Towards Privacy-Preserving IoT Systems Using Model Driven Engineering

Judith Michael*, Lukas Netz, Bernhard Rumpe and Simon Varga
Software Engineering, RWTH Aachen University
Aachen, Germany
Email: {surname}@se-rwth.de
*corresponding author

***Abstract*—Considering the Internet of Things in production processes, the human factor and aspects such as data protection and data transparency are often ignored. However, collecting, storing and processing data is going to be a standard procedure in this domain. This includes data from sensors, machines, and processes as well as individual data about people. Recent approaches such as assistive systems for human-computer and human-machine interaction need more personal data than ever before to provide purposeful, tailored support. For MDE approaches it is important to consider privacy already on model level. This paper discusses a way to create privacy-preserving IoT systems using an MDE approach to support privacy and data transparency. We show the relevance and application on a use case from industrial production processes. Additionally, we discuss abilities for practical realization and its limitation.**

***Index Terms*—Domain-Specific Languages, Generated Enterprise Information Systems, Information Portals, Internet of Things, Model-Based Software Engineering Privacy-By-Design, Privacy Modeling**

I. Introduction

Motivation and research gap. Research on the digitization of work processes for the production of the future is currently focusing strongly on technical solutions, such as the interfaces between software and devices (cyber-physical systems), the recognition of work steps and processes with sensors, mathematical evaluations of the collected data and model-based systems engineering [1], [2]. However, the human factor, both as a working person and as an individual, is often not sufficiently taken into account in these considerations. Human actions can influence processes both positively and negatively and are therefore indispensable in an integrated view of production processes and systems. The rise of wearable technologies makes it possible to equip them with miniaturized sensors [3].

In order to assist people in the execution of their work tasks, e.g. by means of body-hugging assistance systems by using motion capture systems for the markerless acquisition of postures to support an ergonomic analysis and improved ergonomic intervention process [4] in a context- and target-group-specific way, or by providing them means to learn about work tasks, e.g., by using smart glasses [5], individual data will have to be collected, processed and stored. However, this development goes hand in hand with questions about the informational self-determination, the security of data collected as well as data protection and transparency.

To consider privacy in systems design, Hoepman et al. [6], [7] introduce and discuss eight privacy design strategies: minimize, hide, separate, abstract, inform, control, enforce and demonstrate. These design strategies have already been taken into account when discussing our ideas for a privacy model and an according system architecture in [8]. This paper goes a step further and discusses them in relation with model- driven engineering (MDE).

We believe that MDE and model-based software engineering (MBSE) can well help to incorporate privacy considerations at model level. It can provide means to support the aforementioned privacy design strategies.

Research question. These considerations lead us to the following research question: *How is it possible to include privacy considerations in the MDE development process already on model level?*

Contribution. The approach presented in this paper uses MDE tools and frameworks together with a set of domain specific languages (DSLs) to create an Enterprise Information System (EIS) considering privacy- preservation and provide users and data providers with the relevant information to make informed decisions about their data use. We show a possible DSL model structure including domain models, a privacy model and possible instantiations as well as relevant aspects which have to be considered for the system design, e.g., privacy checkpoints.

In previous work [8], we have already presented a concept for user-centered privacy-preserving process mining systems design for IoT. This paper goes a step further and discusses the inclusion of privacy considerations for an MDE approach and appropriate tooling.

The MDE tooling we use in our example for a realization are MontiCore [9] and MontiGEM [10]. MontiCore is a workbench for modeling language development which supports the agile and compositional development of DSLs. MontiGEM, the Generator for Enterprise Management, is based on MontiCore and uses (1) a set of models which are (2) parsed and transformed using a template engine towards (3) the target, namely output files in the target language. As a result, MontiGEM creates an information system out of class diagrams and graphical interface models.

Overview. The next section discusses the term privacy and general concepts for privacy-preserving systems design.

Section III presents a use case from the IoT domain, namely a production process and shows its representation in a domain specific data model. Section IV shows our idea on how to combine privacy-preserving IoT systems and MDE approaches. We show an exemplary system architecture and relevant privacy checkpoints (PrC), the needed privacy model to support the execution of the privacy checkpoints, concrete examples for a purpose tree and the privacy policies (PPs) including privacy policy rules (PPRs) and the description on how to compare PPs and make the decision if data should be provided after a request or not. Section V discusses our approach in comparison to other approaches, weaknesses and limitations of it and advantages on using it. The last section summarizes and concludes our paper.

II. Privacy and Privacy-Preserving System Design

The term privacy is related to informal self-determination, which means the ability to decide what information about a person is passed on [11]. Whereas this paper has a strong focus on privacy, the term is strongly related to security, the notion of trust and data sovereignty.

To ensure data privacy, *security* provides the needed foundations as it preserves the confidentiality, integrity and availability of information and supports the authenticity, accountability, non-repudiation and reliability [12]. An important aspect is access control. There exist different variants such as role-based access control (RBAC) [13], policy-based access control, also known as attribute-based access control (ABAC) [14] or combinations of RBAC and attributes [15]. As the proposed approach needs a detailed way to define who gets access to what data, we use privacy policies together with ABAC.

To *trust* a person or system and in a next step to share date with them, it depends on several factors such as past interactions, what relationship exists to each other, similar personality attributes such as interests or the sensitive nature of the data we are sharing at that moment in time [16]. To ensure consent for data use and show the purpose for each data capture helps to build trust in organizations [3]. It is important that employees are in control of their personal data.

