IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 3, 2021, accepted April 13, 2021, date of publication April 26, 2021, date of current version May 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3075568

Trust Mining: Analyzing Trust in Collaborative Business Processes

MARCEL MÜLLER[1], NADINE OSTERN[2], DENIS KOLJADA[1], KAI GRUNERT[1], MICHAEL ROSEMANN[3], AND AXEL KÜPPER[1]

[1]Chair for Service-centric Networking, Telekom Innovation Laboratories, Technische Universität Berlin, 10587 Berlin, Germany
[2]Chair for Digitization and Process Management, Philipps-University Marburg, 35032 Marburg, Germany
[3]Centre for Future Enterprise, Queensland University of Technology, Brisbane, QLD 4000, Australia

Corresponding author: Marcel Müller (marcel.mueller@tu-berlin.de)

This work was supported in part by the German Research Foundation, and in part by the Open Access Publication Fund of Technische Universität (TU) Berlin.

ABSTRACT The ongoing digital transformation and internationalization of business processes cause a shift towards a more collaborative nature of processes. In such collaborations, different organizations execute separate parts of the process autonomously. This fragmentation leads to uncertainty regarding the correct execution of activities, the proper workflow, and data in the process flow. If organizations engage in business together, trust is needed. Therefore, we propose Trust Mining as an analytical approach to better understand uncertainties and the potential trust issues that arise from them. Trust Mining takes a business process model as an input and analyzes uncertainties and relationships. In the end, an evaluation regarding the trust requirements of specific stakeholders is given. Furthermore, we present a prototypical implementation and illustrate how Trust Mining can be used in trust-aware process (re-)design.

INDEX TERMS Business process management, trust, collaboration.

I. TRUST-AWARE BUSINESS PROCESS MANAGEMENT

In recent years, many business processes have changed towards a more collaborative nature. Two main drivers of this trend are the rise of multi-sided platforms and the progressing internationalization and digitization of business processes. Large-scale multi-sided platforms enable users who have never interacted before to share goods and engage in business. For instance, such platforms enable users to rent cars of others (BlaBlaCar), share rides (Uber and Lyft), or sublet rooms or apartments (AirBnB) [1]. The platform acts as a connecting intermediary between the business partners. This leads to collaboration with a larger number of actors involved in different process instances. The progressing internationalization and digital transformation of business processes imply a more collaborative nature between different actors working towards a common goal. Examples for this trend reach across different domains like e-commerce [2], supply chain management [3], and the Internet of Things [4].

It is characteristic of such collaborative processes that different organizations execute different parts of a shared process. Usually, the activities carried out by one of the collaborators are beyond the control of the other collaborators [5], [6]. This characteristic causes *uncertainty* regarding process execution. Whenever there is uncertainty in a process, there is a need for trust [7]. Trust represents a positive expectation that certain process parts outside of the own realm of control are executed as desired. Thus, collaborative business processes are especially trust-intensive. This trust-intensiveness of collaborations leads to an increasing academic and professional interest in the design and management of *trust-aware business processes*.

Trust-aware process design [7] is a relatively new sub-field of business process management. As a foundation to design and implement trust-aware collaborations, business process engineers need to understand in detail uncertainties (present in different parts of a process) that lead to process vulnerabilities (impact thereof) and trust dependencies (relationships). Therefore, we propose the concept of *Trust Mining* as an automated approach to analyze uncertainty, process relationships, and how uncertainty impacts them. Based on the trust tolerance profile [8] of different stakeholders in the process, called *trust personas*, Trust Mining produces a reduced set of relevant trust issues. These trust issues need to be mitigated to increase the trustworthiness of the process from the perspective of different trust personas. Thus, Trust Mining aims to answer the following questions:

The associate editor coordinating the review of this manuscript and approving it for publication was Zhangbing Zhou.

- *Where in a process is uncertainty present?* To answer this question, Trust Mining proposes an approach that automatically annotates a process model with relevant uncertainties.
- *Which uncertainty-related dependencies does a process have?* Trust Mining's solution to this is analyzing different process-related relationships that take uncertainties into account.
- *How can uncertainties and trust in a process be illustrated in a meaningful way to process engineers?* For the illustration of trust and uncertainties, Trust Mining introduces different metrics that can be presented in different graphs.
- *What are relevant trust problems in a specific process for different trust personas?* To answer this question, Trust Mining introduces a concept to analyze and filter relevant uncertainties and trust relationships from different perspectives of the involved actors. As an output, Trust Mining creates a list of relevant trust issues that can be mitigated by a process engineer.

The mitigation of trust issues is not a part of Trust Mining. The aim of this paper is to introduce Trust Mining as a purely analytical approach that gives actionable insights that can be used for trust improvements of the process.

Methodologically, this paper follows the design science in information systems research (DSR) paradigm [9]. In information systems, DSR focuses on *IT artifacts* that aim to solve particular problems. IT artifacts may be *constructs* (vocabulary and symbols), *models* (abstractions and representations), *methods* (algorithms and practices), or their *instantiations* (implemented in prototype systems) [9]. The problem that Trust Mining aims to solve is analyzing and measuring trust-related concepts in business process models, as presented in the four main research questions. Thus, Trust Mining is a *method* that consumes a business process model and configuration parameters as an input and delivers insights into uncertainties and trust relationships as an output.

This paper follows the phases in DSR as proposed by Johannesson and Perjons [10]. Every design science research action starts with explicating the problem. Since we focus on trust in inter-organizational business processes, Section II illustrates current research in the relevant fields and emphasizes the problem. In the next phase of DSR, a new artifact is constructed based on requirements. Therefore, Section III introduces Trust Mining as a novel method. Trust Mining was created based on the requirements elicited in the related work section and the beginning of Section III. Currently, there is no automated approach to conceptualize and analyze trust issues in business processes based on their process models. Hence, we present a prototypical implementation of an automated tool to support the application of the Trust Mining method. We call this tool *Trust Studio* and discuss it in Section IV. Afterward, the artifact gets demonstrated and evaluated. Therefore, Section V applies Trust Mining to a set of reference process models. We analyze the created metrics critically and discuss the weaknesses of the concept in Section VI. We conclude this paper with an outlook on future work in Section VII.

This paper aims to introduce Trust Mining as a new method to conceptualize trust in business processes. The evaluation in this paper provides a limited first assessment of the concept. We envision this paper to be a foundation for extensive future research on Trust Mining.

II. BACKGROUND AND RELATED WORK

The notion of trust is a highly informal and abstract concept. In current trust-related literature, no universal agreement on the definition of trust exists [11]. Gambetta established one of the classic definitions of trust that is often used in sociology and other related research fields that states that trust is a positive sentiment towards uncertainty that is out of one's own control [12]. In the context of different actors from different organizations working together, Mayer *et al.* describe trust as the "willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party." [13].

To formalize these various notions of trust into a form that can be handled and analyzed by process engineers, we propose in the following a model-driven approach for trust using business process management concepts. We call this novel approach *Trust Mining*. The next paragraphs give a short overview of selected aspects of business process management relevant to Trust Mining. Business process management is an active research area; hence this related work section can only sketch selected parts. For more holistic introductions to business process management, standard literature can provide more details [5], [6]. After establishing the high-level concepts of business process management, we highlight the basics of collaborative business processes, different perspectives, and trust-aware business process management.

A. GENERAL BUSINESS PROCESS MANAGEMENT

Throughout this paper, we discuss the concept of trust in the context of collaborative business processes. Business processes are one of the key artifacts in the research field of business process management (BPM) [5]. BPM roots in business administration and computer science. Conceptually, a *business process* is a set of activities executed jointly to achieve a specific business goal. BPM "includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes" [5].

Formally, a process model is conceptualized as a graph consisting of nodes and edges. Nodes represent activities, events, or gateways. Activities are the basic building blocks that describe units of work in a business process, while events model states of interest. Start or end events, as well as error events, are some examples. Gateways express control flow structures such as splits or joins. Edges between nodes represent relationships, control flows (within the same organization), or message flows (between different organizations).

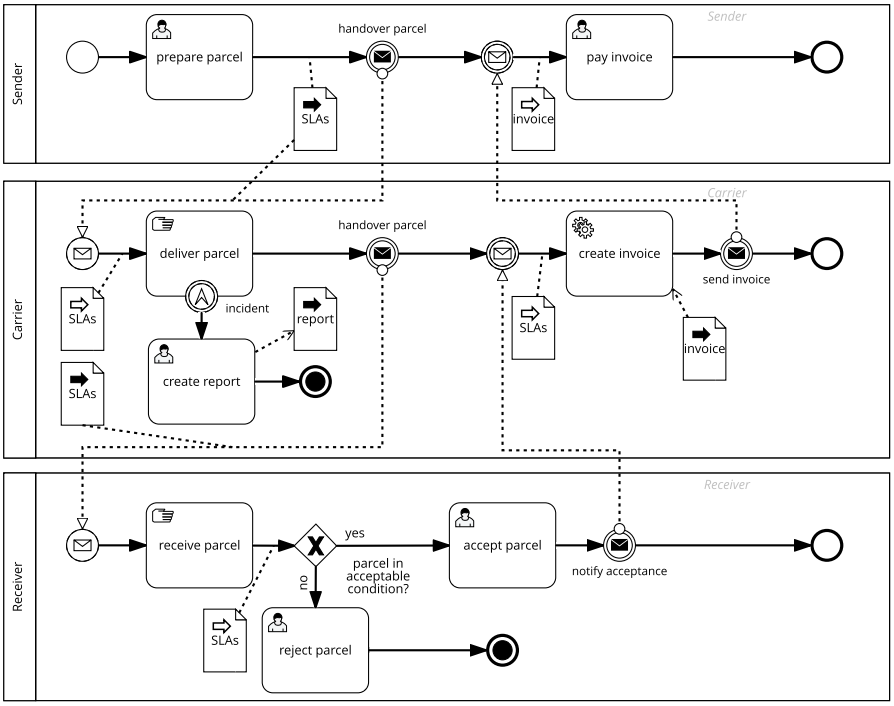

FIGURE 1. Example business process for the delivery of dangerous goods modeled in BPMN.

In addition to these essential process components, process models allow modeling data flows in processes, such as inputs or outputs of activities. In collaborative business processes, different organizations execute separate parts of a shared process. Organizations are often introduced as separate swimlanes in the graph. When a new instance of a process model is created, resources (e.g., people) are assigned to activities.

There are currently many different languages defined to model business processes, including Petri nets, YAWL (Yet Another Workflow Language), workflow nets, or BPMN (Business Process Model and Notation) [14]. Each of these languages has its benefits and challenges, but they all share the same principles of describing a workflow in a formalized way. For better illustration, we employ an example business process from the supply chain management domain throughout this paper. While in general, any business process modeling language can be used for Trust Mining, we utilize BPMN [15] as a graphical example.[1] BPMN has a rich set of components to model inter-organizational collaborations and notations for data objects. These components are needed for certain aspects of the Trust Mining process, for instance, the analysis of data dependencies between collaborators. Figure 1 shows a process model of a dangerous goods delivery process. In the process, a sender wants to send a parcel with fireworks to a receiver. Therefore, the sender uses the delivery services of the carrier, who is responsible for parcel transportation. Every instance of the process starts with the sender preparing the parcel for delivery. During this task, the sender defines *service level agreements* (SLAs) for the delivery. For example, for the delivery of firework rockets, it is crucial to keep them in an anti-static environment to prevent unintended launches. Such requirements are documented in the SLA document and passed on to the carrier. They represent a guideline on how to handle the parcel during delivery. The sender hands the parcel over to the carrier, who then starts to deliver it to its intended destination. If an incident occurs, for example, the carrier acts careless, and the fireworks explode in a postal service truck, the carrier is obliged to create a report. In the case of an incident, the process terminates after that. In case no incident occurs, the carrier arrives at the receiver's locations and hands the parcel over to the receiver. There, the receiver examines the parcel thoroughly to see whether it has taken damage or exploded on the way. If the receiver notices an unacceptable condition, the receiver rejects the parcel. If not, the parcel gets accepted, and the carrier gets notified. Afterward, the carrier creates an invoice and sends it to the sender. The process terminates after the sender paid the invoice.

