing. A well-known solution to reduce the graphical complexity of these models are view concepts. In this sense views can be seen as “vehicles for separation of concerns” [11] and therefore allow to focus on specific aspects of the process model. Different view approaches have been developed. An overview is given in [11]. An early version of the view concept we propose is presented in [4].

The trust in the correctness of the BPMs is essential. Achieving this is challenging with respect to a correct holistic treatment of data as well as to various other functional domains. In order to avoid time-consuming and error-prone manual checking of complex BPMs, it is desirable to automatize checking where possible. Automatic checking requires the formal specification of requirements, which mostly has to be conducted as a mathematical appearing textual statement. Commonly such formulas are not well accepted by the majority of those who work with BPMs. In order to raise comprehensibility for all stakeholders, we utilize graphical validation rules [3] for the expression of privacy requirements on the level of abstraction of BPMs.

In this contribution we present the approach of Integrated Privacy Modeling and Validation (IPMV). The IPMV integrates modeling views—called MultiView—for BPMs and automated checking techniques into the classical modeling workflow. Since the notation for graphical validation rules is not restricted to express privacy requirements, it integrates well with checking of requirements in terms of other functional domains. The realization of IPMV is presented with the BAM [5]. This is a business process modeling tool with integrated validation and verification capabilities. Furthermore, we present an example business process including its validation against a comprehensive set of privacy validation rules.

2. BACKGROUND

Various techniques, frameworks, tools or notations exist to describe the processes in an organization. No matter how the processes are described, the descriptions have to fulfill two major requirements: First, it has to contain a graphical notation and second it has to contain explicit specifications of requirements for this process.

Before analyzing the advantages of a holistic approach for process modeling and description we will consider the challenges that are faced by organizations, companies, and public authorities in terms of data processing.

Compliance with legal requirements – also with regard to data protection – is a basic objective of organizations as well as data subjects. In times of global communication, strengthened consumer protection, and web 2.0 it becomes more and more important for organizations to act in a compliant way. Possible errors and omissions can cost a large amount of money or cause serious damage to reputation. Due to this and the fact that the complexity of business processes and data processing increases, it is necessary to embed mechanism for a law-abiding behavior in every business process and data processing step, to raise the awareness for compliance and to check it on a regular basis.

However, for different reasons in practice many non-compliant process models and implementations exist. These reasons range from efficiency and cost pressure, through a lack of awareness to unclear or contradictory legal situations.

2.1 Monitoring and Controlling

In this context monitoring and controlling (M&C) means to document, analyze, and adjust the (existing) business processes. Obviously, this can already be done in early stages of software projects. There are different methods for such a M&C within an organization, ranging from checks within the organizational unit through internal audits to evaluations or certifications performed by external parties.

A common foundation of any M&C method is a precise definition of the Target of Evaluation (ToE). This is an essential part and the basis of the evaluation itself. Therefore it is important to clearly point out what the ToE is, what it consists of and what it is not. Such a detailed description is for example also a prerequisite for a EuroPriSe (European Privacy Seal) certification project. The daily work as a EuroPriSe certification body2 shows the relevance and the advantages of a clearly defined ToE. The European Privacy Seal certifies compliance of IT products or IT-based services with EU data protection regulations. Often IT-based products and services are described by a number of meshed BPMs, which are part of the ToE. From the certification body’s point of view graphical process modeling notations like event-driven process chains (EPC), business process modeling notation (BPMN) or unified modeling language activities are a preferred kind of process descriptions. The approach presented in this contribution focuses on modeling and validation of such BPMs in order to support M&C methods.

Assumed, that the BPMs are specified on a suitable level of detail, they increase the clarity by visualizing complex relations in a simplified manner. Annotations allow to enrich the models with further, detailed information, increasing their expressiveness and making BPMs a central information storage as mentioned before. This holds for information related to privacy and data handling as well, as for further information from various other domains, e.g. economy, service oriented architecture or enterprise specific demands. The presented approach utilizes the MultiView concept, to handle the raising complexity of information enrichment and to provide the relevant—and only the relevant—information, e.g. needed to support a EuroPriSe certification.