Our understanding of *data sovereignty* is related with the personal rights of the people from whom the data originate [17]. Acquisti et al. [18] state the importance to protect individuals with minimal requirement of informed and rational decision making and that it is important for privacy policies to have a baseline framework of protection already included. By using models and generative approaches, it is possible to develop privacy policies which already include baseline protection. Moreover, the generation of an information portal could help to keep users informed about their data.

It is important to comply with privacy regulations such as the Europe's General Data Protection Regulation (EU GDPR). Due to the EU GDPR[19] it is important to consider privacy throughout the complete development process. Privacy-by-design [20] is the most prominent approach to consider privacy already in the development process when designing a new technology.

To take privacy in systems design into account, Hoepman et al. [6] introduce eight privacy design strategies: minimize, hide, separate, abstract, inform, control, enforce and demonstrate. They should be considered for privacy-by-design approaches, which are compliant with EU GDPR and must be also seen as requirements for the design of privacy-preserving IoT systems, no matter if the design approaches are model-based, model-driven or without relation to modeling at all. These strategies are discusses in [3] in relation with privacy challenges in human-centered industrial environments especially considering process mining. This includes minimization, aggregation, traceability, monitoring and transparency, deletion, consent and purpose, trust and acceptance, privacy vs. benefit, auditing and privacy breaches.

Regarding IoT, privacy became relevant with the massive deployment of sensors in various environments (see section V). It is possible to collect data and information about people and to use analytic tools to profile users and identify them even from anonymized data [21]. The following use case will show an example for data collection in working environments and thus, where privacy considerations become important.

III. IoT Use Case & MDE

When investigating IoT in production processes, humans are often not taken into account as much as necessary. Our use case shows examples where it is important to consider human privacy concerns when collecting, storing and processing data in such processes. Moreover, we show an excerpt of the domain model including data needed for our running example. Please remark that our approach can be applied onto other use cases as well, as the domain information is well encapsulated.

A. Humans in IoT production processes

Fig. 1 shows one station of a manufacturing area. There, several operators and robots collaborate in the production process. We use an IoT box as product to exemplary describe the process. The process steps for the IoT box assembly are: put the lower part of its case on the conveyor belt, assemble the different components such as the USB-port, serial ports, a WiFi and bluetooth module and HDMI port, put the upper part on top, test the functionality of the box and lift it off the assembly line into the transport boxes which are moved to the next production line for shipping.

There are several operators included in this process which are wearing (1) smart glasses and (2) smart watches and are using (2) smartphones and (3) tablets. All of these devices are able to collect data about the usage and location in the manufacturing area. Moreover, health data could be processed for different purposes, e.g., to detect physical and mental stress [22], or to analyze ergonomics. Motion capture systems (including cameras, smart clothes or other technologies) are used for the ergonomic analysis of postures of the operators, to support the ergonomic intervention process and an optimization of the daily personnel deployment planning [4], [23].

The plastic parts of the product (4) include RFID chips which make it possible to track them during the assembly

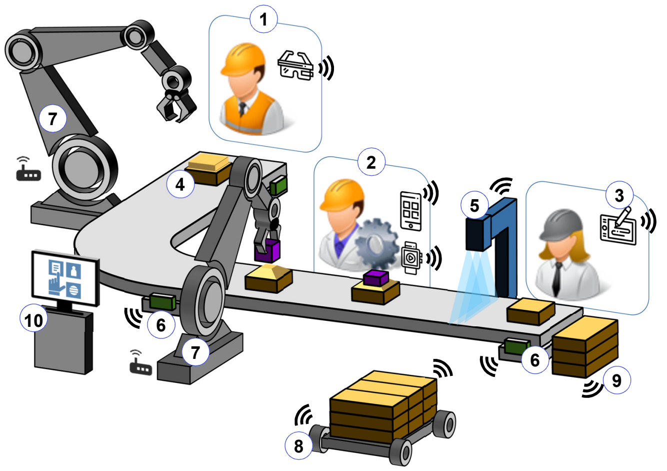


Fig. 1: Station with workers, machines and sensory devices.

process. The assembly line itself (6) and involved machines such as the one for functionality testing (5) trace the product to recognize in which assembly step the process is. The robots (7) mainly support the lift onto and off process steps and (8) the transportation of needed resources and the final product in the transport boxes (9). These could be again tracked using RFID technology. Moreover, information portals (10) could provide the relevant information needs. Mobile and smart devices (1)-(3) could provide this information as well.

Starting with this real life scenario, we create a domain model including relevant context information and data collected in various ways. Clearly, MDE approaches need domain information to create the database and persistence layer.

B. Domain Model

Considering the use case in Fig. 1, we create the domain model including all relevant persons, their abilities, machines, resources, processes and locations as well as attributes for handling sensor data. As suggested in [24], we split it into the main context areas: personal and social context, environmental context, spatial context and behavioral context. Listing 1 shows an excerpt of these concepts and their relations in the notation of the class diagram for analysis (CD4A) language [25]. Models in this textual modeling language can be used by MontiCore and MontiGEM generators to create the according data structure [10].

For the *personal and social context* employees, relationships between persons, their abilities and other personal data is relevant. There exists the general concept Person including information such as the name and the postal address. For Employees it might be relevant to have their birthday and employment dates. Operators (line 12) and other user groups can inherit from this concept. Operators have again specific attributes such as their shoe size to be able to provide them the right safety shoes or their position in the company.

For assistance purposes HealthData (lines 17-23) is needed, e.g., the heart rate, blood pressure or current stress level. For ergonomic analyzes the SkeletonModel including joints and relations in-between them is relevant as well. This data could be collected via smart devices and depth cameras.

