ANALYSING PRIVACY ASPECTS OF BUSINESS PROCESSES USING AUGMENTED PROCESS MINING RESULTS

Maria Weinreuter[1], Sascha Alpers[1, 2] and Marius Take[1]

[1] FZI Forschungszentrum Informatik
Haid- und Neu-Straße 10-14
D-76131 Karlsruhe
Germany
E-mail: {weinreuter, alpers, take}@fzi.de

[2] Hochschule Heilbronn
Max-Planck-Straße 39
D-74081 Heilbronn
Germany
E-mail: sascha.alpers@hs-heilbronn.de

KEYWORDS
Information Confidentiality and Privacy Petri Nets, Augmented Process Mining Results, Process Mining, Data Privacy, Petri Nets

ABSTRACT

Data protection requirements must be complied with when designing and executing business processes. Since organizations usually execute several processes with a large number of instances, systematic verification of compliance is difficult. Process mining offers a solution to this problem. However, existing process mining methods do not take into account information about the types of personal data processed and their reasons (permissions). For this problem, the augmentation of process mining results offers a promising approach. Process models generated by process mining are subsequently augmented to Information Confidentiality and Privacy Nets. These can then be used as a basis for compliance checking.

1 INTRODUCTION

The European General Data Protection Regulation prohibits the collection, storage and processing of personal or personally identifiable data initially to protect natural persons from the consequences. There are some exceptions to this prohibition (permissive acts), but these are subject to compliance with certain principles. A central principle is purpose limitation. Therefore, it is necessary for organizations that process personal data to know and document the justification (permissible facts) for processing the data and the purposes for which they are allowed to process the data. In the business environment, personal data is processed for various justifications such as contract fulfillment and consent for various purposes such as shipping goods.

In many use cases, companies record individual events during process-based information processing. These so-called event logs can then be used, for example, to generate a model of the underlying business process using process mining methods (process discovery) and to compare the conformance of the executed process (process conformance checking). However, the currently known methods are not designed to consider data protection requirements. However, this consideration could support the compliance with regulatory requirements. Therefore, this paper proposes a supplement to the existing procedures (Section 3). Since the supplement is based on ICPN, they are first introduced in Section 2. In Section 4, a case study is conducted to test, whether the proposed approach is practical, and to determine its advantages and limitations. A corresponding extension to the Process Mining Tool PM4Py (Berti et al. 2019) was implemented in Python and is available under the open-source licence GPLv3 at (Weinreuter 2023). Section 5 draws some conclusions and gives an outlook on future research work.

2 INFORMATION CONFIDENTIALITY AND PRIVACY NETS (ICPN)

Petri nets are a formal modeling language for business processes (Desel and Reisig 1998; van der Aalst and Stahl 2011). The core elements of Petri nets are places (for states), transitions (for activities) and arcs (for the logical ordering of states and activities). In a Petri net system, tokens are added to the places to indicate the current state of the system (called marking). For further details and a formal introduction see (Desel and Reisig 1998) and (van der Aalst and Stahl 2011).

A large number of extensions to Petri nets exist for a wide variety of use cases. One of these extensions are the Information Confidentiality and Privacy Nets (ICPN) introduced in (Alpers 2019). ICPN are a Petri net variant and belong to the class of higher Petri nets. They were developed to express and analyze information confidentiality and data protection aspects in business process models. For this purpose, the tokens are interpreted as information objects. Properties of information objects include:

- confidentiality level and/or access-authorized roles
- personal data: Boolean, that indicates whether a reference or relatability to a natural person exists.
- permitted processing purposes (only if personal data)

For example, a customer (with attributes first name, last name, date of birth, bank details, ...) could be described as an information object. This information object would then have, for example, a confidentiality level of 2 and should only be accessible for the roles "shipping", "accounting" and "marketing". In addition, it is obviously personal data (therefore the value of the boolean personal data is true) and the linked permitted purposes are "shipment of goods", "payment debit" (permission circumstance contract fulfillment) and "e-mail-marketing" (permission circumstance consent).

Furthermore, with each information processing transition (activity) the involved resources and the purposes due to which an information object is needed are linked. To each resource a respective level of trustworthiness and/or a (potentially empty) set of roles is assigned.