BPMs combined with Annotations express requirements, e.g. about the use of encryption or restrictions regarding data handling. Nevertheless, they also have to fulfill several requirements. In order to comply to these requirements, measures need to be considered or implemented. Note that these includes organizational as well as technical measures. Hence, these requirements must often be seen in the context of the dynamic behavior of the BPMs. Those requirements, which can be expressed formally are candidates for the automated checking of BPMs against these requirements. The presented approach addresses this issue, by providing an automated checking mechanism and a notation for the graphical specification of requirements on the same level of abstraction like the BPMs, considering their dynamic behavior.

2Christian Prietz is a member of the EuroPriSe certification body that is organized as part of the ULD.


2.2 Protection Goals

From a security point of view protection goals are a well known construct to describe demands on IT systems. They offer the possibility to describe these demands in an abstract way and thus they are guidelines how to develop, implement and maintain such systems. Nonetheless, the classical protection goals confidentiality, availability and integrity consider the security requirements only. Since data processing becomes more and more complex, it is not enough to focus on security. Further important concerns are data protection and rights of the data subject. In this regard the classical protection goals are not sufficient to consider all relevant aspects of data processing. Due to this [14] developed three new protection goals which designed to achieve compliance in the area of data protection:

Transparency in data processing means “that the collection and processing operations of data and its use can be planned, reproduced, checked and evaluated with reasonable efforts” [13].

Unlinkability means “to operationalize purpose bindingness and purpose separation.” Unlinkability is intended to ensure that collected data are not used or processed for any other than the original purpose, or only with excessively high efforts [13].

Intervenability is meant as the operationalization of data subject rights and “the ability of information processing entities respective operators of systems to demonstrate verifiable that they actually have steering control over their systems” [13].

The next step is to derive a set of concrete requirements and measures in order to reach these goals. A possible source for these requirements and measures could be the EuroPriSe criteria catalog. It consist of four different sets of criteria dealing with fundamental issues of personal data processing, like processing purposes or transparencies. Furthermore, the legitimacy of data processing is addressed, e.g. the legal basis or the internal disclosure of personal data. The third set considers technical and organizational measures. These are for example access rights, documentation or encryption. The last set deals with the rights of data subjects, e.g. affecting the right of blocking or access. As part of these sets the catalog defines a variety of questions that can be used to derive the adequate measures.

The transparency requirement for example addresses issues like making data processing transparent to the user. This includes the availability of a data flow diagram, information about the data location, ways of data transmission, a privacy statement or issues dealing with the description of the product or service.

3. INTEGRATED PRIVACY MODELING

A working privacy management should be based on defined and secured (business) processes [2]. But in many practical cases no explicit process definitions or specifications exist. Therefore, the knowledge of organizations about their processes is often limited. Documented and hence defined processes can provide economical advantages and allow the consideration of privacy aspects in a holistic view of the process models. Next to the documentary character of the described processes these models might also be used for general software specification or in a model-driven software development (MDSD [15]).

However, the realization of privacy is seen as hindering and costly especially for existing processes and systems [4]. In order to reduce the extra effort for the handling of privacy aspects the Integrated Privacy Modeling and Validation (IPMV) offers an approach to integrate privacy requirements into the modeling workflow of (existing) business process models. The modeling is supported by automatic checking techniques, as described in section 4.

3.1 Approach

In the business process modeling context, the awareness of privacy and the modeling capabilities for privacy requirements respectively is not very prevailing. Therefore, the primary goal of IPMV is the seamless integration of privacy aspects into (existing) documentations like BPMs in order to manifest the awareness for correct handling of personal data. As mentioned above, there exist various notations and tools to specify business processes. Hence, the IPMV is independent from specific notations or modeling tools.

The second goal of IPMV is to improve the trust in the process models by providing automatic checking techniques to ensure the correctness of the modeled processes. This requires a formal machine-readable representation of privacy aspects that are integrated into the BPMs—the modeling capabilities.

3.2 Modeling Capabilities

In general, the modeling capabilities of IPMV are based on the concept of MultiViews [6]. Basically, the MultiView concept delivers adaptable views on process models which is comparable to a filter for the model content of process models. MultiViews can be defined on user as well as on an enterprise wide level.

The adaption within MultiViews can be done by the four instruments hide, add, change and restrict access. While the latter allows to restrict the access on specific model content like control flow elements the other three instruments allow to adapt the appearance and representation of a BPM and its content. This includes to annotate model elements with domain specific aspects like privacy issues as well as to hide irrelevant model contents like annotations or specific model elements. The instrument change allows to change the appearance of the model content including the model itself (e.g. background color).