CD4A..

```
1  package de.IoT.production;
2  classdiagram domainModel {
3
4    //Personal and Social Context
5    //Person, Employee, Supplier,...
6    class Employee extends Person {
7      ZonedDateTime employmentStart;
8      <Optional> ZonedDateTime employmentEnd;
9      ZonedDateTime birthday;
10   }
11
12   class Operator extends Employee {
13     long shoeSize;
14     String gpsPosition;
15   }
16
17   class HealthData {
18     int heartRate;
19     String bloodPressure;
20     /String stressLevel;
21     /List<String> ergonomicProblems;
22     ZonedDateTime timestamp;
23   }
24
25   class Ability {
26     String name;
27   }
28
29   association [*] Ability -> (type) AbilityType [1];
30
31   class AbilityLevel {
32     String level;
33     ZonedDateTime atTime;
34     Operator person;
35   }
36
37   association [*] AbilityLevel -> Ability [1];
38   association [1] Operator -> HealthData [*];
39   association [*] Operator -> Ability [*];
40   association [1] Operator -> AbilityLevel [*];
41
42   //Environmental Context
43   //Resource, Device, Item, Machine, SmartWatch...
44   class Function {
45     String name;
46     String description;
47     List<Abilities> neededAbilities;
48   }
49
50   association [*] Function -> (type) FunctionType [1]
51   association [1] Device -> Function [*];
52
53   class QualityCheckMachine extends Machine {
54     boolean qualityOk;
55   }
56
57   association [*] QualityCheckMachine -> (minLevel)
         AbilityLevel [1];
58
59   //Spatial Context: Location, Area, Equipment,...
60   class Station {
61     String stationName;
62     ZonedDateTime lastFilling;
63     ZonedDateTime lastPickUp;
64     ZonedDateTime medianDowntime;
65     List<int> hourlyProducedUnits;
66   }
67
68   association [1] Area -> Station [*];
69
70   //Behavioral Context: Operation, Goal, Flow,...
71 }
```

Listing 1: Data model in CD4A notation (excerpt)

Every operator has certain Abilities (lines 25-27). There are different AbilityTypes, such as the ability to control a certain machine type, do a specific process step, having a certain driving license, a specific certificate or further education. With this knowledge it is possible to know what person to place on which position in the company. The AbilityLevel (lines 31-35) defines the concrete level of ability the person has at a certain time. This might be influenced by physical and metal restrictions at a certain time which are reflected in the HealthData.

Other relevant data could be e.g., FinancialData to be able to make the salary payment, or pictures for access control. Also Suppliers and Customers might be relevant, e.g., for customer and supplier relationship management processes.

The *environmental context* describes Resources, which are needed to perform certain process steps. Possible resource types are device, item, fixture and application. Devices such as Robots could be further specified into IndustrialRobot, TransportRobot or other needed variants. Further relevant devices are Machines, such as the QualityCheckMachine (5) in Figure 1, or smart devices such as SmartWatches, SmartGlasses or SmartPhones. It is possible to define Functions (lines 44-48) for Devices and list relevant Abilities (line 47) and AbilityLevels (line 57) to be able to use or operate a certain resource. This is relevant for assisting employees.

The *spatial context* defines all elements relevant for navigation, mobility and virtual relationships. The definition of the relevant buildings, areas and other parts are strongly dependent on the concrete company and its structure. Starting from the Location, it is possible to define relevant FactoryBuildings, Areas on certain floors or Stations (lines 60-66) and relations among them (line 68). To relate certain Resources to a special Area or Station modelers can define Equipment on a certain position.

The *behavioral context* (not further described in Listing 1) includes Behavioral Units, Operations, connections between operations and Goals as well as Events and Traces from event logs (see [26], [8] for details).

Clearly, the class diagram is not complete but it shows the most relevant classes and attributes for discussing privacy considerations of our use case. MDE approaches can be used to create the persistence layer and databases out of this model.

IV. Privacy-Preserving IoT Systems and MDE

The next steps towards privacy-preserving system design for IoT using an MDE approach are (1) to discuss privacy-preserving systems design in the system architecture of our use case including relevant privacy checkpoints and (2) the privacy model which is needed to define the most relevant privacy data. We show (3) concrete examples for a purpose tree and the privacy policies and (4) the description on how to compare privacy policies and make the decision if data should be provided after a request or not.

A. *Privacy-Preserving System Design*

In [8], we have already discussed an approach for user-centered privacy-preserving system design for systems combining process mining and information systems. Fig. 2 presents an overview of the high level system architecture for human-centered industrial environments: (A) Data from resources and sensory devices connected with operators is (B) collected, e.g., via human activity recognition systems and a common observation interface [27]. The data is either (C) stored and afterward (D) used for the main reasons why it was stored (Primary Use) or it might be directly used (D) after collection (B). After a defined time (E) the data should be removed/deleted from the data storage (C) or it might be directly removed after the primary use (D). The data might be used (F) for other services than the ones that affect the employees directly, e.g. for calculation which process steps cause the highest stress level to optimize production processes. Again, data removal (E) after this use should be ensured. Our approach includes an information portal (G), to provide a user friendly representation of stored data, data access attempts, the management of policies and foresee privacy preservation strategies for each data pass.