[1]For simplicity, we define additionally that the process termination element terminates all tokens in all lanes instead of only the current lane as specified in the BPMN standard. Further, throughout this work, we do not distinguish between pools and intra-organizational lanes in BPMN. Therefore, pools are the level of abstraction that represents the separation of subprocesses to different organizations. Further refinement into different units within an organization is not considered. Hence we interpret pools and intra-organizational lanes from BPMN as the concept of swimlanes and refer to them for simplicity as lanes.

TABLE 1. Glossary of the different trust-related concepts used throughout this paper.

Concept	Definition	Example
Uncertainty	Uncertainty describes the unpredictability of an action.	The sender is uncertain about whether or not the carrier handles the parcel carefully enough.
Uncertainty Root	An uncertainty root causes a concrete uncertainty regarding a trust concern.	The truck driver is the root of the uncertainty on whether or not the parcel is handled with care and does not explode.
Trust Concern	Trust concerns describe the subject of uncertainties. Here we use integrity, confidentiality, non-repudiation, availability, performance and resilience.	There is an uncertainty regarding the correct execution (integer execution) of the "deliver parcel" task.
Trust Persona	A trust persona is a stakeholder or collaborator in a business processes. Trust personas have different trust tolerances regarding their trust relationships.	The trust persona "receiver" fully trusts the sender for all activities, but the receiver only trusts the carrier for certain parts of the process.
Process Element	Process elements are the building blocks of business processes and business process models.	The "deliver parcel" task is one process element of the collaborative process.
Process Vulnerability	Process vulnerability describes the impacts different uncertainties can have on a larger part of a process or the process as a whole.	If several uncertainties come true, e.g. the parcel is not handled with care, it explodes and a report is field, this poses a vulnerability to the whole delivery process.
Confidence	Confidence describes the positive expectation towards the desired execution of the process.	The sender is confident that the carrier does handle the parcel carefully enough, because the carrier has a 4.9/5 star review.
Trustworthiness	Trustworthiness describes the extend of actual uncertainty and its impact in a process.	The carrier equips the delivery truck with an anti-static environment for the fireworks parcel. If it explodes anyway, the carrier offers compensation. This increases the trustworthiness.
Trust	Trust means that the subjective assessment of actual and perceived uncertainty (belief) in a process is sufficient for a trustor to engage in the process.	The sender beliefs that the carrier handles the parcel carefully enough.

B. COLLABORATIVE BUSINESS PROCESSES

The idea of inter-organizational business processes was established in the early ages of business process management as a research field [16]. In collaborative business processes, different actors from different companies (or organizations in general) engage in a shared process with a common goal. The common process consists of several (sub-)processes that the organizations execute autonomously. The application areas of inter-organizational collaborations are diverse. For example, in supply chain management, different carriers have to work together to deliver a parcel from a sender to its intended receiver [17], [18]. In finance the transfer of assets between different banks can be seen as a collaborative business process [19], [20]. Nevertheless, also, in non-corporate scenarios, collaborative business processes are omnipresent. For instance, in emergency response scenarios, different emergency response units (e.g., emergency call centers, hospitals, fire brigade) have to collaborate to reach the location of an accident as soon as possible and respond to aid requests [21], [22].

Due to the high relevance of collaborative business processes, current BPM literature provides several approaches to modeling and analyzing such inter-organizational business processes. BPMN [23] supports inter-organizational workflows with the concept of pools (different organizations) and lanes (different departments within the organizations). In Petri nets, different extensions to model collaborations exist. For instance, Zeng *et al.* [24] propose to extend Petri nets with notations for resources and messages, coordinate relations between different organizations, and classify them into different inter-organizational workflow patterns.

Apart from modeling, BPM also enables analysis of inter-organizational collaboration. Therefore, process mining [25] is a widely adopted approach to reverse-engineer business process models from event logs of systems. Process mining is a popular tool that has been implemented for many different process modeling languages [26], [27].

C. PERSPECTIVES IN BUSINESS PROCESS MANAGEMENT

Business process models are, by their design, a flexible approach that inherently has limited semantics. Hence, for expressing different aspects of business processes, a practice called *x-aware business process management* emerged as the common approach to extend the core BPM methodology with *other* objects or phenomena in a wider organizational context [28]. Recker argues that "awareness is generally defined as a state of consciousness in which we perceive and recognize the relevance of a certain object. This means that as individuals, awareness refers to our ability to sense objects and cognitively react to them." [28]. With that, different instances of x-aware BPM have been proposed over time, including but not limited to context-aware BPM [29], risk-aware BPM [30], cost-aware BPM [31], quality-aware BPM [32], privacy-aware BPM [33]–[35], and sustainability-aware BPM [36].

Following the practice of the x-aware BPM pattern, this work proposes to utilize *trust-aware business process management* [7] to extend the traditional BPM with awareness for uncertainty, trust, and related objects. This model-driven approach is our tool of choice to describe the informal concepts of trust in the context of business processes in a structured way to apply Trust Mining as an analysis technique. The following sections give a more detailed overview of trust-aware BPM and its relation to other x-aware BPM fields.

D. TRUST-AWARE BUSINESS PROCESS MANAGEMENT

Trust is a central aspect of collaborative business processes in the digital age. Different research fields have examined trust concerning economic properties [1], [37], [38], from an information systems perspective [39], [40] and regarding

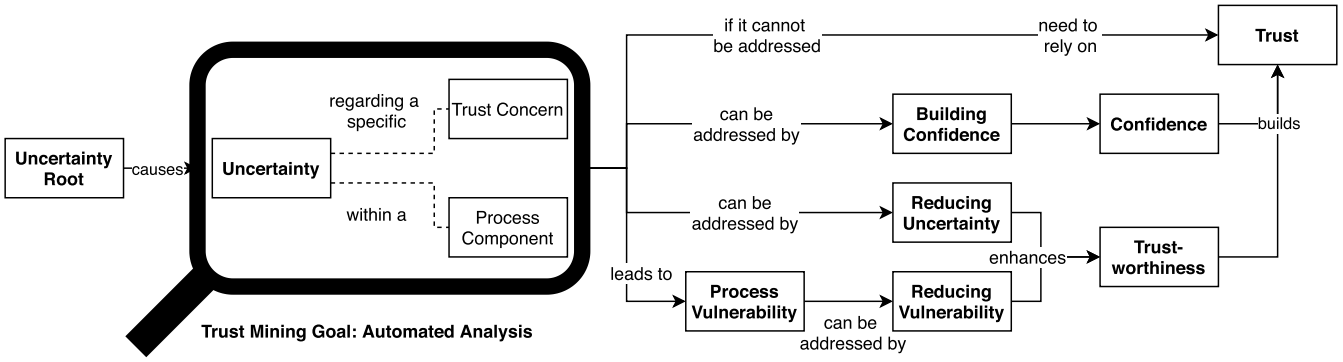

FIGURE 2. A schematic description of trust related concepts in collaborative business processes. Trust Mining addresses the automatic analysis of uncertainties and trust issues.

its implications for the architecture of complex software systems [41], [42]. In general, this paper defines trust according to the sociological definition given by Gambetta: ''When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him'' [12]. This definition reflects how trust relates to expectations and is well-suited in the context of collaborators in business processes. The notion of probability (in terms of chance or possibility) in the definition shows that trust only becomes relevant in a situation when there is *uncertainty* present.

A general meta-model for trust-aware business process design has been introduced by Rosemann [7]. The author has proposed a four-step method for the trust-aware design of business processes. The concept was further enhanced with concepts by Müller *et al.* [43], [44] that added a more fine-granular differentiation of uncertainties in the context of a business process. Figure 2 illustrates the concept that we use throughout this paper. It is based on the mentioned publications. In a business process, an *uncertainty root* causes a specific *uncertainty*. This uncertainty is always specified *regarding a trust concern* and becomes relevant within the scope of a *process component*. A process component is every element in a business process model. For example, in an international delivery process of a parcel, there are different uncertainties present at different process components, as illustrated in the running example in Figure 1. Within one activity, an employee has to deliver a parcel in a post truck to a certain location. There are different uncertainties of relevance within that process component. For instance, the employee (uncertainty root) causes uncertainty regarding integrity (trust concern) of the activity execution (process component). The integrity trust concern within the scope of an activity describes that this activity might not be executed correctly. In the example, this means that there is uncertainty about whether the employee delivers the parcel to the right point and does not break it on the way. The meta-model describes that uncertainty causes *process vulnerability*. Vulnerabilities describe the impact and costs that an uncertainty causes when a specific part of the process does not perform as desired. The presence of uncertainties and vulnerabilities implies the need for trust.

In trust-aware process design, the main goal is to perform actions that *build trust*. Therefore, it is possible either to *reduce uncertainty* of specific process elements, *reduce the vulnerability* to the process once an uncertainty manifests, or to *build confidence* in the process through external sources. Reducing uncertainty focuses on an atomic process element. It aims to reduce the probability *that* the process element is not executed as intended. This implies that reducing uncertainty is a *proactive* approach. In the parcel delivery example, the carrier can decrease the uncertainty that the employee might drive the parcel to the wrong place by equipping the truck with a navigation system. On the other hand, reducing vulnerability is a *reactive* approach that aims to mitigate the impact *when* a part of the process is not performed as planned. For example, reducing vulnerability would mean distributing compensation to a customer when the parcel is delivered to the wrong location and sending it with express delivery to its real destination. The last way to build trust in a process is *building confidence* in it. Therefore, external sources to the process are added to decrease the *perceived* uncertainty. For example, this can be done by adding a reputation system to the delivery process in the running example before the sender decides to use a particular company's parcel delivery services. A reputation system that shows that a particular organization has an average of 4.9/5 stars review from 124 past jobs indicates to a customer that the organization performed well in the past. Hence, it increases the perceived confidence in the process.

Before the existence of the trust-aware process design paradigm, several different approaches to trust management, in general, have been proposed. Mohamaddi *et al.* also leverage business process models together with different concepts for trust-aware requirements engineering. In [45], they introduced a method to identify trust concerns of users of a software system. The identification is made manually by consulting the user upfront regarding their trust concerns. This method can be seen as a top-down approach, starting at the end-user layer. In [46], they have further described how to

use these requirements and illustrate them within a business process. In contrast to this top-down approach, Trust Mining, as proposed in this paper, is a bottom-up approach in which trust is analyzed starting from a business process and adding user requirements at the end of the analysis.

Besides these process-centric concepts, the idea of observing trust in situations where different actors engage with each other has been around for years. Trust management [47], for instance, evolved in the early ages of e-commerce. The main idea of this approach was to have a common system (for example, an online shop) as a foundation, where different trust management activities can be executed on top. Most approaches in the field of trust management focus on the introduction of reputation systems [48]. However, this setup is mostly not applicable to situations where no intermediate platform is utilized. Business-to-business collaborations are often carried out in an ad-hoc setup without any system that could be used as a base for a trust management system, hence making the approach unpractical.

Thus, we argue to observe the trust topic from a process-centric way without the dependency on a platform or a trust management system. Therefore, we build upon the concepts by Rosemann [7] and borrow ideas from other business process management sub-fields that are related but not similar to trust. The main fields, therefore, are quality-aware and risk-aware BPM. The next section gives an overview of the fields. We clearly distinguish the field of trust-aware BPM from risk-aware BPM and quality-aware BPM to clarify the objective of trust-aware BPM and the different views it provides. This essential understanding is needed to truly understand the boundaries and viewpoints that we adapted in Trust Mining.

E. TRUST-AWARE BPM IN THE BPM LANDSCAPE

We introduce Trust Mining as a core analytical activity in trust-aware business process management. To understand its focus and objective, we discuss trust-aware BPM and its borders to related fields. *Trust* has a close relationship to process *quality* and *risk*. The respective BPM sub-fields of trust-aware BPM [7], quality-aware BPM [32], and risk-aware BPM [30] take a different focus of view on a business process as illustrated in Table 2.