In the MultiView IPMV annotations are primarily used to specify special privacy aspects for model elements. For example, a data cluster, which contains personal data, is annotated with the attribute pD. Figure 1 shows this annotation in a part of the example process (cf. section 3.4). Other examples for annotations in the MultiView IPMV are attributes to declare the type of function (e.g. delete) or the persistence attribute for (application) systems.

As shown on the left hand side of figure 1, next to annotations, the graphical representation of BPMs can be adapted by MultiViews. On the right hand side the same process is shown in the MultiView Web Service Security. This way a MultiView provides a specific view on the BPM for different domain experts.

The definition of the means of the MultiViews and the MultiView IPMV respectively is independent of process modeling notation. Hence, the defined annotations of IPMV are
not solely usable for EPCs but although e.g. for BPMN.

3.3 Realization of IPMV in BAM

Generally, the intention of BAM is to provide an integrated solution to check the correctness of process models and BPMs respectively. In order to facilitate these checks, BAM has to provide on the one hand modeling capabilities and on the other hand validation and verification mechanisms.

The modeling capabilities of BAM are based on a generic meta-meta-model, which allows the definition of process models that are based on nodes and edges. Next to the pure modeling of process models BAM offers the MultiView concept which has been described before. Independent of a meta model, BAM realizes annotation of BPMs generically by letting the modeler place arbitrary attributes (key-value pairs) on any model element. Furthermore meta models may be extended by additional elements.

The validation and verification mechanism of BAM basis on the Temporal Logics Visualization Framework [3] which delivers the conceptual basis for the graphical definition of formal requirements for process models. Furthermore, the TLVF offers a basic architecture for modeling and validation tools which intends to realize the TLVF concept. Hence, the TLVF does not especially focus on privacy requirements its realization in BAM is not bounded to privacy issues, too.

3.4 Example Process

In this section we introduce an example process to demonstrate the presented approach. This process is depicted in figure 2, modeled in the EPC notation. It shows a possible description of an once-only order in an online shop. An once-only order in general is intended to narrow the business relationship between shop and customer to a minimum. In terms of privacy this also means to request a minimum of data and to use these only for the purpose of order handling.

The process starts when the customer enters the payment site after assembling it’s shopping cart. The import of the shopping cart data into the process is indicated by the data cluster symbol, attached to the start event. The first action of the customer is to choose the type of order. Here we focus on the once-only order branch of the process behind the first XOR split in the control flow.

Now the customer enters her/his data. The created data are depicted by the cluster symbol customer data. The directed, outgoing, dashed edge denotes, that these data are written, respectively created. This cluster carries the attribute pd, indicating that these data are considered to be personal data. Furthermore there is a box attached to the cluster by a non directed edge. It symbolizes a condition, describing the legal basis on which these data can be used. This might be for example a permission given by the user.

In the next step, the customer selects the preferred payment method. In this example shop it is either cash on delivery or the whole order process may be canceled. Otherwise, if the credit card has been charged successfully the process jumps back to the point, where the customer selects the payment method. That means that the process contains a loop and the customer may try again to charge the credit card, or choose cash on delivery or the whole order process may be canceled. Otherwise, if the credit card has been charged successfully the received amount is booked by the shop system, again using the ERP and customer data.

If the order has not been canceled the payment is processed completely and the invoice can be created. Here the customer data and ERP are involved. In the next step, the stock is refreshed in the ERP on basis of the shopping cart data. This is followed by the preparation of the shipping. After shipping the customer has the possibility to return the received products. Here the process branches in two paths, one in case of no returns, the other denotes the handling of returns. Finally the customer data are deleted by
the function delete customer data. This function is annotated with the "delete" attribute. It indicates, that this function deletes all its input data.

4. SPECIFICATION AND VALIDATION

BAM allows the automatic validation of process models against graphically specified formal requirements. In this section we will give an introduction into the notation for such graphical validation rules. This is followed by a presentation of privacy rule examples. Apart from showing how to check a selection of important privacy issues, these examples point out some issues regarding the specification of rules like being aware of loops or the modularization of larger process models. Furthermore it will be explained why the example process is valid against these rules or not.