Figure 1 illustrates an example. In this, the purpose "shipment of goods" and the resource "warehouse clerk" with trustworthiness level 2 and role "shipping" are assigned to an activity "ship goods". Accordingly, this activity can process the information object described above in the example and may occur - if all other conditions which result from the underlying Petri nets are fulfilled. The activity is enabled, that means it can switch[1], because the purpose is contained in the set of allowed purposes, the trustworthiness level of all involved resources is greater or equal to the confidentiality level of the information and the resource has one of the authorized roles. Another activity "telemarketing" with the purpose "telemarketing", and the assigned resource "call center agent" (trustworthiness level 1 and role "marketing") cannot switch with this information object. This has two independent causes: First, the purpose "telemarketing" is missing in the set of allowed purposes and second, the trustworthiness level of the resource is too low (lower than level 2 which is propagated by the respective information object).

ICPN has been defined such that simulation and analysis methods from traditional Petri nets can be applied or transferred. This allows statements to be made regarding how certain confidentiality or data protection requirements affect business processes and their execution. In addition to a formal notation, ICPN graphical notation supports process modelers and managers in designing and redesigning business processes. In (Alpers 2019) the PriCon4BPM method (Privacy and Confidentiality for Business Process Management) is described and a consistent approach from modeling to decision making (for example, regarding process alternatives) is presented. However, (Alpers 2019) does not yet offer the possibility to automatically transfer log data of executed processes into ICPN for later analysis. Therefore, the corresponding augmentation of process mining results is presented in the following section.

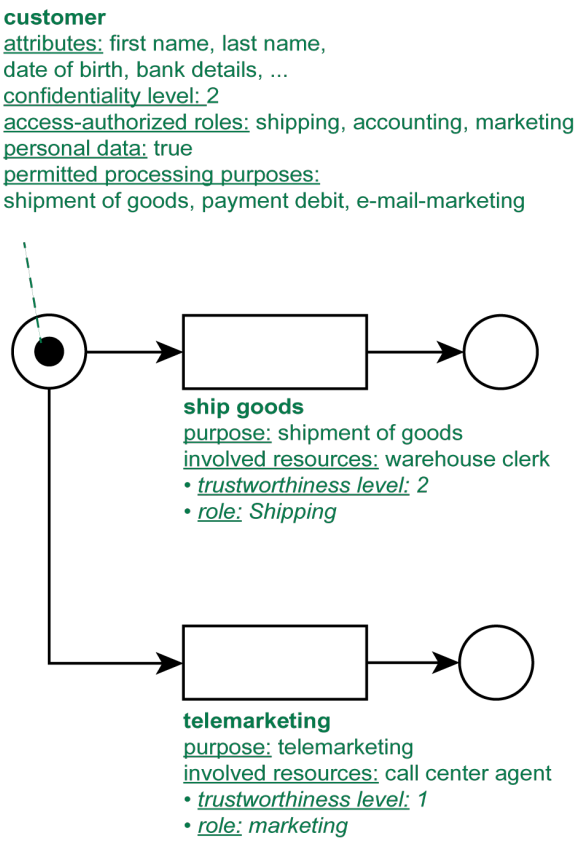

Figure 1: ICPN System Example

3 AUGMENTED PROCESS MINING RESULTS

Process mining allows the acquisition of knowledge from event logs in the form of process models (van der Aalst et al. 2012). In addition to process model discovery, process mining can be used to compare the process actually executed and logged in event logs with a specified process (conformance checking) and/or to improve an existing process (process enhancement) [(van der Aalst et al. 2012) Fig. 3]. Various software tools are available for process mining. Commercial solutions include Celonis and open-source solutions, such as ProM (van Dongen et al. 2005) and PM4Py (Berti et al. 2019). An overview is provided at www.processmining.org.

Augmented process mining is the method of applying existing process mining methods and subsequently enriching their results[2]. In detail, process models generated by proven discovery algorithms are extended with additional information derived from the event log. One important feature of augmented process mining is the unmodified use of the actual mining algorithm; the mining algorithm must not be changed by augmentation. The input for the augmented process mining must, on the one hand, be compatible with the process mining algorithms and, on the other hand, include additional information used for the augmentation. The

[1] Partly instead of the term 'switch' in the literature the terms 'occur' or 'fire' are used.