Quality focuses on the performance of the process itself or artifacts related to it. One part of the quality view is, for instance, whether the process outcome fulfills the quality requirements of the process stakeholders or not. Quality attributes are partially quantifiable. For example, the reliability of a software system can be expressed with the percentage of system down-time. On the other hand, user-centric quality attributes, such as usability, can only be assessed with fuzzy metrics. This makes the quality view on a process highly subjective. Whether a quality requirement has been met depends on the evaluators if the metric is fuzzy. Quality-aware process management can be executed in all states of the BPM lifecycle. For instance, in the design phase, a process engineer can include quality-assuring activities in the process. Throughout the execution of a process, quality attributes of process artifacts can be monitored.

TABLE 2. Comparison of risk-aware, quality-aware and trust-aware BPM.

	Quality-aware BPM	Risk-aware BPM	Trust-aware BPM
focus of view	Process performance and outcome	Threats and their probability	Uncertainties and their impacts
quantifiable	yes	yes	no
perspective	subjective	objective	subjective
phase in BPM lifecycle	all	all	design phase

Risk deals with threats to a process and its outcomes together with its probability. Risks can be objectively quantified. Therefore, an internal perspective of the process is needed. Attributes related to risks are often related to security and safety. Risk-aware process management assigns a threat with a quantified probability to a process element. Threats can be detected and counteracted. Hence, the risk view on a process can be seen as an analytical tool in the design phase in the BPM lifecycle and as a tool during the execution of a process instance.

Trust is strongly related to quality and risk. The main focus of trust is uncertainties and their impacts from an external view on the process. As introduced by Müller *et al.* in [43], uncertainties are defined regarding specific trust concerns. Some trust concerns are rooted in the field of (information) security, such as integrity, confidentiality, availability, and non-repudiation, which are also analyzed in the context of risk. But also quality attributes, such as performance and process resilience, are trust concerns. Hence, trust-aware BPM has an overlap with quality-aware and risk-aware BPM regarding the objects of observation. However, trust-aware BPM takes up another point of view. While quality can often be seen as a promise from one process collaborator to another, a trust-aware viewpoint focuses on this promise's uncertainty. Trust-aware process management enables the analysis of objects which are not necessarily measurable by one collaborator in contrast to a threat from the risk-aware viewpoint. That risk can be assessed confidently, an internal view on the process details is needed. A trust-centric view does not require that and is thus suited for situations where only assessments from the outside can be made. This is especially common in collaborative business processes. A comparison of the three different concepts can be seen in Table 2.

With trust-aware business process management as a specific view on BPM, trust is a subject mostly in the design phase of a process. Therefore, the design phase needs an analytical part to examine uncertainties and their impacts before the trust-aware implementation of the process.

III. TRUST MINING

In this paper, we introduce Trust Mining to understand uncertainty and its impacts within a business process. Within the BPM lifecycle, Trust Mining takes place at design time. The approach is a base to implement business processes

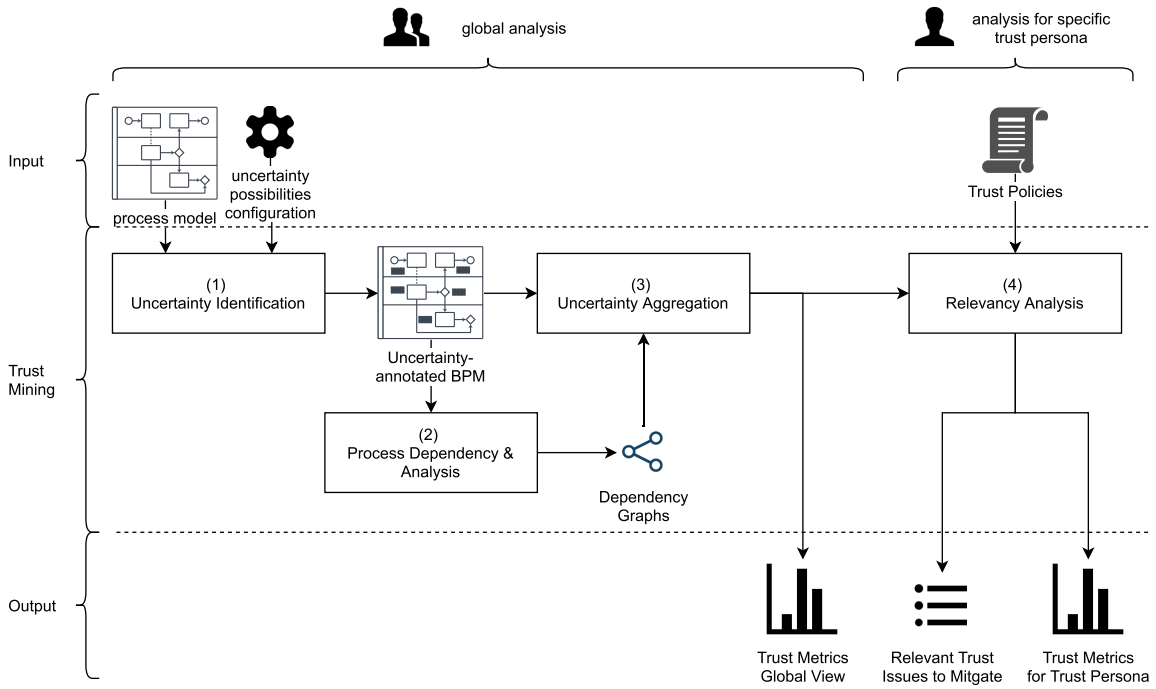

FIGURE 3. The four-step process of Trust Mining: uncertainty identification, process dependency analysis, uncertainty aggregation and relevancy analysis.

complying with the trust tolerance profiles defined through trust policies of certain *trust personas*.

A. OVERVIEW

Trust Mining builds upon the trust model for collaborative business processes as introduced in [7] and [43]. The concept incorporates four steps, see Figure 3. The first three steps comprise a global trust analysis. The last step assesses the relevancy of certain trust issues to specific trust personas. Trust Mining takes a business process model and a set of trust policies associated with trust personas as an input. Additionally, Trust Mining requires a definition of uncertainty possibilities as a case-independent configuration parameter.

In the first step, Trust Mining identifies uncertainties in a business process using the process model and a list of uncertainty possibilities. The configuration defines the uncertainties of interest in a specific process component. The outcome of this step is a business process model with an annotation of uncertainties. The annotation illustrates which uncertainties are potentially relevant at specific parts of the process. Thus, the first step of Trust Mining adds a trust layer to the model, which is used as the input in the later steps.

The process dependency analysis builds upon the uncertainty-annotated process model. In this second step, Trust Mining uses the model to analyze process dependencies between different process components. This analysis uncovers functional and non-functional dependencies. The outcome of Step 2 are message and data dependency graphs that illustrate the relationships between the different organizations regarding uncertainties in message and data exchanges.

We use the uncertainty-annotated process model from Step 1 and the relationship graphs from Step 2 as an input to Step 3. In this step, different trust metrics are calculated to illustrate the uncertainties in a process. Different metrics are either calculated process-wide or as an aggregation depending on the subprocess that the involved organizations are responsible for. The uncertainty aggregation produces trust statistics to quantify uncertainties. The aggregation also shows the uncertainty distribution.

To analyze how trust issues affect trust personas (process stakeholders and collaborators), we perform a relevancy analysis in Step 4. Therefore, trust policies need to be specified per trust persona. These policies define the trust persona's trust tolerance profiles. Step 4 marks the last step in Trust Mining. The final step delivers a list of trust issues as an output for each trust persona. These outputs are used as a base for purely analytical purposes (e.g., convincing someone of the trustworthiness of a process) or for the trust-aware implementation of the business process by mitigating trust issues.

The remainder of this section describes the four steps of Trust Mining in detail.

B. REQUIREMENTS

The creation of Trust Mining as a new method follows the problem-solving research paradigm in DSR. Trust Mining aims to accommodate the following requirements:

1) AUTOMATION

Previous work of trust analysis often followed an open-world assumption. For example, Grandison and Sloman assess trustworthiness by utilizing stakeholder interviews [49].

This approach does not provide any limitations on trust concerns and thus makes identifying trust issues challenging. Moreover, an open-world assumption regarding the trust concerns, process elements, and requirements does not allow for comparability. An open approach without any restrictions on the trust-related aspects leads to the problem that the identification process needs to be executed mostly manually. Manual approaches lead to human errors. Thus, Trust Mining aims to be as automated as possible.

2) SEMANTIC GENERALIZATION

To describe trust, different semantics in business processes influence the overall analysis. Business process models introduce meta-semantics to a process. Activities represent units of work in a process. Yet, the semantics of the work of two activities might be fundamentally different. For example, one activity ''log in to the website'' and another activity ''handover parcel'' have different semantics for the same trust concerns. In the scope of the first activity, confidentiality might be concerned with the user's password security. In contrast, in the second activity, the handover not being visible to competitors or criminals who might want to damage the physical object may be the focus of confidentiality. Trust Mining needs to be semantically general. Yet, it still needs to be compatible with a large number of processes.

3) PROCESS MODEL LANGUAGE INDEPENDENCE

Trust Mining should not be tied to a specific process modeling language. It should enable adaption to different standards such as BPMN or Petri nets.

These three requirements are the largest design challenges for Trust Mining.

C. A META-MODEL FOR PROCESSES

Trust Mining utilizes business process models as a foundation. Many different process modeling languages are being used in academics and the industry. They include BPMN, Petri nets, YAWL, Workflow Nets, and others [14]. Most of these languages have a formal definition. For example, BPMN has an extensive reference documentation of more than 500 pages that specifies the elements of the language [23]. Trust Mining aims to be applicable to process models with certain features and is not built for a specific language. The method requires the following four main capabilities from any process modeling language. A process modeling language needs to be capable of expressing activities, events, and gateways (1). These elements need to be associated with different organizations to express responsibility (2). Data input and data output elements are essential to indicate data flow across organizational borders (3). Finally, any process modeling language that can be used for Trust Mining needs to have the possibility to model control flows and message flows between different parts of the process (4).[2]

With these requirements, we sketch a simple formal notation for a process modeling language that can be used in Trust Mining. The process modeling language introduced in the following can be interpreted as a *meta-language* or *meta-model*. The BPMN specification is compatible with it, but also Petri nets can be applied to it by using different extensions [27]. The meta-model is inspired by similar notations introduced by van der Aalst [14].

A collaborative business process model p in the context of this paper is defined as a 4-tuple of nodes N, edges E, and (swim-) lanes L. Nodes and edges are associated through the function λ with a lane.

$$p = (N, E, L, \lambda) \tag{1}$$

Nodes have a type such as activities, events, gateways (splits and joins), data input, or data output as described previously. We model these types as a set of element type descriptors ET. Concrete process modeling languages have different sets of type descriptors. For the meta-model, we utilize a minimal set of element types and a function τ that assigns every node to one type.

$$\begin{aligned} N &= \{n_0, \ldots, n_k\} \\ ET &= \{\text{activity, start, end,} \\ &\qquad \text{intermediate, split, join,} \\ &\qquad \text{data-in, data-out}\} \\ \tau &: N \to ET \end{aligned} \tag{2}$$

Edges connect the different nodes in the process. They either express sequence or message flows. Sequence flows indicate a workflow executed within one organization, while messages indicate an interaction between actors from different organizations. In general, we define edges as a tuple of nodes and introduce the function η that illustrates whether the edge represents a sequence or message flow.

$$\begin{aligned} E &= \{e_0, \ldots, e_k\} \\ e &= (n_i, n_j), \quad n_i, n_j \in N,\ e \in E \\ \eta &: E \to \{\text{sequence, message}\} \end{aligned} \tag{3}$$

Swimlanes L are a way to model the domain of different organizations in a business process. In the meta-model, we do not further distinguish between pools and lanes as some modeling languages do (e.g., BPMN). We reason that as follows: Weske [5] introduces the concept of swimlanes independent of any concrete language. BPMN specifies pools to depict organizations and intra-organizational lanes as departments within the organization. When we use the term ''lane'' in the following sections we mean swimlanes (the abstract concept and super set of pools and intra-organizational lanes in BPMN) and not the intra-organizational lane in BPMN. Thus, every (swim-)lane is atomic and represents one closed organization. This is done for simplicity, but in general, the model can also be used to accommodate the semantics of pools and lanes as defined in BPMN with extensions. Formally, process nodes and edges get assigned to a lane $l \in L$ with the function λ. In the case of the nodes, the mapping is intuitive in a way that the collaborators that execute the node

[2]These requirements are also aligned with the generic meta-definition of business process models in Section II.

get assigned to it. For the edges, we assign the collaborators at the start of the edge to it.

$$L = \{l_0, \ldots, l_n\}$$
$$\lambda \ : \ N \cup E \rightarrow L \tag{4}$$

The principles of Trust Mining as presented in the following are compatible to any process modeling language that accommodates this meta-model. Graphically, we illustrate the examples with BPMN.