4.1 Graphical Validation Rules

We will give a brief introduction into the Graphical Computation Tree Logic (G-CTL), which is described more detailed in [5, 3]. The G-CTL notation is based on the Computation Tree Logic (CTL). The CTL extends classic boolean logic with temporal operators, allowing to specify assertions about the temporal behavior of a system. Therefore CTL and G-CTL focus on the specification of the dynamic behavior of modeled systems.

The G-CTL notation provides a graphical representation...
of the textual CTL operators\(^3\) as depicted in figure 3. On basis of this, G-CTL offers two core capabilities:

1. Specification of atomic expressions, using patterns of process model elements. It allows to express formal specifications on the level of abstraction of the business process models.

2. Formulation of reusable, generalized G-CTL rules by using placeholders in the patterns. This allows the application of rules on other process models and gains a larger coverage of single rules.

Figure 4 shows a simple example of a G-CTL rule, requiring the eventual deletion of data. This already addresses an important issue in data protection and will be revisited again later. The rule has two patterns of process element enclosed in the solid rectangles with rounded corners. The dashed rectangles can be read as brackets. An unary operator may be docked directly to a bracket, which then encloses the operand.

As denoted above: The patterns are the atomic expressions in G-CTL rules. They evaluate to true, if the process is in a state, in which a constellation of process elements matches the pattern. The first pattern in the upper part of the example rule matches in every state of the process in which a function occurs, that creates or writes to (direction of dashed arrow) a data cluster, which is annotated with the attribute personal data (pd).

The second pattern matches on the occurrence of a function, which is annotated with the delete attribute and which takes a cluster with pd attribute as input. Here, the delete attribute means that this function deletes all its input data. Moreover, there is a further constraint regarding these two patterns: The \(i = 1\) expression inside of the clusters requires the identity of these data clusters. So the second pattern matches all functions, that delete exactly the data, created by the function in the first pattern.

The outer operator in this example rule is the All Globally (AG) operator (filled square). It is satisfied if its operand evaluates to true in every state of the process. Hence the boolean implication operator inside the bracket must be satisfied. The antecedent of the implication is true if the first pattern matches. The apodosis points on the All Future (AF) operator (filled rhombus). This AF operation evaluates to true, if the process eventually reaches a state starting from \(s\) (immediately or later), in which the second pattern occurs.

\(^3\)The semantic of G-CTL and CTL operators is identical. We refer to literature for detailed information about CTL.

Hence the rule in figure 4 requires, that in any state of the process holds: If in some state a function creates personal data then (from this particular state on) the process must always reach a state, in which a function deletes these data. To illustrate the effect of the E and A quantifier in CTL, figure 5 shows the rule from figure 4, but the AF operator is replaced by an EF operator. In this case the process must not always reach the delete function. There must only exist at least one course in the process, reaching the delete function. Hence this rules expresses a weaker requirement, since deletion must not necessarily take place. It must only be possible.

Additionally should be mentioned, that the G-CTL notation also allows to use concrete element names in patterns as well instead of the placeholders.

4.2 Specification of Privacy Rules

In this section we are going to demonstrate on the basis of examples how to use G-CTL rules for the specification of privacy requirements.

4.2.1 Deletion of Data

The previous section showed a first example to explain the G-CTL notation. As mentioned before it also addresses the important issue of reliable data deletion, which affects all protection goals. In this section the application of the presented example rule from figure 4 against the example process in figure 2 will be shown. Actually, this and various other rules can not be specified without taking into account some frame conditions. These regard for example the expressiveness of the modeling language or modeling style and conventions. To point out some of these issues, a more sophisticated deletion rule will be presented.

To check the example process against the rule in figure 4, the patterns need to be matched firstly. The first pattern matches in two places in the process. One is the customer enters data function with the customer data cluster, the other is the function customer enters credit card data including the cluster credit card data. The only possible match for the second pattern is the function delete customer data, which takes the cluster customer data as input. According to the identity constraint of the data cluster, it is clear that the process cannot be valid with respect to the deletion of the credit card data, because a matching delete function does not exist.

Due to two reasons, the process is also invalid with respect to the deletion of the customer data. The first reason is the possible early termination, if the user cancels the order while
choosing the payment method. It could for example be fixed by connecting the event **order canceled** with the XOR join ahead of the delete function. The second reason is, that the process may get stuck in the loop if credit card payment is chosen repeatedly and the charging always fails.