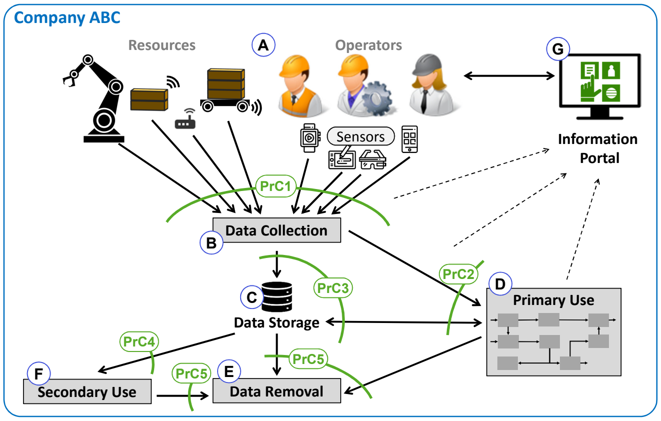


Fig. 2: System architecture with privacy checkpoints

On each data pass, we have introduced and extended the privacy checkpoints (PrC 1- PrC5) from [3] to our use case. They show at which points it is important to consider privacy also when using MDE approaches.

- **PrC 1:** Inform in (G) which data is collected, the duration of storage, possibilities for data removal and how raw data is combined (PP for data collection). Operators can give their consent and withdraw it. The portal ensures privacy control, the traceability of data needs to be ensured in the system architecture by considering all checkpoints.
- **PrC 2:** Inform in (G) which (real-time) analysis will be conducted and/or which services will receive the data for what purpose (PP for data use), about risks and benefits of analyzes and services, which services cannot be provided without access to the data and again options to delete the data at any point.

- **PrC 3:** Inform in (G) for what purpose the data is used for (primary and secondary use), provide an option to determine how long data can be stored, provide possibilities to determine who has access to the data and obtain consent (PP for data storage and data use). A sustainable level of abstraction has to be considered before storing the data, unnecessary personal data has to be anonymized before storing (PP for data storage). (C) needs to provide means for data encryption and empower the data provider to be in control of that. If new purposes for data use occur (additions in the purpose tree) or other attributes of data are relevant for a purpose as well (changes in the purpose tree), (G) needs to inform the employee about these changes and provide possibilities to define new PPRs or change existing ones.
- **PrC 4:** Inform in (G) which service has asked for access to the data and to whom access was granted (comparison of PPs), about aggregation with other data, if the data is exported or shared with a 3rd party. If new purposes occur, the employee has to be asked for consent again (define a new PPR). Moreover, the employee should have the ability to see the results of the secondary use.
- **PrC 5:** (G) provides possibilities to delete the data at any point (during, at the end or after a service). Based on the retention time in the PPRs the deletion has to be done automatically after a certain period. After a deletion request, the data controller has to ensure that also analysis results and aggregated data are only kept if no connection to the data provider is possible.

These privacy checkpoints have to be considered in the system architecture.

B. Privacy Model

The next step towards privacy-preservation is to define the privacy model which includes the most relevant data to identify important roles, define privacy preferences, define a companies purposes for data processing and to handle data requests. As discussed in [8], we rely on attribute-based access control and use privacy policies and rules.

Figure 3 shows the relevant privacy data as a class diagram and how it is related. Privacy-preserving systems need at least four roles: the `DataProvider`, the `DataConsumer`, the `DataController` and a `DataAuthority`. The `DataProvider` is the data source and should be enabled to verify the correct use of his data. Thus, it defines a `PrivacyPolicy` where it defines e.g., who can do what with his data. The `DataConsumer` is an entity with an interest in the data. It has to define a `PrivacyPolicy` which declares e.g., what it wants to do with which data for what purpose. Note that it has to be ensured, that only one relation between `PrivacyPolicy` and either `DataProducer` or `DataConsumer` can exist.

The `DataController` processes and stores the data and has to ensure the correct use of it. The `DataAuthority` is able to control the processing of data and can check compliance with data protection regulations.

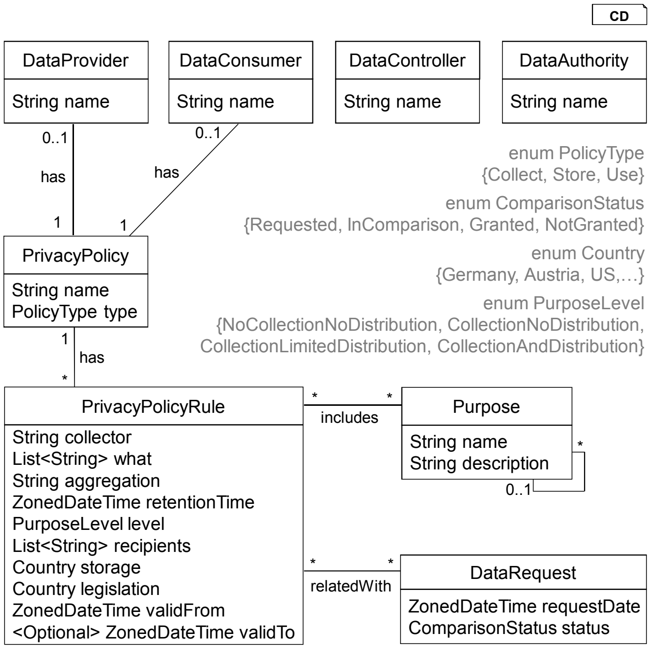


Fig. 3: Privacy Model

Each `PrivacyPolicy` (either for collecting, storing or using data) consists of several `PrivacyPolicyRules`. They define very detailed (1) who collects the data (collector), (2) what attributes are collected and/or stored, (3) on which aggregation level the data is stored, e.g., each person, station, production line, and daily, weekly, monthly, (4) how long the data could be stored (retention time), (5) what purpose level is addressed (see `enum PurposeLevel`), (6) which recipients are allowed to have the data, (7) in which country the data is stored and (8) the legislation of the country in which the data processing is carried out. Additionally, it is important to store historical information to know which PPR was valid when.

Every `PrivacyPolicyRule` is related to one or more `Purposes`. They can have further hierarchies as each purpose can be related with another purpose. Constraints need to check that the purposes for a tree structure in order to be computable.