[2] The term "Augmented Process Mining Results" was chosen in analogy to "Augmented Reality". In this process, the human perception of reality is enriched by additional information. Ideally, this additional information should merge with reality in such a way that the viewer perceives it as a single entity. Information on "Augmented Reality" in (Azuma 1997) and (Azuma et al. 2001).

augmentation uses the result of an already implemented mining algorithm as a parameter that follows only after the actual mining of the flow model. This allows to exchange the mining algorithm later. Figure 2 describes the relationship between the existing procedures (grey) and additions to the augmentation of process mining results (green). The strength of being able to use existing process mining algorithms unchanged and only to enrich their results is at the same time a weakness of the approach because the initially generated process model (e.g. the logical order of the activities) cannot be influenced by the additional information.

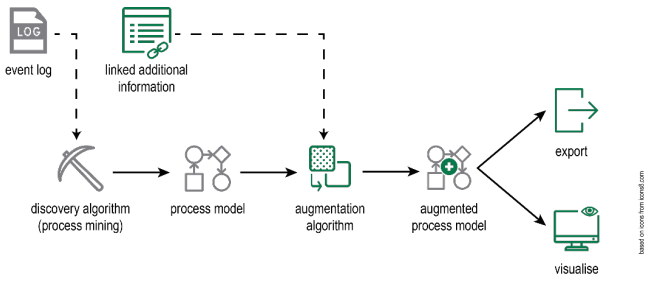

Figure 2: Generate Augmented Process Mining Results

4 CASE STUDY

In this case study, we use the example of purpose limitation as part of ICPN to explain, implement, and evaluate the method of augmented process mining. The example was chosen because, on one hand, it can be presented compactly and, on the other hand, it is relevant for a large number of use cases owing to the European General Data Protection Regulation (EU-GDPR). Purpose limitation is one of the principles of the data processing listed in Article 5 of the EU-GDPR.

4.1 Preliminary Considerations and General Conditions

First the conditions for the concept and implementation of the case study are defined. In particular, the conditions should help to grasp the scope of the case study and to be able to perceive future opportunities for extensions. The selected conditions are as follows:

- Implementation of essential principles for purpose limitation of ICPN, which fulfill only a part of the ICPN definition. Accordingly, the generated model should contain the essential purpose limitation elements of the ICPN; however, it does not have to extract all ICPN information from the event logs. But the idea of generate ICPN from event logs by the augmentation of process mining results can be extended to generate more than the actual implemented parts (future work).
- The typing of information objects of individual process instances to information object types (of the process model) can be realised using a separate mapping function, which is not part of this first case study.
- The software tool PM4Py (Berti et al. 2019) implemented in Python is to be used. One reason is that PM4Py uses the open-source licence GNU General Public Licence Version 3 (GPLv3). The exemplary implementation of the case study was also conducted using GPLv3.
- The case study focuses on process discovery as a process mining method. Process discovery involves the exploration of process models based on event log data. In addition to process discovery, process conformance and enhancement are the main methods used in process mining, which can only be used after a desired process model has been designed or generated (Berti et al. 2019). Therefore, this first case study was limited to discovery.
- The alpha algorithm (van der Aalst et al. 2004) is to be used for the case study.

4.2 Requirements

The specific requirements for augmentation are derived from comparing the process model generated by normal process mining with the augmented process model as expected in the case study after augmentation.
Figure 3 shows a graphical comparison of the selected examples. The result of the non-augmented process miner is shown at the top, and the expected result after augmentation [derived from (Alpers 2019), Fig. 13] is shown at the bottom. The augmented process must contain an information object if a transition consumes (removes from places in the transition's pre-set) or produces (inserts into places in the transition's post-set) an information object. Augmentation must include purposes for which an information object is produced or for which purposes an information object is consumed.

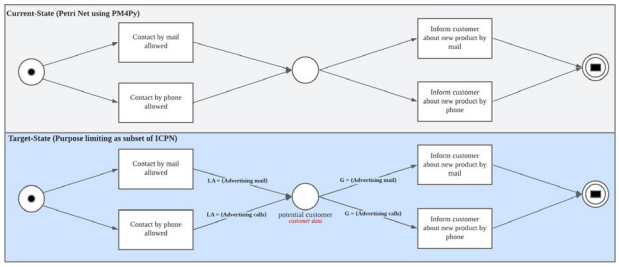

Figure 3: Graphical Comparison of a Petri Net Generated by PM4Py (Current-State) with a Model as we Expect it by the Definition of ICPN (Target-State)

In addition, the requirements for the export must be defined. Figure 3 suggests that process models, which are discussed in a graphical form, such as Petri nets, owing to better readability. Therefore, this should also be provided by augmentation.