The example business process for the delivery of dangerous goods, as illustrated in Figure 1, can be translated to the meta-model as follows. The example process p^{ex} consists of nodes, edges, lanes, and a function to assign the elements to lanes. Nodes are all activities, events, gateways, and data objects depicted in the diagram, for example, the start event n_{start}, the prepare parcel activity $n_{\text{prepare parcel}}$, or the SLA data output $n_{\text{SLA out}}$.

$$\begin{aligned} p^{ex} &= (N^{ex}, E^{ex}, L^{ex}, \lambda^{ex}) \\ N^{ex} &= \{n_{\text{start}}, n_{\text{prepare parcel}}, \\ &\quad n_{\text{SLA Out}}, \ldots, n_{\text{end}}\} \\ \tau(n_{\text{start}}) &= \text{start} \\ \tau(n_{\text{prepare parcel}}) &= \text{activity} \\ \tau(n_{\text{SLA out}}) &= \text{data-out} \\ &\ldots \\ L^{ex} &= \{\text{sender, carrier, receiver}\} \\ \lambda^{ex}(n_{\text{prepare parcel}}) &= \text{sender} \\ &\ldots \end{aligned} \tag{5}$$

Overall, translating the running example in Figure 1 yields 30 nodes (all tasks, data objects, events, and the gateway) along with 31 edges (20 control flow, 4 message flows) in 3 lanes (sender, carrier, receiver).

D. TRUST MODEL AND CONFIGURATION PARAMETERS

Conceptionally, Trust Mining utilizes a process model and a trust model as a foundation. Therefore, it follows previously established concepts in trust-aware business process management [7], [43]. The model states that *uncertainty roots* cause *uncertainty* regarding a specific *trust concern*. The uncertainty can be related to a *process element* of a business process model. Therefore, a process element can have many uncertainties. Different uncertainties can be related to many different process elements. This creates a many-to-many relationship as illustrated in Figure 4.

In order to analyze uncertainties in a process, Trust Mining needs to semantically understand which uncertainties are possible and of relevance for certain process elements. Therefore, we introduce the *uncertainty possibility list (UPL)*. The UPL can define, for example, that in an activity (process element) that is executed by a manual task of an employee, this employee (uncertainty root) may cause uncertainty regarding the correct execution (i.e., integrity, the trust concern) of the activity. This holds for *all instances of activities*.

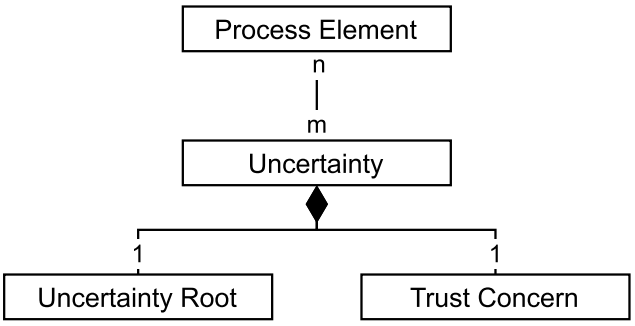

FIGURE 4. Schematic model of the connection of process elements, uncertainties, trust concerns and uncertainty roots.

Formally, we express the UPL as follows. The *UPL* consists of n uncertainty possibilities up. Each of the ups is a triplet of an element type et, a trust concern tc, and an uncertainty root r.

$$\begin{aligned} UPL &= \{up_1, \ldots up_n\}, \quad \forall up \in UPL : \\ up &= (et, tc, r) \\ et &\in ET \\ TC &= \{t_1, \ldots, t_n\} \\ tc &\in TC \\ R &= \{r_1, \ldots, r_n\} \end{aligned} \tag{6}$$

In Trust Mining, any set of trust concerns and uncertainty roots may be used. They can be seen as *configuration parameters* of the utilized trust model. This paper illustrates a reference set of configuration parameters that we advise to use. However, these can be changed according to the process engineer's preferences.

1) UNCERTAINTY ROOTS

Suitable uncertainty roots can be derived by analyzing *who or what can cause uncertainty in a process*. In a collaborative business process, different companies (or *organizations* in general) are executing different parts of a process. The execution of activities in the process can be carried out by an employee (*human resource*) as a manual task. An employee may also utilize an information system (*software*) in a hybrid task. It is also possible that the tasks are fully automated and only rely on software. Therefore, the different organizations share and exchange information (*data*). The coordination (message and sequence flow) is also either executed through software or by human resources. Thus, we argue that organizations, human resources, software, and data comprise a meaningful set of uncertainty roots.

2) TRUST CONCERNS

Following the same principle, trust concerns can also be interpreted as configuration parameters. In current business process management and information systems literature, there is no consensus on a set of universal trust concerns relevant to business processes. In the scope of this paper, we use integrity, confidentiality, non-repudiation, availability, performance, and resilience. We argue that we derived them

from information security, trustworthy systems, and quality of service attributes. In a process collaboration, different organizations share and exchange data with each other. Thus, trust concerns from the domain of *information security* are of relevance. In the field of information security, the CIA-triad is one of the most commonly used sets of properties [50]. The CIA-triad consists of confidentiality, integrity, and availability. The extended ISO 27000 definition for trust concerns regarding information security management systems includes the CIA-triad as well as authenticity, accountability, non-repudiation, and reliability.

Apart from information security, we argue that also the systems themselves are a source of trust concerns. Ross and McEvilley describe in NIST 800-160 an approach to engineering trustworthy secure systems [51]. They introduce safety, security, reliability, dependability, performance, resilience, and survivability as example requirements for a trustworthy system.

In addition to the fields of information security and trustworthy systems, we argue that quality of service attributes are a relevant source of trust concerns for collaborative business processes. In a collaborative business process, different organizations execute autonomous (sub-)processes. One organization may utilize the outcome of a subprocess of another organization. This subprocess can be regarded as a service from the view point of the consuming organization. Hence, the quality of that service might be a source of relevant trust concerns. The Object Management Group (OMG) defines performance, security, integrity, coherence, latency, efficiency, demand, and reliability as quality of service attributes [52].

The goal of the exemplary reference set of trust concerns is to have a minimal yet semantically versatile collection of trust concerns. Hence, we combine and reduce the different sources of trust concerns. For example, latency and efficiency can be subsumed under the term performance. Applying such semantic combinations lets us obtain the reference set consisting of integrity, confidentiality, non-repudiation, availability, performance, and resilience as trust concerns. We neither claim their universality nor their completeness.

3) CREATING THE UPL

With the sets of possible trust concerns, uncertainty roots and process elements identified, the next step is deriving the UPL. This includes the following steps.

For every combination of every element type, trust concerns and uncertainty root, we create a question of the following schema:

- Is it possible that an *uncertainty root* may cause uncertainty regarding the *trust concern* in a *process element*? Example 1: Is it possible that a human resource may cause uncertainty regarding the confidentiality of a manual task? Example 2: Is it possible that an employee may cause uncertainty regarding the integrity of a message transfer?
- Is the created question semantically valid? For Example 1, it is semantically valid that a human resource might, in some instance of a manual task, cause an uncertainty regarding its confidentiality? When an employee of a bookkeeping firm is manually filling out tax sheets, it is possible that the employee could leak that information to unauthorized parties. For Example 2, the question is semantically not valid. The employee is not responsible for the message exchange. Instead, the organization globally coordinates the communication with other organizations.
- If the question is not valid, we terminate and continue with the next combination. If it is valid, we add the uncertainty to the UPL.

If a process engineer always observes the same uncertainty roots, trust concerns, and process elements for all process models under investigation, the UPL creation needs only to be executed once. Otherwise, the recreation of a new UPL is only required if the configuration parameters change.

4) EXAMPLE UPL

For better illustration, this paragraph provides a reference UPL that has been obtained by using the previously suggested uncertainty roots, trust concerns, and process elements. Table 3 shows an overview of the reference UPL. The table also depicts a question that illustrates the uncertainty in addition to the triplet of trust concern, process element, and uncertainty root.

Activity-related uncertainty can mostly be attributed to human resources or software executing an activity. The integrity trust concern is focused on whether an activity is executed correctly. The confidentiality trust concern is centered on the privacy of the activity execution. Availability deals with the uncertainty that all required resources (software or human resources) are available when needed. Non-repudiation is concerned with whether an organization can deny the activity execution. Performance and resilience are uncertainties centered on the consumption of any resource (e.g., computing power or time) and the proper handling of failures during an activity if something undesired occurs.

Event-related uncertainty is similar to activity-related uncertainty. It mostly originates from software or human resources that are tasked with emitting an event. Examples for events include start, end, and intermediate events. While the meta-model does not divide the set of events deeper, in languages like BPMN, many different types of events for communication, compensation, or time-based logic are present. Regarding the meta-model and its generic events, the integrity trust concern caused by software or resources expresses the uncertainty of whether the right event is emitted at the right time. Some events are meant to be concealed within an organization (for example, internal escalation events). Hence confidentiality is another trust concern relevant to events. The availability trust concern illustrates uncertainty regarding the availability of software or human resources which are needed to emit and communicate the event. In inter-organizational

TABLE 3. Classes of uncertainties regarding trust concerns, their roots and questions to ask regarding it.

	Trust Concern	Process Element	Uncertainty Root	Question
activity-related	integrity	activity	software or resource or data	Is the activity executed correctly?
	confidentiality	activity	software or resource or data	Is the internal execution of the activity only visible to authorized resources?
	availability	activity	software or resource	Are the resources, needed for the execution of the activity available?
	non-repudiation	activity	organization	If a certain activity is performed, is it non-repudiable?
	performance	activity	software or resource	Is the activity executed within the needed border of time and resource consumption?
	resilience	activity	software or resource	Can the activity handle the case of failure of one of the involved components?
event-related	integrity	event	software or resource	Are the right events emitted from an activity?
	confidentiality	event	software or resource or data	Are the emitted events only visible to those we are authorized to see them?
	availability	event	software or resource	Are the ways to emit an event available once the event occurs?
	non-repudiation	event	organization	If an event was emitted, can the emitter deny it?
	performance	event	software or resource	Is the evaluation and emission of events executed within the right time and resource consumption constraints?
gateway-related	integrity	gateway	resource	Are the decisions made as desired?
	confidentiality	gateway	software or resource or data	Is the logic how decisions have been made only visible to authorized actors?
	availability	gateway	software or resource or data	Are the tools needed to evaluate a gateway available?
	non-repudiation	gateway	organization	If a certain gateway was evaluated, is it not deniable?
	performance	gateway	software or resource	Is the evaluation of gateways executed within the right time and resource consumption constraints?
process flow-related	integrity	sequence flow	organization	Are activities executed in the right order (as specified)? Is the process only terminating if it is supposed to be terminating?
	integrity	message flow	organization	Is the message flow between collaborators executed as intended? Is the process only terminating if it is supposed to be terminating?
	confidentiality	sequence flow	organization	Is the (internal) sequence flow and all associated data objects only visible to authorized actors?
	confidentiality	message flow	organization	Is the message flow and all associated data objects only visible to authorized actors?
	availability	sequence flow	organization	Is everything needed to coordinate activities intra-organizationally available?
	availability	message flow	organization	Is everything needed to coordinate activities inter-organizationally available?
	performance	sequence flow	organization	Is the flow between activities (intra-organizationally) conducted within the right time and resource consumption constraints?
	performance	message flow	organization	Is the flow between activities (inter-organizationally) conducted within the right time and resource consumption constraints?

workflows, the non-repudiation of events is also fundamental for a successful collaboration. For instance, one organization triggers an escalation event that starts error-handling workflows. The error-handling workflow might also trigger activities at another organization. If the organization later decides to deny that the event occurred, this might lead to inconsistencies and re-execution of certain parts in the collaborative workflow. The performance uncertainty is concerned with the time it takes to trigger an event after it occurred.