`PrivacyPolicyRules` can be related with several `DataRequests`. Here it is stored who has requested access to this data by using which `PrivacyPolicyRule` and if the access to it was granted or not.

This privacy model is domain independent, so please remark that the privacy model is used additionally to the domain model. This means that relations between privacy and domain model have to be added such as the definition which class of persons can be a data provider or consumer, e.g., via additional and/or external tagging of the domain model.

C. Instances of Privacy Policies and Purpose Trees

Additionally to the defined models (domain and privacy model) it is important to define a concrete instance of the purpose tree and instances of the PPs (see Figure 4). The *purpose tree* should be defined by the company which collects the data to make it possible to use the purposes in the *instances of the PPs* which are defined by operators in a further step.

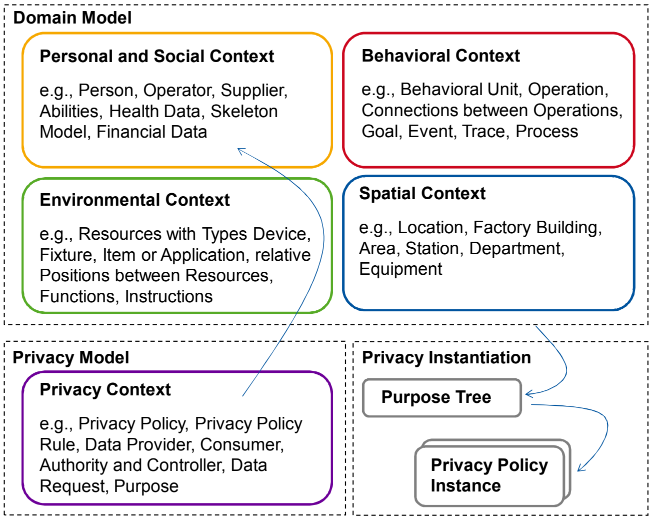

Domain Model
Personal and Social Context
e.g., Person, Operator, Supplier, Abilities, Health Data, Skeleton Model, Financial Data
Behavioral Context
e.g., Behavioral Unit, Operation, Connections between Operations, Goal, Event, Trace, Process
Environmental Context
e.g., Resources with Types Device, Fixture, Item or Application, relative Positions between Resources, Functions, Instructions
Spatial Context
e.g., Location, Factory Building, Area, Station, Department, Equipment
Privacy Model
Privacy Context
e.g., Privacy Policy, Privacy Policy Rule, Data Provider, Consumer, Authority and Controller, Data Request, Purpose
Privacy Instantiation
Purpose Tree
Privacy Policy Instance

Fig. 4: Models and related instantiations

There are several different ways for data controllers using MDA approaches to define a purpose tree: it is possible to use object diagrams (OD), a tagging language or any DSL with a tree like structure. Figure 5 shows an excerpt of such a purpose tree by using a simple graphical representation. The attributes named at the leaf level of the tree are clearly related to the ones in the domain model. Thus, it is important to check the purpose tree instance and domain model for consistency and to tag used attributes with the purpose in the domain model.

In our concrete example *productivity* analysis and to provide assistance e.g., by using *ergonomic analysis* or *stress detection* are relevant purposes. Figure 5 shows the related attributes for each of them. Other relevant purposes are e.g., to make the work contract by the human resources department, salary payment for the financial department, health insurance payments, access control or quality assurance.

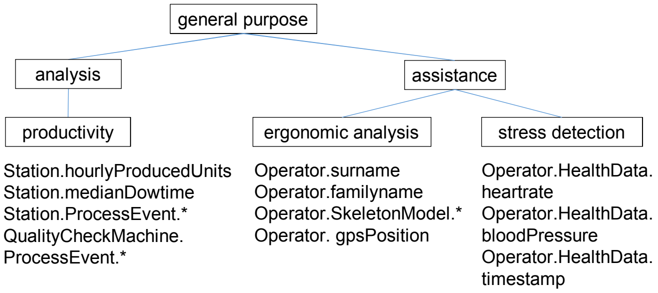
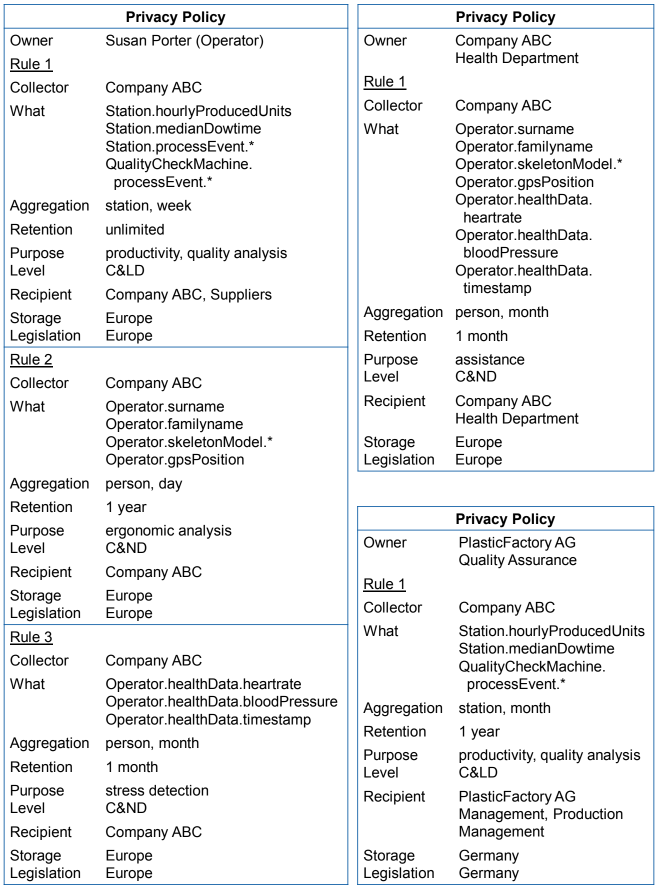