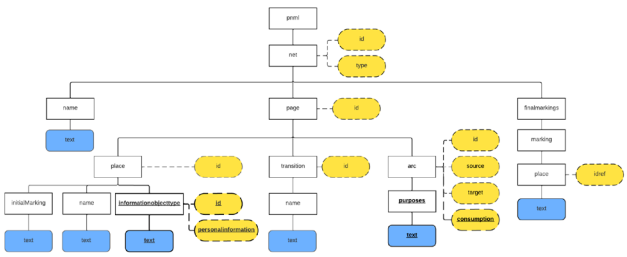

Figure 4: Root Graph of the PNML Standard Extended for Simplified Purpose Limitation as Part of ICPN

The international standard "Petri Net Markup Language" (PNML) can be used for the export of Petri nets (ISO/IEC 2011; Treves et al. 2009). PM4Py also uses this standard. Augmented mining results, such as ICPN, should also be exportable as extended PNML. Therefore, the PNML format is extended to the output format of some purpose limitation aspects in ICPN. This is based on the XML syntax, in which the data can be stored in a hierarchical form (Damiani et al. 2019). The root graph of the PNML extension for exporting some purpose limitation aspects in ICPN from PM4Py is shown in Figure 4. The nodes that must be added to the standard to represent the contents of the simplified purpose limitation (as a part of ICPN) are shown in bold-underlined.

4.3 Input Format

Various formats, such as XES and CSV have been proposed for event logs. For the case study, CSV is used; however, the generalisability is not limited because XES files can be converted to CSV and vice versa. This can be realised using pandas (pandas 2023) and PM4Py. A CSV file could be imported via pandas and saved in a pandas data frame, formatted into a PM4Py data frame, and finally converted into an XES file via PM4Py. For other sources (for example, an event database), necessary converters can be created if they contain the necessary information.

The structure of the input files is more decisive. For the case study, we decided to retain supplementary information in additional files. In addition to the event log as the first CSV file, the produced information objects are stored in a second file, and the consumed information object types in a third file. The files are connected by the composite key (primary or foreign key) case ID and activity (if cycles are to be enabled other than in the case study, the timestamp is also required). Figure 5 illustrates the structure and other attributes of the three input files. The case ID is the identification number of the process instance (data type integer), activity is the name of the event (string), timestamp is the event's occurrence time (datetime/pandas.timestamp), information object type is the type of the information object (for example, PERSON) and not the concrete information object (for example, a person with the first name Elise) (string), permission purpose is the purpose for which a produced information object may be processed (string), and personal information is a truth value that specifies whether an information object contains personal data or data that is related to a person (Boolean).

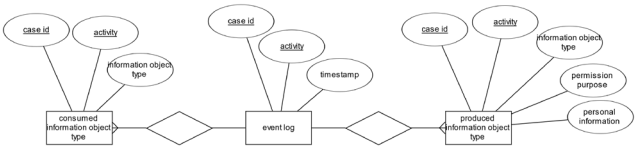

Figure 5: Three Input Files and their Relationships

4.4 Further Conceptual Considerations

Based on the definition of the concept conditions and the required contents of the input and output formats, assumptions and conceptual considerations for the implementation of augmentation can be made. The building blocks by which the Petri net should be augmented are (a) information object types, (b) an indication of whether this is personal data, and (c) the number of purposes for which the respective information object type is produced or consumed. Table 1 outlines the assumptions and conceptual considerations for each building block. We concentrate on the assignment of building blocks to the respective nodes or edges that are to be enriched by this information through augmentation.