Uncertainty in gateways is semantically concerned with whether decisions are made correctly (for exclusive splits and joins) or for parallel behavior (for inclusive splits and joins). Confidentiality of decisions made by software or human resources may be important for competitive reasons. Suppose anybody in the world can see why a logistics company routed a parcel through one hub and not another one. In that case, this enables competitors to reverse-engineer crucial parts of the process. The reasoning for availability, non-repudiation, and performance as trust concerns is similar to activity-related trust concerns. To make decisions in a process, certain human or software resources are needed. If they are not available, the process cannot proceed. Reversing of decisions (i.e., violating non-repudiation) may lead to process inconsistencies with other collaborators. Performance problems (e.g., a decision takes too long) might pose a problem to the timely termination of the process.

For process flow-related uncertainty, we use the same set of uncertainties. Nevertheless, there are different semantics for inter-organizational message flows and intra-organizational sequence flows. The uncertainties are all caused by the organization since different actors (software or human resources) have to coordinate for control flow on an organizational level. Integrity is concerned with the right order of the process flow. Confidentiality is regarded concerning the privacy of the process and coordination within or between companies towards the outside. Availability regards the needed resources to orchestrate the flow. Performance deals with whether or not this orchestration is fast enough.

Within the scope of this paper, we utilize the reference UPL for better illustration. In general, any UPL that has been created as discussed previously can be passed as a configuration parameter to the four main steps of Trust Mining.

E. STEP 1: UNCERTAINTY IDENTIFICATION

Trust Mining's first step deals with the automated identification of uncertainty. Therefore, Step 1 takes the input process model in conjunction with the UPL and automatically annotates the process model with the uncertainty possibilities from the list. Formally, the annotation step yields an annotated process model p_a. This process model is a 5-tuple, containing the process model elements and the uncertainty annotation function α. α assigns every process element to a set of uncertainty possibilities.

$$\begin{aligned} p_a &= (N, E, L, \lambda, \alpha) \\ \alpha &: N \cup E \rightarrow UP \subseteq UPL \end{aligned} \tag{7}$$

The assignment happens in the following way:
For every node $n \in N$ and every edge $e \in E$ of the process:

- Get the type of $\tau(n)$ or ($\eta(e)$ respectively).
- Look up the process elements in the UPL.
- For all up in UPL where $et = \tau(n)$ (or $et = \eta(n)$ respectively): annotate n (or e) with the corresponding up.

Every process element gets assigned all uncertainty possibilities, that are relevant for its element type. In principle, this operation is an iterative traversal of the UPL and the process element with piece-wise comparison for the annotation.

Figure 5 shows the uncertainty annotation for a portion of the delivery example. The delivery parcel activity is a manual activity (marked with the small hand icon in BPMN). Hence, for this activity, a human resource (Res.) causes uncertainty, regarding integrity (Int.), confidentiality (Conf.), availability (Av.), performance (Perf.), and resilience (Resl.). The organization causes uncertainty regarding the non-repudiation, as defined in the reference UPL.

F. STEP 2: DEPENDENCY ANALYSIS

Step 1 of Trust Mining annotated the input model with uncertainties. Step 2 uses the model and analyzes it for cross-organizational dependencies in the process. Process dependencies may be *data dependencies* or *message dependencies*. The outcomes of this step are two dependency graphs. Step 3 utilizes these graphs and the uncertainty-annotated model to aggregate uncertainties in order to create uncertainty metrics and aggregations.

1) DATA DEPENDENCY

The integrity trust concern regarding data flows in processes is essential for the correct execution of activities that are consuming data from another collaborator. If one collaborator executes a task based on a delivered piece of data as an input, this poses an uncertainty. However, in contrast to the local uncertainties identified in Step 1, this uncertainty is not entirely caused by the executing organization. Also, the origin of the data input causes uncertainty *transitively*. Therefore, data dependency introduces a method to analyze such relationships.

In the process meta-model, *data objects* are nodes that are inputs and outputs to elements like activities in

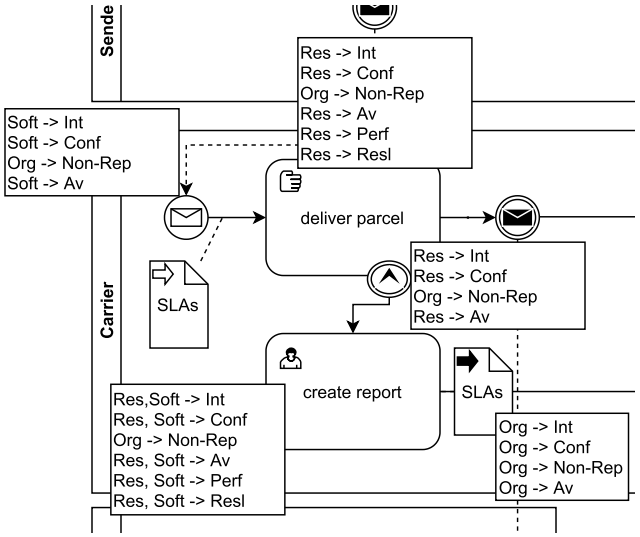

FIGURE 5. Illustration of the annotated model. This diagram shows a small portion of the BPMN diagram of the parcel delivery case in Figure 1 annotated with the reference uncertainty possibilities defined in Table 3. The full annotated graph includes many more annotations; this illustration only depicts a subset of the graph for simplicity.

the process. The BPMN standard adopts the same principle, including input data objects, output data objects, and plain data objects. Other process modeling languages, like Petri nets, for instance, can support data flow modeling and data operations with special markings [53]. Trust Mining requires all data objects to be either input or output objects. The plain data object's semantics is not expressive enough to analyze data dependency relationships. Additionally, we require that the same data objects have the same name, following the previously introduced meta-model.

Data dependency analysis aims to recognize situations where one organization is consuming data that another organization is producing, modifying, or forwarding. In the running example, the receiver obtains the SLA data object from the carrier. The carrier initially receives it from the sender. The sender is the origin of the SLA data object. The receiver depends on the sender to specify the correct SLAs and on the carrier to not tamper with the data object before forwarding it. This situation constitutes a data dependency.

The relationship can be formally modeled and analyzed by building a relationship graph. The data dependency graph p_{dep}^{data} has the lanes L of the business process as nodes and data dependency edges E_{dep}^{data} connecting them. The data dependency graph has an edge (l_i, l_j) for every pair of lanes, where at least one path from a data input to a data output exists for the same data object. We associate the data object that is subject to the data input or data output as $do()$.

$$\begin{aligned} p_{dep}^{data} &= (L, E_{dep}^{data}), \quad \forall e_{dep}^{data} = (l_i, l_j) \in E_{dep}^{data} : \\ &\quad \exists((n_i, n_k), \ldots (n_l, n_j)), \quad n_i, n_k, n_l, n_j \in N : \\ \tau(n_i) &= \text{data-in} \wedge \tau(n_j) = \text{data-out} \\ do(n_i) &= do(n_j) \wedge \lambda(n_i) \neq \lambda(n_j) \end{aligned} \tag{8}$$

Conceptionally, this means the data dependency graph p_{dep}^{data} has an edge where one organization consumes a data

object as an input that another organization produces as an output. We annotate the edges of the graph with the data object descriptor.

To construct the data dependency graph p_{dep}^{data} for an annotated process model p_a, we iterate over all data-in nodes in the process model and search for all data-out nodes of the same data object. When a data dependency is found, we add a new edge to p_{dep}^{data} if the edge is not existing. Additionally, we add the data object the set of dependency objects between two lanes δ_{l_i,l_j}^{data}.

$$\delta_{l_i,l_j}^{data} = \{dataObject_1, \ldots, dataObject_n\} \tag{9}$$

The edges of the data dependency graph can be annotated with the δ^{data} sets, as seen in Figure 6. This figure shows the dependency graph from the delivery example. The carrier consumes the SLA data object as an input, that the sender produced as an output. This creates an edge from the sender to the carrier. The carrier forwards the SLA object to the receiver, creating a data dependency from the receiver to the carrier for the SLA object. After the delivery is concluded, the carrier sends an invoice data object to the sender. This creates a data dependency from the sender to the carrier. In general, one edge can have multiple data objects in its dependency set. Formally, the graph can be described as follows:

$$p_{dep}^{data} = (\{\text{sender, carrier, receiver}\}, \{(\text{sender, carrier}), (\text{receiver, carrier}), (\text{carrier, sender})\}) \tag{10}$$

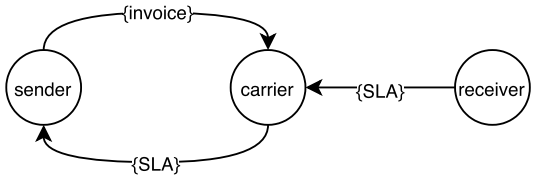

FIGURE 6. Data dependency graph of the running example.

The data markings of the graphs are structured as follows:

$$\begin{aligned} \delta_{\text{sender, carrier}}^{data} &= \{\text{invoice}\} \\ \delta_{\text{receiver, carrier}}^{data} &= \{\text{SLA}\} \\ \delta_{\text{carrier, sender}}^{data} &= \{\text{SLA}\} \end{aligned} \tag{11}$$

2) MESSAGE DEPENDENCY

In addition to data dependency, message exchanges between different collaborators also pose uncertainty. They cannot be identified by analyzing process components in isolation. Thus, we need to analyze relationships to identify message-related uncertainties. In contrast to data dependency, some messages do not have data objects associated with them. Messages may not be triggered as intended, contents associated with the message may be corrupted, or the workflow may not continue as desired.

In the previously introduced meta-model, message flows are modeled through edges. In the running example, the data message flow between carrier and receiver is concerned with the handover of a physical object. Uncertainty regarding the handover message exchange may be concerned with whether the right parcel is transferred. Also, the question if all information included in the message is correct poses a message-related uncertainty.

Formally, the message dependency graph p_{dep}^{msg} has edges E_{dep}^{msg} for every edge in the process model p between two different lanes.

$$\begin{aligned} p_{dep}^{msg} = (L, E_{dep}^{msg}), \quad e_{dep}^{msg} = (l_i, l_j) \in E_{dep}^{msg} : \\ \exists e = (n_i, n_j) \in E : \\ \lambda(n_i) \neq \lambda(n_j) \end{aligned} \tag{12}$$

The visual representation of this graph looks similar to the data dependency graph. The difference is that the edges' labels do not denote the data objects as δ^{data} does. Instead, the labels δ^{msg} denote the carnality of the message flows between the two lanes.

$$\delta_{l_i,l_j}^{msg} = |e = (n_i, n_j) \in E : \lambda(n_i) = l_i, \lambda(n_j) = l_j| \tag{13}$$

In the running example, there is one message flow between sender and carrier, carrier and receiver, receiver and carrier, and carrier and sender. Thus, all edges are annotated with 1. The graph can be seen in Figure 7.

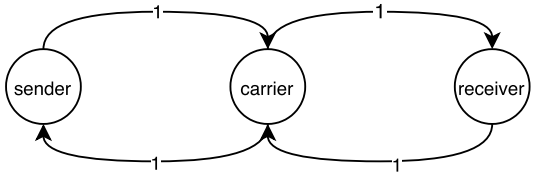

FIGURE 7. Message dependency graph of the running example.

Formally, the message dependency graph of the example can be expressed as follows:

$$\begin{aligned} p_{dep}^{msg} = (\{(\text{sender,receiver,receiver})\}, \\ \{(\text{sender, carrier}), (\text{carrier, receiver}), \\ (\text{receiver, carrier}), (\text{carrier, sender})\}) \end{aligned} \tag{14}$$

G. STEP 3: UNCERTAINTY METRICS

The third step of Trust Mining consumes the uncertainty-annotated business process model from Step 1 and the process dependency graph from Step 2. Step 3 applies aggregation to generate trust-related metrics. The metrics answer the following questions. *How much uncertainty is present in the process? Which collaborator is responsible for which uncertainty? How dependent are collaborators on uncertain process elements executed by other collaborators?* In the following, we discuss different metrics based on the established model and how they can answer the questions.