**Loop Awareness in Rule Design.**

The process model contains no machine readable information, which limits the number of loop runs or guarantees that the loop will not run infinitely often. Hence staying in the loop forever is a kind of a stable end state of the process.

The more sophisticated deletion rule, depicted in figure 6, takes loops into account. Here the occurrence of a function, which creates personal data implies, that always a delete function must be reachable until this delete function actually occurs. This rule uses the **Always Weak Until operator**[^1], which requires that a certain condition must hold until a release condition occurs. In contrast to the "strong" until operator the release condition does not have to become true eventually. So here the EF operator enforces the existence of a path to the delete function until the data are deleted by such a function. Since the loop can terminate, the loop doesn’t render the process invalid against this deletion rule. Specifying the rule this way avoids to make fairness assumptions about the loop.

**Data Flow over Multiple Models.**

Another issue might occur, when the overall process is modularized into multiple process models, which is highly probable for large processes. Most likely there exists no consistent overall model, aggregating all the single process models and connecting their workflows.

Imagine to split the example process in figure 2 at the **payment processed** event into two separate process models. The **payment processed** event would be an end event of the left part and the start event of the right part. After this split the deleting and creating functions are not in the same model. Hence the model, which creates the data would be invalid with respect to the presented deletion rule.

Ensuring the deletion of data, is a matter of controlling the data flow. So it might help to introduce a modeling convention, to provide analyzable information about the flow of personal data. Here this convention required that personal data must be either created or imported and either deleted or exported inside the same process. The import and export is expressed by attaching the relevant cluster to the start and end events. This import and export would appear similar to the shopping cart, which is attached to the start event of the example process. This allows to see easily which personal data are required and emitted by a process.

Figure 7 shows a deletion rule, which takes this modeling convention into account. The logic is identical to the rule in figure 6. But now data may be introduced not only by creating them, they may also be imported and instead of deletion they may be exported. Note that this rule also requires, that exported data are annotated with a destination attribute. The value of this attribute might for example be a link to a process model, which uses these data next. Hence this rule enforces the user to comply with the import export convention and to provide at least further information of the data flow and how it is related to other process models. Such information may be used for further (automated) consistency checks.

Hence, when applying this rule, the missing deletion of credit card data may also be fixed by exporting them. But here the modeler would be forced to name a destination, which should raise doubt. If not, a reviewing privacy expert will most probably detect this issue, since he will examine imports and exports of a process.

**Complementary Rules.**

Enforcing certain aspects like the deletion of data, might nevertheless come along with related modeling mistakes. For example data could be deleted too early and a concurrent part of the process might still try to read them. It is not unlikely, that a modeler will miss this issue.

Hence it is useful to provide complementary rules. An example is shown in figure 8. The rule forbids to read or export data after they have been deleted. This prohibition is only released if a state occurs, that recreates these data, which

[^1]: Can be expressed with basic operators: $A[xW y] = \neg E[\neg y U (\neg x \land \neg y)]$
Figure 8: Data should not be read after deletion until they are recreated.

Figure 9: Avoid creation or import of data, which are never read.

is expressed by the until statement. The example process is valid in terms of this rule. Although this rule does not directly affect data protection issues (except availability), it helps to achieve a correct holistic treatment of data.

4.2.2 Data Avoidance and Usage Conditions.

Avoiding the unnecessary collection or creation of data, is a major principle to achieve strong data protection. If data are actually needed, certain conditions must be fulfilled, e.g. a permission given by the user or a legal basis. Such conditions should be documented in a process model, so that privacy experts can evaluate these (for example in the context of certification).

The rule in figure 9 shows a very simple rule which requires that all imported or created data will be read in some state of the process. The logic in this rule is quite similar to the deletion rules in figure 6 and 7, making it robust with respect to loops. The example process is again not valid, because in case of canceling, the data were never read. This again could be fixed by linking the event to the XOR-Join ahead of the delete function.

In fact, the rule in figure 9 only works if personal data are correctly introduced by import or creation. Otherwise there

might exist paths in the processes in which the data will not be used. This is one issue, addressed by the rule in figure 10. It requires, that no data may be read unless they have been imported or created before. Additionally, this rule requires the declaration of a condition for the usage of these data. This condition is symbolized by a box attached to the cluster with a non directed edge. The example process is valid with respect to this rule.