Fig. 5: Example purpose tree (excerpt)

In a next step the operators define their PP instances. Figure 6 shows some examples, whereas the left side shows the PP of type *use* including three rules of a data provider (Susan Porter) and the right side two PP instances of different data consumers. These PP definition processes of data providers and data consumers happen independent from each other. Susan has defined a rule for productivity and quality analysis, one for ergonomic analysis and one for her data for stress detection. Her employers health department has defined one for providing assistance based on the health data. The quality assurance department of a supplier has defined a PP as well for productivity and quality analysis purposes.



Privacy Policy	
Owner	Susan Porter (Operator)
Rule 1	
Collector	Company ABC
What	Station.hourlyProducedUnits Station.medianDowtime Station.processEvent.* QualityCheckMachine. processEvent.*
Aggregation	station, week
Retention	unlimited
Purpose Level	productivity, quality analysis C&LD
Recipient	Company ABC, Suppliers
Storage Legislation	Europe Europe
Rule 2	
Collector	Company ABC
What	Operator.surname Operator.familyname Operator.skeletonModel.* Operator.gpsPosition
Aggregation	person, day
Retention	1 year
Purpose Level	ergonomic analysis C&ND
Recipient	Company ABC
Storage Legislation	Europe Europe
Rule 3	
Collector	Company ABC
What	Operator.healthData.heartrate Operator.healthData.bloodPressure Operator.healthData.timestamp
Aggregation	person, month
Retention	1 month
Purpose Level	stress detection C&ND
Recipient	Company ABC
Storage Legislation	Europe Europe

Privacy Policy	
Owner	Company ABC Health Department
Rule 1	
Collector	Company ABC
What	Operator.surname Operator.familyname Operator.skeletonModel.* Operator.gpsPosition Operator.healthData. heartrate Operator.healthData. bloodPressure Operator.healthData. timestamp
Aggregation	person, month
Retention	1 month
Purpose Level	assistance C&ND
Recipient	Company ABC Health Department
Storage Legislation	Europe Europe

Privacy Policy	
Owner	PlasticFactory AG Quality Assurance
Rule 1	
Collector	Company ABC
What	Station.hourlyProducedUnits Station.medianDowtime QualityCheckMachine. processEvent.*
Aggregation	station, month
Retention	1 year
Purpose Level	productivity, quality analysis C&LD
Recipient	PlasticFactory AG Management, Production Management
Storage Legislation	Germany Germany

Fig. 6: Examples for privacy policy instances

D. Comparison of Privacy Policies and Decision Making

The data controller has to compare the policies of data consumers and providers to allow data transmissions. In case of policy conflicts, the highest data protection restriction of one or more data providers always win. If no PPR is defined for a purpose, there is no access granted to the data. The system compares each policy element of potential rules of the provider and consumer and decides whether access is granted or not. Decisions are stored as a DataRequest object.

Figure 6 shows an example: When the health department asks for the data defined in the PPR, each attribute has to be compared. The attributes in *what* are defined in Rule 2 and 3 of the data provider, the purpose *assistance* is in the purpose tree above *ergonomic analysis* and *stress detection*, so Rules 2 and 3 are relevant. The aggregation level *month* is the same as in Rule 3 and more general as in Rule 2, retention time is the same as in Rule 3 and less than in Rule 2, the purpose levels are the same, as well as recipient, storage, and legislation. Thus, access would be granted.

The same occurs for the PlasticFactory AG and their data request for the data provider: The attributes they request are less than in Rule 1 and the according purposes in the purpose tree, the aggregation level is higher with a month compared to a week, retention time is unlimited in Rule 1 and thus irrelevant, storage and legislation are in Europe. As a result of the comparison, access is granted. These decisions are stored in the `DataRequest` class of the Privacy Model in Figure 3.

Code Generation. The described models can be used as input for MontiGEM to generate the system code and the information portal (see (G) in Figure 2) [10]. The privacy checkpoints needs to be included in the architecture to provide PP checks in each application interface, e.g. database connection or network communication, to make sure the policies are fulfilled at all times. Using architecture description languages, such as MontiArc [28], a relation has to be established between the privacy checkpoints and the communication between components of the basic system architecture. Further details are currently under investigation.

V. Discussion and Related Work

Related work. Improvements for human workers regarding the interaction with production systems in processes are already ongoing work such as the transformation of the shop floor into a smart environment with multimodal interaction facilities to bridge the gap between physical world and the digital part of the production system [29].

Considering privacy, security and trust in the *IoT domain*, a broad variety of approaches exist. Nevertheless, most of the lacks to tackle the human factor including information for and control by involved humans. Sicari et al. [30] provide an extensive overview of security requirements as well as privacy, trust, enforcement, secure middlewares and mobile security in IoT. [31] discusses open issues for security and privacy. [32] discusses the security and privacy of IoT architectures and systems but lacks to discuss the human factor. [33] presents a security-and quality-aware system architecture for IoT systems considering data quality including data annotation. Moreover, the IoT-A privacy model [21] includes functional components for aspects such as identity management, authentication, authorization, trust and reputation. [34] considers information privacy research in information systems. [3] discussed technological and organizational privacy challenges for process mining in human-centered industrial environments. There exist approaches to consider privacy in the *system architecture* such as [35], discussing privacy-friendly systems in case of privacy-by-policy, privacy-by-architecture or privacy-by-design [20].