1) GLOBAL PROCESS UNCERTAINTY

The *global process uncertainty (GU)* is the quantification of uncertainty in the whole process. Therefore, we sum up

the set size of the α-function (uncertainty annotation) of all process elements. Hence, this metric presents the count of all uncertainty possibilities across all process elements. This metric gives a simple overview on how much uncertainty is present in the process. In the running example, GU is 172.

$$GU(p_a) = \sum_{i \in N \cup E} |\alpha(i)| \tag{15}$$

The GU metric shows the overall uncertainty possibilities. The more elements a process has, the more uncertainty possibilities may exist. The exact number varies on the concrete elements used. To enable a better comparability between different processes, we introduce the *average element uncertainty (AEU)*. This metric is the GU divided by the number of process elements. In the running example, AEU is $3.\overline{90}$.

$$AEU(p_a) = \frac{GU(p_a)}{|N \cup E|} \tag{16}$$

2) LANE UNCERTAINTY

The *lane uncertainty (LU)* illustrates how much uncertainty is present within the processes in the collaboration for which certain collaborators are responsible. We distinguish between *absolute lane uncertainty (ALU)* and *relative lane uncertainty (RLU)*. To describe ALU and RLU formally, we introduce a helper function $isLane(i, l)$. This function returns 1 if a certain process element i belongs to a lane l and 0 otherwise.

$$isLane(i, l) = \begin{cases} 1, & \text{if } \lambda(i) = l \\ 0, & \text{otherwise} \end{cases} \quad i \in N \cup E,\ l \in L \tag{17}$$

Subsequently, the ALU is the sum of all uncertainties within the domain of a specific organization.

$$ALU(p_a, l) = \sum_{i \in N \cup E} isLane(i, l) \cdot |\alpha(i)| \tag{18}$$

The RLU is the normalized version of ALU. RLU produces a number between 0 and 1. It is computed dividing ALU and GU.

$$RLU(p_a, l) = \frac{ALU(p_a, l)}{GU(p_a)} \tag{19}$$

In the running example, the sender collaborator has an RLU of 0.25, the carrier has 0.407, and the receiver has 0.343. The metric indicates that the carrier has more uncertainties in her domain of influence than the receiver.

3) UNCERTAINTY BALANCE

The introduced metrics for lane uncertainty give isolated measures on the proportion of uncertainty in the influence domain of different collaborators. The interpretation of these metrics depends strongly on the number of process collaborators. We introduce the concept of *uncertainty balance* to enable the comparison between different process collaborators. If the global uncertainty in a process was equally distributed among $|L|$ collaborators, each of them would be responsible for $1/|L|$ uncertainties. We introduce the *lane uncertainty balance (LUB)* to identify deviations from such a perfectly balanced scenario.

$$LUB(p_a, l) = -\frac{1}{|L|} + RLU(p_a, l) \tag{20}$$

Definition 20 can be interpreted in the following way. In a process with three collaborators, every collaborator would be responsible for $1/3$ of the uncertainties if all uncertainties would be distributed equally. In this case, *LUB* would be 0 for all collaborators. If the balance is off, *LUB* would not be 0. For example, if one lane l is responsible for $2/3$ of all uncertainties, then $LUB(p_a, l) = +1/3$. On the other hand, if one collaborator is responsible for fewer uncertainties, then $LUB(p_a, l) \leq 0$. For example, if one lane l is responsible for $1/6$ of the uncertainties in a process with 3 collaborators, then $LUB(p_a, l) = -1/6$.

4) CROSS-LANE UNCERTAINTY DEPENDENCIES

All previously discussed metrics were focused on atomic uncertainties without the consideration of relationships between actors. In the following, we discuss *cross-lane uncertainty dependency* metrics. These metrics utilize the different process dependency graphs p^{data}_{dep} and p^{msg}_{dep} as a base. Therefore, we use the in-degree and out-degree of nodes. In the directed data dependency graph, an edge (l_i, l_j) expresses that the organization of lane l_i consumes data as an input, which l_i received from l_j. Hence, l_j can tamper with the data, which constructs a data dependency. From a graph-perspective, a situation where many different lanes are consuming data from one specific lane l_j is expressed in a large in-degree deg^{in} of l. This symbolizes that l has a large *data influence (DI)* on other lanes.

$$DI(p^{data}_{dep}, l) = deg^{in}_{p^{data}_{dep}}(l) \tag{21}$$

Equivalently, a large out-degree deg^{out} of a lane l symbolizes that this lane is strongly dependent on data from other collaborators. We call this metric *data dependency (DD)* of a lane.

$$DD(p^{data}_{dep}, l) = deg^{out}_{p^{data}_{dep}}(l) \tag{22}$$

Similar metrics can be applied to the message dependency graphs. With p^{msg}_{dep} as a base, a large in-degree deg^{in} of a lane l means that this lane has a large *message influence (MI)*.

$$MI(p^{msg}_{dep}, l) = deg^{in}_{p^{msg}_{dep}}(l) \tag{23}$$

Equivalently, a large out-degree deg^{out} of a lane l symbolizes that this lane is strongly dependent on messages and associated objects from other collaborators. We call this metric *message dependency (MD)* of a lane.

$$MD(p^{msg}_{dep}, l) = deg^{out}_{p^{msg}_{dep}}(l) \tag{24}$$

In the running example, the carrier has incoming and outgoing message flows with each of the other two collaborators, which results in a message dependency and message influence of two for both metrics ($MD = 2, MI = 2$).

H. STEP 4: RELEVANCY ANALYSIS

Steps 1 to 3 of Trust Mining analyzed trust properties of a business process globally. In Step 4, we compare these properties to the trust tolerance profiles of different *trust personas*. This comparison aims to identify which trust issues are of relevance from the *perspective* of a specific trust persona. Trust personas can be any process stakeholders. They have different trust relationships and preferences with collaborators. Trust personas express their tolerance profiles with *trust policies*.

1) TRUST PERSONAS

From a meta-level, trust analysis follows the schema of observing *who trusts whom for what* [54]. A *trustor* trusts a *trustee* for a certain *trust subject*. A trust persona is every entity interested in the trust properties and the trustworthiness of a business process. In the running example, as seen in Figure 1, all organizations collaborating in the process can be trust personas. The sender and receiver are interested in the trust properties of the process because they want the parcel to be delivered as desired. The carrier is interested in the trust properties of the process because the carrier wants to receive reimbursement for the provided services. In addition, other process stakeholders can be trust personas. For example, government organizations that need to track the supply chains of explosives for regulatory and security purposes can be trust personas. An arbitrary number of trust personas with trust tolerances profiles can be subject to trust analysis for every process.

2) TRUST POLICIES

Trust policies define the trust tolerance profile of a specific trust persona. Policies are based on the trust model proposed by Grandison and Sloman [49] and the homonymous concept in the domain of the *Web of Trust* as defined by Khare and Rifkin [54]. One trust persona is associated with n trust policies, as illustrated in Figure 8. A trust policy defines for a trust persona (trustor), which entities (trustees) the persona trusts for a trust subject. Trust entities are collaborators of the process. The trust subject is composed of one or more process elements and trust concerns. For example, the sender (trust persona) trusts the receiver (trust entity) that the evaluation of the condition of a parcel (gateway, process element) is always done correctly (integrity, trust concern). However, the sender might not trust the carrier that the parcel does not explode.

Trust policies do not distinguish between different levels of trust. Either a trust persona trusts a trust entity for a certain trust subject or not. Hence, when observing on an atomic level, trust is not a probability but a boolean value. This is in contrast to other business process management subfields such as risk-aware or quality-aware business process management. ''Partial'' trust does not exist on the atomic level. However, from an overall perspective, a trust persona may trust a collaborator for one activity, but not for another one. This can be interpreted as partial trust on a process-wide abstraction level.

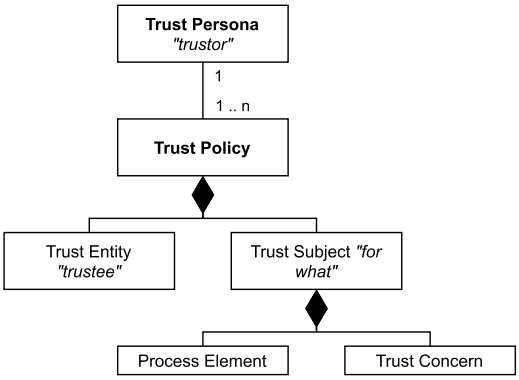

FIGURE 8. Model for trust policies. One trust persona (trustor) has n trust policies. Policies describe a trust entity and a trust subject.

Trust policies represent a positive trust relationship. They express that a trust persona trusts a trust entity fully for a trust subject. If the trust persona does not trust a particular trust entity for a trust subject, there is no trust policy associated with it. Subsequently, every other combination of trust entity and trust subject is not trusted by a trust persona. All trust policies together create the *trust tolerance profile* of a trust persona.

An example of a trust policy definition can be seen in Table 4. The following values can be used as an entry in the respective field of the trust policies:

- Trust Entity: This can be *all* organizations as represented in the lanes in the model or any subset of process collaborators.
- Process Element: This can be *all* elements, classes of process elements, e.g., manual tasks, or specific instances of process elements, e.g., the create invoice script task.
- Trust Concern: This can be *all* trust concerns or a specific subset of the previously defined trust concerns, i.e., integrity, confidentiality, non-repudiation, availability, performance, and resilience.

Formally, we define a trust policy *pol* as a triplet of one or more trust entities, process elements, and trust concerns. All policies of one trust persona together create the trust tolerance profile *TTP* of a trust persona *pers*.

$$\begin{aligned} TTP_p^{pers} &= \{pol_1, \ldots, pol_n\} \quad \forall pol_i \in TTP : \\ pol_i &= (L', ET', TC') \\ L' &\subseteq L \\ ET' &\subseteq ET \cup N \cup E \\ TC' &\subseteq TC \end{aligned} \tag{25}$$

3) TRUST ASSESSMENT

To analyze which trust concerns in the process model are relevant for specific trust personas, Trust Mining performs a *trust assessment*. The trust assessment takes the annotated business process model and *reduces* the possible uncertainties based on the trust policies.

Formally, we define the process model with annotated reduced trust issues p_r^{pers} as an annotated model, which has

TABLE 4. Trust policies of the sender in the running example of the delivery of dangerous goods.

Trust Persona	sender	
Trust Entity	**Trust Subject**	
	Process Element	**Trust Concern**
sender	all	all
receiver	all	all
all	sequence flow	all
all	message flow	all
carrier	create invoice	confidentiality
carrier	handover parcel	all
carrier	send invoice	all

been reduced with a function ϱ according to the trust policies of a trust persona TTP_p^{pers}.

$$p_r^{pers} = \varrho(TTP_p^{pers}, p_a) = (N, E, L, \lambda, \alpha') \tag{26}$$

Therefore, the ϱ-function works for every trust persona *pers* as follows:

- First we start with the annotated process model with all possible uncertainties $p_r^{pers} \leftarrow p_a$.
- For all process elements $ne \in N \cup E$: If there exists a trust policy *pol* in the trust tolerance profile of that persona TTP_p^{pers} that matches the type of the process element, the lane, and the annotated trust concern, we delete it from p_r^{pers}.

With this approach, the reduction iteratively deletes all trust issues that are not relevant to the trust persona.

In the running example, the sender trusts the receiver fully for every part of the process that the receiver is involved in. Hence, the annotated model gets stripped of all uncertainty in that lane from the perspective of the sender trust persona.

The metrics from Step 3 are applied to p_r^{pers}. While in Step 3 the metrics were applied on a macro level, the metrics applied to the reduced uncertainty-annotated model gives a specific trust persona a detailed overview of the *relevant* trust issues from the *persona's point of view*.

I. OUTCOME AND FOLLOWING STEPS

The four steps of Trust Mining aim to analyze trust issues in collaborative business processes. The insights can be utilized in two different ways:

- *Trust issue description (comprehensive)*: The descriptive approach lets process stakeholders *comprehend* the uncertainties in their process. They get detailed insights into the relationships that are present and the issues that these imply.
- *Trust issue improvement (creative)*: This engineering-centered approach lets process engineers take the analysis as a starting point to mitigate trust issues. Thus, Trust Mining can be the starting point for the trust-aware re-engineering of the process.