Table 1: Assumptions and conceptual considerations

(a) information object types	**Conceptual considerations to the assignment** • As soon as a new information object type is registered (new name registered), an object is created • A transition connected to an output place produces objects of exactly one information object type: Assign information object type to the place • A transition connected to an input place consumes objects of exactly one information object type from that place: Assign information object type to the place • Otherwise: Form the intersection of the consumed information object types of the activities to an input place and assign an information object type to the place, if the intersection exactly contains one element. Consequence: Consumed information objects should be specified for each activity (otherwise the set is empty).
(b) personal data	**Conceptual considerations to the assignment** • Existence of an event noting the production of an information object containing personal or person-related data (entry in column personal_data = TRUE): Information object type contains personal data
(c) purposes	**Assumption** • Purposes are represented as edge attributes **Conceptual considerations to the assignment** • Edge from transition to place: Quantity represents permitted purposes • Edge from job to transition: Quantity represents consumed purposes • Information object type of the job allows unambiguous assignment of purposes. Permitted purposes can either be explicitly specified or implicitly added by consumed purposes

4.5 Implementation of the Case Study

The implementation is shown as a flowchart of the use of augmentation. Augmentation provides the following three functions: the creation ("Create"), visualisation ("View"), and export ("Export") of some purpose limitation aspects of ICPN. After presenting the flowchart, the algorithm for creating the subset of ICPN is described in detail. The implementation was done in Python and is published under the GPLv3 licence as a GitHub project under (Weinreuter 2023).
The augmentation process concerns three essential components that interact with each other. They include the user, augmentation, and PM4Py. The components and their interactions are shown in the flowchart in Figure 6 by different swimlanes. At the abstract level, this process can be divided into four steps. In step 1, the input parameters are to be correctly imported. In step 2, PM4Py generates a Petri net by calling and executing the alpha algorithm. Based on this, augmentation functions can be started. First, purpose limitation as part of ICPN can be generated while preserving the Petri net structure (step 3). Subsequently, the two functions "View" and "Export" can be called and executed by the user in step four. In addition, the source code excerpt in Figure 7 shows how the steps in the flowchart are implemented.

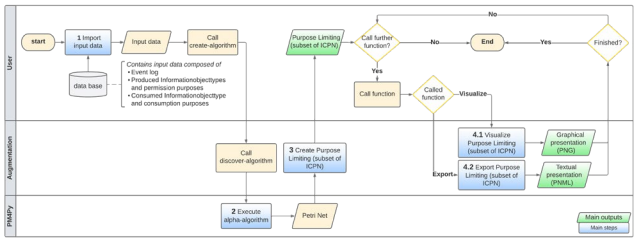

Figure 6: Flowchart for the Implementation of Augmentation

```python
from pandas import read_csv

from create.create_algorithm import create_purpose_limitating_petrinet
from view.view_algorithm import view_purpose_limitating_petrinet
from export.export_algorithm import export_purpose_limitating_petrinet

if __name__ == "__main__":

    #import data
    log = read_csv('C:/Workspace/data/cases_numered_act.csv', sep=";", encoding='latin1', dtype='str')
    IT_consumed = read_csv('C:/Workspace/data/IT_consumed_act.csv', sep=";", encoding='latin1', dtype='str')
    IT_produced = read_csv('C:/Workspace/data/IT_produced_act.csv', sep=";", encoding='latin1', dtype='str')

    #call functions
    pl_net, pl_im, pl_fm = create_purpose_limitating_petrinet(log, IT_consumed, IT_produced)
    view_purpose_limitating_petrinet(pl_net, pl_im, pl_fm)
    export_purpose_limitating_petrinet(pl_net, pl_im, pl_fm, 'C:/Workspace/data/pl_petri_net.pnml')
```

Figure 7: Exemplary Implementation of the Import and Method Calls

Creation algorithm. The creation algorithm is the realization of step 3 and provides a basis for visualization and export of the ICPN-subset. In this algorithm, the Petri net generated by PM4Py is augmented with additional elements of an ICPN-subset. The assumptions and conceptual considerations presented before primarily concern the creation algorithm.

First, in step 2, a Petri net is created by executing a discovery algorithm (here, concrete discover_petri-net-alpha from PM4Py). Subsequently, a new Petri net object is created, to which the set of transitions, as well as the property dictionary of the original Petri net, are passed in the constructor. This is followed by the creation of PlaceIot objects, whose class is inherited from the Place class in PM4Py. The creation occurs via a run through all place objects of the original Petri net. The names of the place objects of the original Petri net are passed directly to the PlaceIot objects.