Work on *model-based* and *model-driven* approaches considering privacy exist mainly in other domains, e.g., [36] discuss MDE for privacy management in business ecosystems or [37] in e-Health systems. The general idea to combine privacy-preserving system design with MDE approaches was shortly introduced in [8]. To the best of our knowledge no other approach exists which combines MDE and privacy engineering for the IoT domain.

Weaknesses and Limitations of the approach. The proposed approach is easily applicable for greenfield design, we expect that for adding privacy consideration into existing projects and architectures further considerations have to be made. This paper does not discuss security issues such as data encryption or decryption [33] or privacy preserving techniques applied directly on data such as k-anonymity [33] or differential privacy [38]. A challenging aspect for IoT systems is the continuous addition of interfaces. Here it is important to reconsider relevant PrCs every time a new interface is added as it is a possible privacy leak. This can be improved by automated checks of the interfaces against the architectural models (including the PrCs) at compile time. Clearly, this needs further investigation. The purpose tree instance has to be kept up to date by the data controller himself. He has to be aware of changes in the real life (e.g. new purposes for data use) and is responsible for the ongoing maintenance of the privacy aspects of the system. Nevertheless, this also occurs for system design without MDE approaches. Moreover, further investigations about the useability and understandability of the PPs by users have to follow.

Advantages using the approach. The use of MDE approaches improves the maintainability of privacy-preserving systems: for changing domain models or PrCs in architectural models, continuous re-generation facilitates the creation and change process and improves consistency requirements. As such changes might have effects on the operators' privacy policy instances, these can be automatically checked against the domain model and suggest changes or additions for users. Further investigations of the maintainability are ongoing work.

Regarding the design strategies [6], creating an information portal supports users as they are *inform*ed and have fully *control* over their data collection, storage, operation, and dissemination. The information portal provides means for operators and data consumers to easily create, read, update and delete privacy policies and their rules and for data controllers to easily maintain their purpose tree instance. The approach *enforce*s data controllers for creating, ensuring, and complying with contractual and legal policy obligations. Moreover, it helps to *demonstrate* the data authority that a controller adheres to legal requirements including auditing, logging, and reporting. The minimize, hide, separate and abstract strategies are strongly related with database functions itself, so how they are related with MDE approaches needs further investigation.

Our approach is domain independent and can thus be applied onto other use cases and scenarios where data is collected, stored and processed as well. Domain information is only included in the domain CD and the mapping between the domain model and the privacy model.

VI. Conclusion

This paper presents an approach to include privacy considerations in the MDE development process of IoT systems and shows its application to a use case from human-centered industrial environments. We use a set of DSLs and MDE tools and frameworks to create an information system considering privacy-preservation and provide users and data providers with the relevant information to make informed decisions about their data use. We show a possible DSL model structure including domain models, a privacy model and possible instantiations as well as relevant aspects which have to be considered for the system design such as privacy checkpoints.

To sum up, our approach is easily applicable on similar use cases and system designs. Privacy preservation is important for other IoT systems as well such as assistive systems in general, smart home environments, wearables and health applications. Further investigations in other domains will follow.

References

[1] A. L. Ramos, J. V. Ferreira, and J. Barcelo, "Model-based systems engineering: An emerging approach for modern systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 101–111, 2012.

[2] K. Hölldobler, J. Michael, J. O. Ringert, B. Rumpe, and A. Wortmann, "Innovations in model-based software and systems engineering," *The Journal of Object Technology*, vol. 18, no. 1, pp. 1–60, Jul 2019.

[3] F. Mannhardt, S. Petersen, and M. Fradinho Duarte de Oliveira, "Privacy challenges for process mining in human-centered industrial environments," in *Intelligent Environments 2018*. IEEE Xplore, 2018.

[4] C. Brandl, D. Bonin, A. Mertens, S. Wischniewski, and C. M. Schlick, "Digitalisierungsansätze ergonomischer analysen und interventionen am beispiel der markerlosen erfassung von körperhaltungen bei arbeitstätigkeiten in der produktion," *Zeitschrift für Arbeitswissenschaft*, vol. 70, no. 2, pp. 89–98, 2016.

[5] M. Spitzer, I. Nanic, and M. Ebner, "Distance learning and assistance using smart glasses," *Education Sciences*, vol. 8, no. 1, p. 21, 2018.

[6] J.-H. Hoepman, "Privacy design strategies," in *ICT Systems Security and Privacy Protection*. Springer, 2014, pp. 446–459.

[7] M. Colesky, J. Hoepman, and C. Hillen, "A critical analysis of privacy design strategies," in *IEEE Security and Privacy Workshops (SPW)*, 2016, pp. 33–40.

[8] J. Michael, A. Koschmider, F. Mannhardt, N. Baracaldo, and B. Rumpe, "User-centered and privacy-driven process mining system design for iot," in *Information Systems Engineering in Responsible Information Systems*, ser. LNBIP. Springer, 2019, vol. 350, pp. 194–206.

[9] K. Hölldobler and B. Rumpe, *MontiCore 5 Language Workbench Edition 2017*, ser. Aachener Informatik-Berichte, Software Engineering, Band 32. Shaker Verlag, December 2017.

[10] K. Adam, J. Michael, L. Netz, B. Rumpe, and S. Varga, "Enterprise information systems in academia and practice: Lessons learned from a mbse project," in *Digital Ecosystems of the Future: Methods, Techniques and Applications (EMISA'19)*, ser. LNI, 2019, pp. 1–8, (in press).