This paper focuses on the Trust Mining concept itself and not on the following steps. However, in the following paragraph, we give some suggestions and examples of how Trust Mining can be utilized in different scenarios.

Trust Mining can be used in a purely descriptive manner. It lets process stakeholders comprehend to which trust-related situations they commit. For example, Alice wants to start a new e-commerce business. After some market analysis, she considers to sell industrially produced Kombucha worldwide. Therefore, Alice needs a logistics carrier capable of delivering her fermented black tea to her customers. She wants to serve customers worldwide. Alice thinks that customer satisfaction is critical for building a sustainable business. Thus, she decides to use Trust Mining to discover potential trust issues and comprehend to which relationships she needs to commit. With Trust Mining, she is required first to model the process. Alice decides to use the standard configuration parameters for uncertainty roots and trust concerns. The Trust Mining metrics show her that there is an uncertainty imbalance regarding second-level carriers. Also, the correct handling of documents and agreements cause many uncertainties. Alice realizes how trust-intense the process is. A lot of conflict resolution may be required when working with a carrier that does not handle SLAs well. Alice thinks that might be a problem she does not want to deal with. With the insights from Trust Mining, she considers starting an e-commerce business that does not require physical items and carriers to bring them to the customers. In this example, Alice purely utilized Trust Mining to understand what trust constellations she has to position herself in and made a business decision.

Trust Mining can also be utilized as a starting point for the trust-aware process (re-)engineering [55]. The process of trust-aware (re-)engineering can be executed by analyzing all relevant trust issues separately and utilizing *trust patterns* to improve the process. In [44] a taxonomy of different trust patterns is proposed. Thus, it is possible to directly identify in which situations certain trust patterns can be used to mitigate a trust issue. For example, Bob owns an e-commerce store where he sells luxurious vegan sneakers. He wants to expand his business to southern Europe. Therefore, he has to find new partners that will deliver his packages in the new region. With his current partners in central Europe, he has an excellent relationship and trusts them. However, he has no partners in the new region. He wants to make his process as trustless as possible. Thus, Bob decides to use Trust Mining and analyzes the metrics. Bob sees that much uncertainty originates from the correct handling of documents in the process for SLAs. For his business, it is important that the packaging of the sneakers looks pristine when it arrives at the customer's house. Bob decides to alter his process and mitigate the trust concern of a carrier denying the commitment to service level agreements. Therefore, he digitizes the agreement and stores a timestamped hash of the agreement on the Bitcoin blockchain [44]. It is now harder to deny that the SLA exists. Hence, this is not a relevant trust issue for Bob anymore, and he improved his process.

IV. IMPLEMENTATION

We demonstrate the Trust Mining concept through the practical implementation of a Trust Miner called *Trust Studio*.

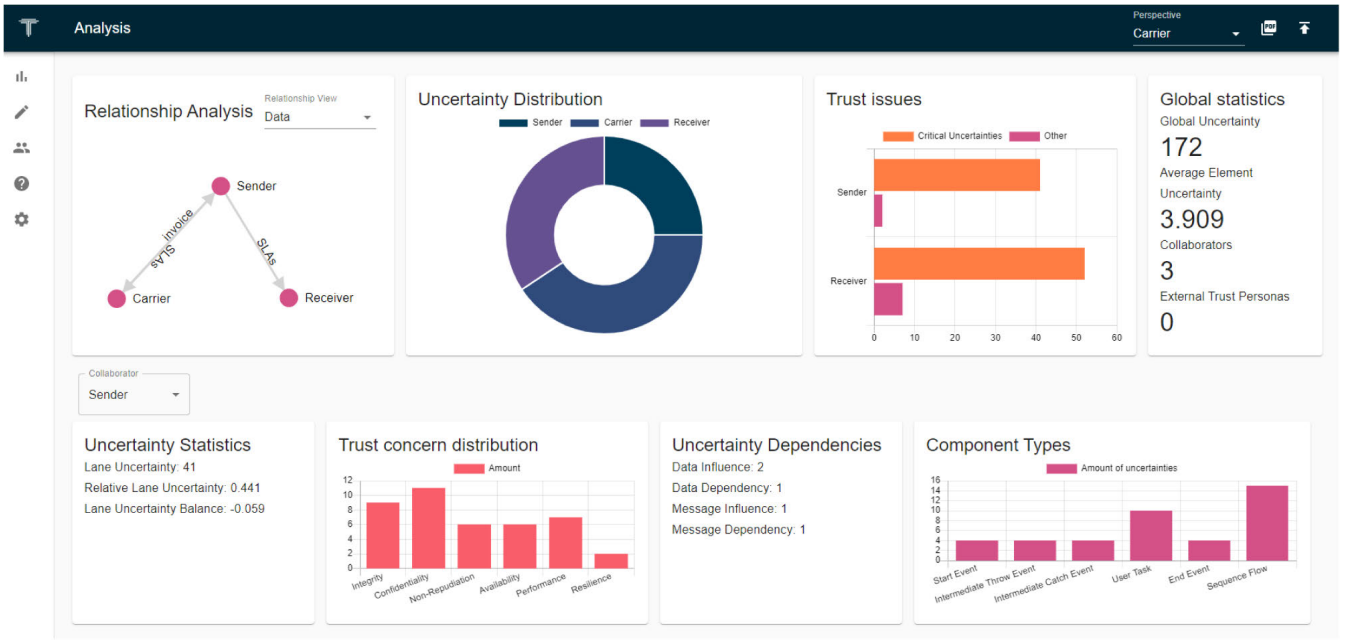

FIGURE 9. Trust Studio's uncertainty dashboard for the running example after its uncertainty annotation.

Trust Studio is a web-based application that executes all Trust Mining tasks. The JavaScript-based application is purely encapsulated in a frontend environment using the React.js framework. It can be run in a browser and does not require a connection to any backend. Trust Studio has a modeling component, in which users can upload an existing BPMN file. Users can also edit their process model or define a new process model using the open-source BPMN.IO library.[3] The software adds the proposed trust layer as custom artifacts to the process model as shown in Figure 5. Users can define the uncertainty possibility list through a form, use a predefined one, or upload a CSV-file of uncertainty possibilities. Users can also define trust policies for different trust personas and generate metrics based on them. The code of Trust Studio can be found on GitHub.[4]

The metrics, as defined in Section III-G, are illustrated using different graphs in an uncertainty dashboard. The dashboard of the example process is depicted in Figure 9. In this dashboard, the user can select from a set of different perspectives that represent the trust personas.

To better understand and analyze the impact of the actual trust concerns in relation to the process and uncertainties, the user is also given the ability to export the "mined" data in the form of a *Trust Report*. This report contains a textual description concerning the general state of the process in relation to uncertainties, followed by each trust persona's perspective.

V. EVALUATION

In this section, we evaluate the feasibility of Trust Mining concerning utility and performance. This approach follows the demonstration and evaluation phases in design science research [9]. Hence, we demonstrate Trust Mining utilizing 137 BPMN models as an input. To evaluate the utility, we analyze and interpret the established metrics on an aggregated level. Conceptual-analytical discussions establish how the metrics help understanding uncertainty and trust issues. Additionally, the computation time of the Trust Mining tasks is analyzed. The computation time can be seen as a reference indicator of Trust Mining's performance.

A. DATA SET

We utilize a set of BPMN collaboration diagrams that originates from three different open-source repositories [56]–[58]. We filter the initial set of diagrams to obtain a subset that is compatible with the meta-model. Therefore, we select only syntactically valid BPMN diagrams. They are required to have at least two different organizations. Organizations need to be modeled as pools. Further, we also perform a semantical analysis to obtain a set of BPMN diagrams that can be considered "meaningful". We consider those BPMN diagrams as meaningful that have no disconnected elements and that have a start and end event for all process flows. Meaningful diagrams are also required not to perform implicit gateways. Additionally, we utilize a set of linting rules as a best practice for meaningful BPMN diagrams.[5]

The data set includes process models from 16 different application domains. Figure 10 shows the distribution of BPMN models and their application domains, which reach from e-commerce to finance, HR, and healthcare processes.

Before analyzing the metrics specific to Trust Mining in a quantitative fashion, we describe the features of the processes

[3]https://github.com/bpmn-io
[4]https://github.com/justdeko/trust-studio
[5]see https://github.com/bpmn-io/bpmnlint/tree/master/docs/rules for more details

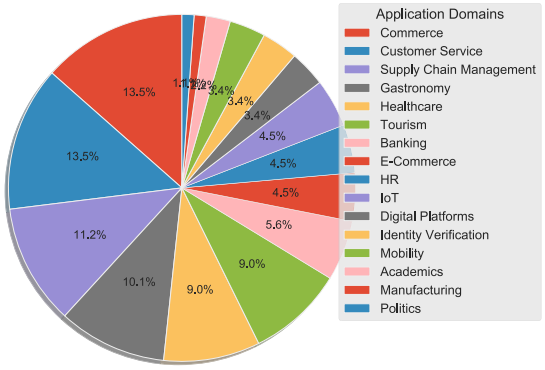

FIGURE 10. Application domains of the 137 test BPMN diagrams for the evaluation of Trust Mining.

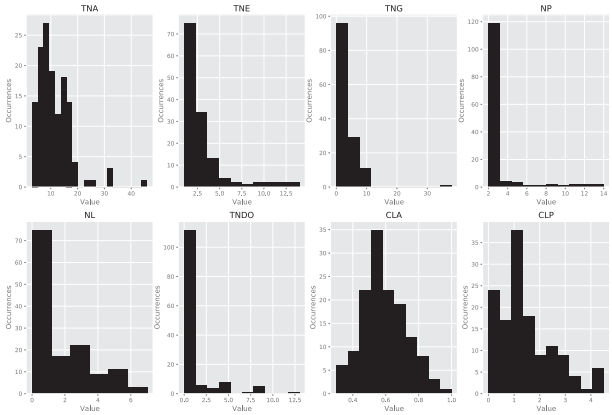

FIGURE 11. Static characteristics of 137 BPMN diagrams utilized for evaluating trust mining. TNA: total number of activities, TNE: total number of event, TNG: total number of gateways, NP: number of pools, NL: number of lanes, TNDO: total number of data objects, CLA: connectivity level between activites, CLP: connectivity level between pools.

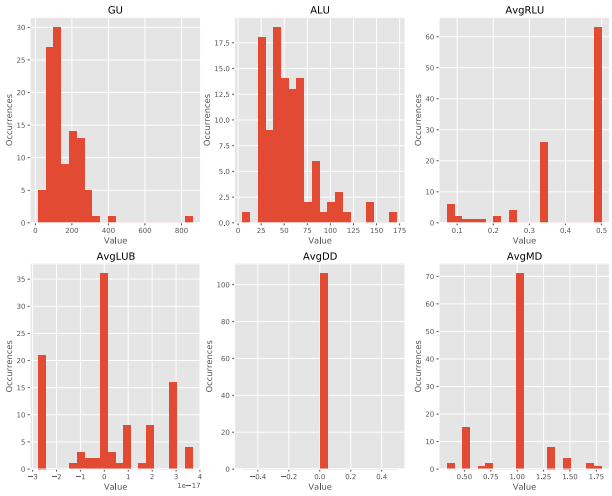

FIGURE 12. TAPE characteristics of 137 BPMN diagrams utilized for evaluating Trust Mining. GU: Global uncertainty, ALU: average lane uncertainty, AvgRLU: average relative lane uncertainty, AvgLUB: average lane uncertainty balance, AvgDD: average data dependency, AvgMD: average message dependency.

according to the metrics proposed in [59]. Figure 11 illustrates in a histogram the distribution of certain feature occurrences across all process models. The utilized diagrams have an average of 11 activities per model, as seen in the upper left corner. The mean total number of gateways (TNG) is 2.74 (min. 0, max. 38) and the mean total number of events (TNE) is 1.05 (min. 0, max. 17). This includes intermediate events, without start and end events. Since Trust Mining is only helpful for inter-organizational processes (assuming everybody trust themselves), we only consider diagrams with at least two organizations. Thus, the mean number of pools is 3.03 (min. 2, max. 14). Most of the pools do not have a further split into different lanes to model the separate intra-organizational units (number of lanes NL, min. 0, mean 1.56, max. 7). As seen in the upper right corner, the diagrams have an average of 0.95 data objects per process (min. 0, max. 13). The two last metrics show the connectivity level of activities (CLA, min. 0.3, max. 1, mean 0.59) and the connectivity level between pools (CLP, min. 0, max. 4.5, mean 1.41).