The corresponding information object types are then assigned to the PlaceIot objects. To do this, two auxiliary functions are first called. In these, information objects are created and stored in a dictionary for each transition in which they are produced or consumed. The subsequent assignment of the information object-type objects to PlaceIot objects is shown in Figure 8.

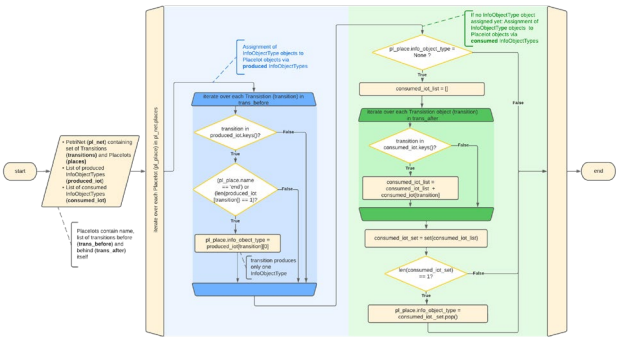

Figure 8: Assignment of Information Object Types to Places

After the information object types have been assigned to the corresponding PlaceIot objects, ArcWithPurposes objects are created. The class ArcWithPurposes inherits from the PM4Py class Arc. The creation occurs in a loop passing over all Arc objects of the original Petri net. An if-condition ensures that during the creation of ArcWithPurposes objects, the correct PlaceIot objects, and not Place objects, are stored in the corresponding source or target attributes. Subsequently, purposes are assigned. This process is shown in Figure 9.

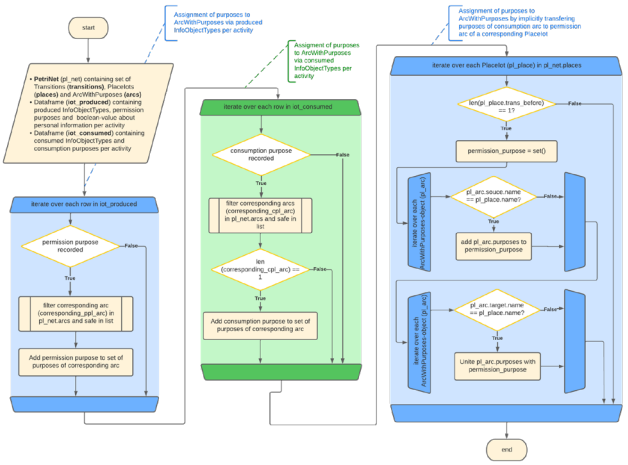

Figure 9: Assignment of Purposes to Edges

4.5 Evaluation

The implementation of the case study was evaluated using various event logs and additional information. By testing with different input data and checking the output data, the correct functioning was observed, at least in the sample space. The test data used for this can also be found in the GitHub project (Weinreuter 2023) and addresses the application field of mobility, especially vehicle information. The event log includes activities such as “analyse influencing factors” and “estimate range”. The information object types are for example, “driver”, “track” or “vehicle” and the consumption purposes, for example, “consider the profile of the route” or “send emergency signal”. In addition to the successful implementation, the evaluation shows that some of the assumptions made (for example, freedom from cycles) are too restrictive for some use cases. Corresponding features must be implemented in the next iteration to be able to deal with these use cases without restrictions.

5 CONCLUSION AND FUTURE WORK

ICPN extend the possibilities of process modeling by the systematic consideration and analysis of aspects related to the areas of information security and data protection (privacy). This equally supports business process managers and those persons responsible for information security and data protection in the design and redesign of business processes. In the future more aspects should be addressed (e.g. other EU-GDPR principles of data processing).

ICPN was used as an example of a language extension, there are also other language extensions to address other issues. The paper has shown that with the idea of augmented process mining results for language extensions models are generated by running the process discovery algorithm and later augmentation of the respective model. Augmented process mining results can be a possibility to adapt process mining procedures to use process mining methods for the different language extensions.

However, this approach is limited to cases in which the process model or its derivation does not depend on the augmented data itself. Otherwise, the additional information would have to be considered in the process discovery step. Therefore, the set of parameters must be extended, because common process mining procedures only consider the case ID, timestamp, and activity name. For example, if an "activity 1" is executed once by "resource A" and once by "resource B" and the further process depends on the executing resource, this information must already be considered by process discovery. Such an adaptation is already planned for future work to enable extended use case coverage.