[11] E. Bergeron, "The difference between security and privacy," 2000. [Online]. Available: https://www.w3.org/P3P/mobile-privacy-ws/papers/zks.html

[12] I. 27000, "Information technology - security techniques - information security management systems - overview and vocabulary," International Organization for Standardization, Standard, 2018, fifth edition, 2018-02.

[13] P. Colombo and E. Ferrari, "Privacy aware access control for big data: A research roadmap," *Big Data Research*, vol. 2, no. 4, pp. 145–154, 2015.

[14] L. Wang, D. Wijesekera, and S. Jajodia, "A logic-based framework for attribute based access control," ser. FMSE '04. ACM, 2004, pp. 45–55.

[15] D. R. Kuhn, E. J. Coyne, and T. R. Weil, "Adding attributes to role-based access control," *Computer*, vol. 43, no. 6, pp. 79–81, 2010.

[16] O. Sacco, J. G. Breslin, and S. Decker, "Fine-grained trust assertions for privacy management in the social semantic web," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013, pp. 218–225.

[17] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *IEEE Int. Confe. on e-Health Networking, Applications and Services (Healthcom)*, 2016, pp. 1–3.

[18] A. Acquisti, L. Brandimarte, and G. Loewenstein, "Privacy and human behavior in the age of information," *Science*, vol. 347, no. 6221, pp. 509–514, Jan. 2015.

[19] European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council 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 (GDPR)," *Official Journal of the European Union*, vol. L119, pp. 1–88, 2016.

[20] P. Schaar, "Privacy by design," *Identity in the Information Society*, vol. 3, no. 2, pp. 267–274, 2010.

[21] J. Holler, V. Tsiatsis, C. Mulligan, S. Avesand, S. Karnouskos, and D. Boyle, *From Machine-to-Machine to the Internet of Things*. Burlington: Elsevier Science, 2014.

[22] M. Fellmann, F. Lambusch, and A. Waller, "Stress-sensitive it-systems at work: Insights from an empirical investigation," in *Business Information Systems, Int. Conf. (BIS); Part II*, ser. LNBIP, W. Abramowicz and R. Corchuelo, Eds., vol. 354. Springer, 2019, pp. 284–298.

[23] P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, Eds., *Human-computer interaction - INTERACT 2011; part II*, ser. LNCS. Berlin: Springer, 2011, vol. 6947.

[24] J. Michael and C. Steinberger, "Context modeling for active assistance," in *Proc. of the ER Forum 2017 and the ER 2017 Demo Track colocated with the 36th Int. Conference on Conceptual Modelling (ER 2017)*, C. Cabanillas, S. España, and S. Farshidi, Eds., 2017, pp. 221–234.

[25] B. Rumpe, *Modeling with UML: Language, Concepts, Methods*. Springer International, July 2016.

[26] J. Michael and H. C. Mayr, "Conceptual modeling for ambient assistance," in *Conceptual Modeling - ER 2013*, ser. LNCS, vol. 8217. Springer, 2013, pp. 403–413.

[27] V. A. Shekhovtsov, S. Ranasinghe, H. C. Mayr, and J. Michael, "Domain Specific Models as System Links," in *Advances in Conceptual Modeling Workshops (ER'18)*. Springer, 2018, pp. 330–340.

[28] A. Haber, J. O. Ringert, and B. Rumpe, "MontiArc - Architectural Modeling of Interactive Distributed and Cyber-Physical Systems," RWTH Aachen University, Technical Report AIB-2012-03, February 2012.

[29] K. Schilling, S. Storms, and W. Herfs, "Environment-integrated human machine interface framework for multimodal system interaction on the shopfloor," in *Advances in human factors and systems interaction*, ser. Advances in Intelligent Systems and Computing, I. L. Nunes, Ed. Cham: Springer, 2019, vol. 781, pp. 374–383.

[30] S. Sicari, A. Rizzardi, L. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146 – 164, 2015.

[31] M. Abomhara and G. M. Koien, "Security and privacy in the internet of things: Current status and open issues," in *Int. Conference on Privacy and Security in Mobile Systems (PRISMS)*. IEEE, 2014, pp. 1–8.

[32] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier, and P. Kikiras, "On the security and privacy of internet of things architectures and systems," in *2015 International Workshop on Secure Internet of Things*. Piscataway, NJ: IEEE, 2015, pp. 49–57.

[33] S. Sicari, C. Cappiello, F. de Pellegrini, D. Miorandi, and A. Coen-Porisini, "A security-and quality-aware system architecture for internet of things," *Information Systems Frontiers*, vol. 18, no. 4, pp. 665–677, 2016.

[34] Bélanger and Crossler, "Privacy in the digital age: A review of information privacy research in information systems," *MIS Quarterly*, vol. 35, no. 4, p. 1017, 2011.

[35] S. Spiekermann and L. Cranor, "Engineering privacy," *IEEE Trans. on Software Engineering*, vol. 35, no. 1, pp. 67–82, 2009.

[36] C. Feltus, E. Grandry, T. Kupper, and J.-N. Colin, "Model-driven approach for privacy management in business ecosystem," in *5th Int. Conf. on Model-Driven Engineering and Software Development*, INSTICC. SciTePress, 2017, pp. 392–400.

[37] F. Amato and F. Moscato, "A model driven approach to data privacy verification in e-health systems," *Trans. Data Privacy*, vol. 8, no. 3, pp. 273–296, 2015.

[38] F. Mannhardt, A. Koschmider, N. Baracaldo, M. Weidlich, and J. Michael, "Privacy-preserving process mining: Differential privacy for event logs," *Business & Information Systems Engineering (BISE)*, pp. 1–33, 2019, (in press).