Such a condition symbol may be filled freely with text, which is interpreted by a reviewing privacy expert. Nevertheless, there might exist some special cases, which should be expressed explicitly. It is often highly critical if external services are involved in a process, and personal data are required to use them. The rule depicted in figure 11 forbids the occurrence of data in the context of external systems, unless the condition permits the external usage of data explicitly. External systems can be recognized by the attribute external. If personal data occur in the context of such a system, it must be announced by providing a corresponding attribute at the condition symbol, where the data have been created. Since this has not been done in the example process for the credit card data, the use of the external payment service system renders the process invalid.

4.2.3 Awareness of Data Linkability

A further essential concern is to be aware of the possible linking of data that have been unrelated before. Linking these data must be an explicit part of the purpose for which these data have been collected. The data may originate from various sources, also the linking of non personal data with personal data may be critical. It is therefore important to identify activities in processes, in which linking of data is potentially possible. Here the modeler should provide a clear statement whether these data will be linked or not.

Figure 10: All personal data must be imported or created including a condition for usage.

Figure 11: External use must be assigned in condition.
The model checker requires a textual representation of the business process model as Kripke Structure, generated by applying an (operational) EPC semantic, based on [10]. In this transformation each function or event (including its attached objects like clusters or systems), is represented by a variable called sub-state variable, indicating its activity or inactivity. Asynchronous behavior is reflected by allowing sub-states to stay active for an arbitrary long time but eventually terminate (fairness).

Also the rules need to be transformed into a textual representation. Since the CSMV allows CTL specifications, the logic can be transformed straight forward. The patterns are translated into tests for certain sub-state variables. They are determined by matching the patterns against the process model. Patterns, that do not match are translated into false. Due to the identity constraints in some rules, one G-CTL rule may be translated into multiple textual instances, each combining a set of sub-state variable tests, that does not conflict with these constraints.

After this transformation, the validation plug-in launches the CSMV. This either delivers "true", if no rules has been violated or a counter example is presented for each violated rule. This counter example consists of a trace, listing the states that occurred in a course of action the process took. Hence the trace shows how the sub-state variables evolved, which can be mapped back to the functions and events of
the graphical process model. BAM highlights the model elements that occurred on the faulty trace by surrounding them with a box, to make the debugging of the process easier.

Figure 13 shows the visualization of an error trace, as a result of checking the example process\(^5\) against the rule in figure 4. It shows how the process is stuck in the loop, so that the delete function cannot be reached.

5. CONCLUSION

The paper presents the concept of Integrated Privacy Modeling and Validation (IPMV). IPMV aims to provide modeling capabilities for privacy aspects for business process models (BPM). The modeling capabilities comprise the adaptation of the graphical representation of BPMs and the annotation of model elements with privacy aspects. These capabilities are demonstrated on an EPC example process.

In order to strengthen the trust in the modeled processes IPMV provides automatic checking techniques to improve the correctness of BPMs. The checking is based on comprehensible graphical validation rules (G-CTL rules), which can be used to specify privacy requirements as well as requirements from other domains. The specification and application of the graphical validation rules is demonstrated with different privacy example rules like deletion of data, data avoidance, or data linkability. Moreover, the example rules provide an overview how privacy rules could be used, how the rules are interrelated or how the rules complement each other.

IPMV allows next to the modeling and validation capabilities its integration into the modeling workflow of BPMs. This is achieved by reusable privacy validation rules and attribute definitions which can easily be exported from one BPM and applied to another (existing) BPM. Though, it is likely that such a rule system might need to be complemented with specialized rules or existing rules need to be modified in order to fit the demands of a certain domain or organization.

The rules introduced in the paper focus on testing for dynamic behavior of processes. Our future work will moreover include the testing of static properties of BPMs. E.g. some critical points of data linking can also be identified by finding functions in a BPM, which are associated with at least one cluster annotated with pd and at least one different cluster. Such rules are decidable without considering the dynamic behavior of a BPM. Hence, less expensive checking mechanisms (compared to model checking) can be applied. Nevertheless, this static rule wouldn’t help to identify cases which for example the rule in figure 12 would detect.

The presented privacy validation rules represent only an extract of a possible rule system. Therefore, our future work is focused on the development of a comprehensive rule system including dynamic as well as static rules. However, it is not be expected that such a rule system could provide an overall coverage. Hence, it should rather be seen as a best effort support to design better models and allowing to make stronger assertions about the models.

6. REFERENCES