Privacy compliance analysis using augmented process mining results was demonstrated using Purpose Limitation as an example. Other principles of the EU-GDPR can also be considered. For example, the principle of data minimization can be analyzed by checking whether there is a business process at all that (meaningfully) processes a collected data. For this purpose, it must also be recorded for each activity which data (i.e., for example, which attributes of a data record for a person) are actually processed. This can be considered further in subsequent work for example as an extension of the GitHub open source project augPM (Weinreuter 2023).

ACKNOWLEDGMENTS

The content of this paper is partly a result of the project "SofDCar Software-Defined Car" with eight industry partners and four research partners. This project was supported by the German Federal Ministry for Economic Affairs and Climate Action on a decision by the German Bundestag.

REFERENCES

van der Aalst, W; T. Weijters; and L. Maruster. 2004. “Workflow mining: discovering process models from event logs“. *IEEE Transactions on Knowledge and Data Engineering* 16, No. 9 (Sep), 1128–1142. doi: 10.1109/TKDE.2004.47.

van der Aalst, W. and C. Stahl. 2011. *Modeling Business Processes: A Petri Net-Oriented Approach*, *Illustrated Edition*. Cambridge, The MIT Press.

van der Aalst, W. et al. 2012. “Process Mining Manifesto“. In *Business Process Management Workshops*. Berlin, Heidelberg, 169–194. doi: 10.1007/978-3-642-28108-2_19.

Alpers, S. 2019. *Modellbasierte Entscheidungsunterstützung für Vertraulichkeit und Datenschutz in Geschäftsprozessen*. Karlsruhe, KIT Scientific Publishing. doi: 10.5445/KSP/1000094545.

Azuma, R.T. 1997. “Survey of Augmented Reality, Presence: Teleoperators and Virtual Environments”. In Presence. Teleoperatorsand Virtual Environments (Volume: 6, Issue: 4), p. 355-38, doi10.1162/pres.1997.6.4.355

Azuma, R.T. et al. 2001. “Recent advances in augmented reality, Computer Graphics and Applications”. In IEEE Computer Graphics and Applications (Volume: 21, Issue: 6) , p. 34-47. doi10.1109/38.963459

Berti, A.; S. van Zelst; and W. van der Aalst. 2019. “Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science“. In *Proceedings of the ICPM Demo Track 2019.* Aachen, Germany, Bd. 2374, 13–16. https://ceur-ws.org/Vol-2374/#paper4

Damiani, E.; B. Oliboni; E. Quintarelli; and L. Tanca. 2019. “A graph-based meta-model for heterogeneous data management“. *Knowl Inf Syst* 61, No. 1 (Okt), 107–136. doi: 10.1007/s10115-018-1305-8.

Desel, J. and W. Reisig. “Place/transition Petri Nets“. 1998. In *Lectures on Petri Nets I: Basic Models: Advances in Petri Nets*, W. Reisig and G. Rozenberg (Eds.). Springer, Berlin, Heidelberg, 122–173. doi: 10.1007/3-540-65306-6_15.

van Dongen, B. F.; A. K. A. de Medeiros; H. M. W. Verbeek; A. J. M. M. Weijters; and W. M. P. van der Aalst. 2005. “The ProM Framework: A New Era in Process Mining Tool Support“. In *Applications and Theory of Petri Nets 2005*. Berlin, Heidelberg, 444–454. doi: 10.1007/11494744_25.

ISO/IEC. 2011. “ISO/IEC 15909-2:2011(en), Systems and software engineering — High-level Petri nets — Part 2: Transfer format“. https://www.iso.org/obp/ui/#iso:std:iso-iec:15909:-2:ed-1:v1:en

pandas. 2023. “pandas - Python Data Analysis Library“. https://pandas.pydata.org/. Accessed 14 July 2023.

Treves, N.; L. M. Hillah; F. Kordon; and L. Petrucci. 2009. “A primer on the Petri Net Markup Language and ISO/IEC 15909-2“. In *10th International workshop on Practical Use of Colored Petri Nets and the CPN Tools (CPN’09)*. Aarhus, Denmark. https://hal.archives-ouvertes.fr/hal-01126017

Weinreuter, M. “augPM“. 2023. https://github.com/fzi-forschungszentrum-informatik/augpm. Accessed 14 July 2023.