We argue that this data set represents a significantly large set of application domains. The features show that the prerequisite of meaningful Trust Mining are given. However, most of the data sets do not utilize data objects to model data flow excessively. Thus, the metrics related to the data flow in Trust Mining can only be validated to a limited extent.

B. TRUST MINING METRICS

Figure 12 describes the proposed trust metrics without any trust policies. Trust policies pose the introduction of a specific perspective on trust in a process. Since all of the utilized process models have different organizations, they are not comparable to each other. Thus, we focus on the global perspective of the trust metrics. In the following, we analyze them in a discussion style concerning their explainability and utility. As seen in the upper left corner, the mean global uncertainty is 146 (min: 13, max: 862).

The mean average lane uncertainty (ALU) is 55 (min: 4.67, max: 172.4). The mean number of pools can explain this number. The majority of the 137 test diagrams have between two and three pools. Dividing the mean global uncertainty by the mean number of pools results in 55 average uncertainties per lane. For the mean relative lane uncertainty (RLU), it is visible that most of the processes have an RLU value of close to 0.5 or close to 0.33. The histogram of the average lane uncertainty balance shows a peak at 0. This peak indicates that our data set has characteristics where the uncertainty between the different lanes is mostly balanced. This characteristic is quite evident since we did not apply any trust policies and the lanes are similar in their complexity.

The bottom center histogram shows that most test diagrams have an average data dependency of 0. We can explain this characteristic of the distribution of the trust metrics by the characteristic of the data set. In the input process model data set, data objects are used infrequently. Process models with no data objects always have a data dependency of 0. This rare use of data objects in the evaluation data set is also visible in the TNDO histogram. Hence, interpreting the values of data dependency is only possible to a limited extent.

The average message dependency (MD) on the lower right figure, on the other hand, shows that there is a mean of 1 for message dependency, but also other values occur. This is explainable by the fact that the diagrams often use the request-response message pattern [5]. In this pattern, messages to a single recipient are responded with a single answer.

C. PERFORMANCE

For real-life applicability, it is essential that a process engineer can obtain the outcomes of Trust Mining quickly. In the prototypical implementation, the algorithms relating to the trust analysis are fully implemented in a web application using JavaScript. The following execution time analysis refers to the total time of executing all steps of Trust Mining without utilizing any trust policies.

Figure 13 shows the execution time of annotating the process model and computing all Trust Mining metrics for every diagram from the input set. The test was executed on a machine with a 2.3 GHz Quad-Core Intel Core i5 processor and 8 GB RAM. It is visible that the execution time in general increases linearly with an increasing number of process elements and sequence flows between them. This observation supports our expectations: Trust Mining traverses the process model iteratively several times. For the uncertainty annotation, the process model needs to be traversed once. This leads to an algorithmic complexity of $\mathcal{O}(n)$, where n is the sum of process elements. Regarding the relationship analysis, the data dependency analysis has to compare each data object with every other data object. This leads to a worst-case complexity of $\mathcal{O}(n^2)$.

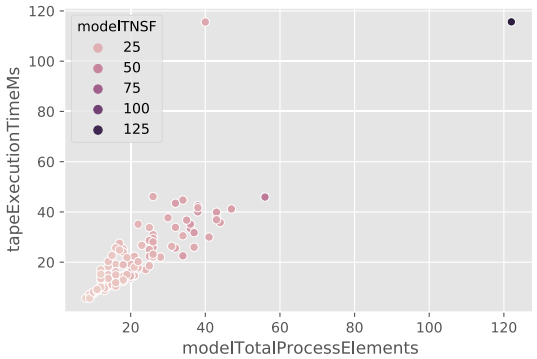

FIGURE 13. TAPE trust analysis execution time scatter plot. The x-axis depicts the number of process elements of the input model, the y-axis shows the execution time. The hue illustrates the number of sequence and message flows in the model.

Observing the execution time of Trust Mining on the 137 BPMN diagrams in Figure 13 shows that the execution time increases with the number of process elements. The scatter plot shows a linear trend, even though there is some variance visible. The linear trend is explainable because data objects are rarely utilized in the evaluation process diagram set. Thus, the computation of the trust metrics does, in most cases, not involve a quadratic component.

Overall, we can conclude that with an execution time between 5 and 40 milliseconds for most of the models, the execution of Trust Mining is nearly instant. Hence, it is fast enough to be a useful tool for trust analysis that does not require the user to wait for long-lasting and expensive computations.

VI. DISCUSSION AND FUTURE WORK

The evaluation in the previous section provided some insights on the interpretability of the metrics and the performance of Trust Mining. For a more conceptual analysis, this section presents a discussion of Trust Mining. The goal of this discussion is to identify some conceptual weak points and outline possibilities for future work.

A. FLEXIBILITY AND EXTENSIBILITY

Trust Mining is designed in a way that it can be configured, modified, and extended to enable different use cases.

Regarding configurability, Trust Mining uses the UPL as a configuration parameter to its first step. The Trust Miner annotates the input process model based on the possibilities defined in the UPL. The list used for the work in this paper is based on [43]. In general, the configuration can be changed to annotate different uncertainties. For example, a user might not want to observe a particular trust concern. In this case, the uncertainty possibilities regarding the particular trust concern may be deleted from the list. The same principle may be applied to extend the UPL. This configurability enables Trust Mining to work with any uncertainty possibilities as long as they are defined according to Definition 6.

It is possible to extend Trust Mining semantically with a *context property*. Within a trust context, it is possible to define labels like ''activity uses standard software'' or ''source code is open-source''. Based on this context, trust policies can be defined that enable a richer semantical analysis. For instance, a trust persona might have a trust policy that standard software or open-source code is always trusted. The specification and extension of the Trust Mining concept are subject to future work.

The four-step approach of Trust Mining can also be extended and modified. This may happen by adding new steps to the core workflow. It is also possible to modify existing steps. For example, in the third step of Trust Mining, new metrics can be added to analyze other trust-related aspects.

B. ABSTRACTION BIAS

Varying levels of abstraction within the subprocesses of different organizations in the collaboration poses a major challenge for Trust Mining. In inter-organizational collaborations, it is common that cooperating organizations have only limited knowledge of the processes of other organizations. In the supply chain example, the activity to deliver the parcel is modeled as one single activity. This portrays an outside perspective onto the processes. In reality, this activity may be a larger subprocess. Different employees may pack, load, and track the parcel within the carrier's

organizational boundaries. The internal processes may be hidden to the external collaborators.

These different levels of abstraction pose a challenge to the usefulness of the uncertainty annotation. If the ''deliver parcel'' activity is modeled as a collapsed subprocess hidden to all other collaborators, the Trust Miner annotates at maximum every uncertainty possibility once to that subprocess. This yields an absolute lane uncertainty of the maximum number of uncertainty possibilities defined for an activity. On the other hand, if the process is modeled so that three activities replace the subprocess, then the Trust Miner annotates each of these activities with possibilities. For a case where the deliver parcel subprocess is further specified into three different activities, this yields an ALU of three times the maximum number of uncertainty possibilities for an activity. In such a situation, also other metrics, for instance, RLU, are increased. Compared to the other collaborators, this might indicate a particular high uncertainty balance towards the organization with more fine-grain modeling of uncertainties. We call this the *abstraction bias*.

The abstraction bias may distort the comparative metrics between the different involved organizations significantly. Trust Mining cannot automatically address the abstraction bias. The most obvious way to mitigate the abstraction bias is to ensure that the process as a whole is modeled on the same level of abstraction. Alternatively, it may also be possible to use BPMN's grouping syntax and group activities together to indicate that they are one ''uncertainty unit'' that needs to be analyzed in an atomic way. In general, many other possibilities that include leaving the process as it is and creating different analysis metrics that are agnostic of the abstraction bias are possible. The mitigation of the bias is outside of the scope of this paper. Future work needs to create and compare different approaches to find the optimal solution for different situations. Without either future work to mitigate the abstraction bias conceptually or ensuring the same abstraction level manually, the abstraction bias poses a threat to the validity of Trust Mining.

C. SYSTEMATIC LIMITATIONS

Trust Mining takes a business process model as an input for trust analysis. That means that Trust Mining cannot analyze aspects of a process that cannot be represented in the model. This circumstance poses a systematic limitation of the approach.

Trust relationships that are established implicitly outside the process concerning functional relationships can also not be analyzed explicitly with Trust Mining. For example, Trust Mining can analyze a situation where one organization depends on data coming from another organization. However, a situation where a trust persona trusts a process collaborator implicitly due to their relationship can hardly be analyzed. An example of this would be when one organization is the subcontractor of another organization. Trust Mining can model this situation by introducing explicit trust policies for that organization from the viewpoint of a certain trust persona. However, it cannot get automatically derived from their implicit relationship. We call this *relayed trust*. Concepts to analyze this apart from the introducing trust policies may be subject to future work.

The utilization of a graph-based process model and traversal of it *piece-wise* limits the focus of observation to only one process element at a time. In case two activities are intended to be performed in parallel, the current version of Trust Mining only analyzes each of them separately. But there might be uncertainty if both activities will be finished before a given deadline, hence having *cross-component uncertainties*. Future work needs to focus on an extension of the initial Trust Mining concept to be able to analyze these dependencies semantically.

In the presented concepts, we do not differentiate between different ''trust weights''. The computation of the quantitative metrics that use uncertainties as an input gives each the same weight of 1. We established this design on purpose to position Trust Mining as an alternative to risk-aware process analysis. In risk-aware BPM, activities are always assigned a risk probability and a risk impact once it happens. The product of these two real numbers can be seen as a metric for the overall risk. For risk-aware BPM to unfold its maximum impact, it is necessary that the process engineer can assess these risk values confidently. But especially in collaborative processes, this might not be possible. In such cases, Trust Mining can be applied as an alternative to the risk-centered perspective. Ultimately, it still may be desired to have very rough notions of different levels of trust. For example, the notion that one trust persona trusts one organization generally ''more'' than another organization might be important. This would still not imply assessing real values for risk and impact but merely give a relative metric. Thus, such trust levels can be beneficial in cases where assessing exact values is not possible. Trust levels are subject to future work.

D. SYNTAX AND SEMANTICS

Trust Mining is a mostly syntactical approach to annotate a model with uncertainties and analyze trust. The semantics of the trust relationships and uncertainties are injected through the UPL and the process engineer's interpretation. For example, the uncertainty regarding the integrity trust concern within an activity can be described with the question, ''is the activity executed correctly?''. In the running example, this can be translated to ''is the parcel delivered correctly?''. In the case of the create invoice script task ''is the invoice created correctly by the system'' would be the semantical equivalent. While both can be described with the most general question, ''is the activity executed correctly?'', the translation to the context of a certain process component may still be semantically non-trivial.

Trust Mining faces this issue to be semantically as general as possible to cover all potential business processes while still being specific enough so that the process engineer can easily derive value from the trust analysis. In the presented version, this also poses an entry barrier for applying Trust

Mining. The way of thinking to understand the trust issues semantically may be non-trivial for many process engineers and analysts. Adding easier to understand semantics to Trust Mining is subject to future work.

E. IMPACT ANALYSIS

Trust Mining, as presented in this paper, has theoretical and practical impacts on the current state of the art. Theoretically, we contributed an in-depth conceptualization of trust in business processes. We add them with a new custom layer to business process models. The new syntax can also be applied to other theoretical analyses of trust in processes. Practically, we envision Trust Mining as a useful tool for process engineers and analysts. The possibility to automate trust analysis can be harnessed to simplify currently complex manual reasoning. While this paper sets the foundation, future work needs to resolve the identified challenges. We see Trust Mining as one fundamental tool in the currently nascent field of trust-aware business process management.

VII. CONCLUSION

This paper outlined Trust Mining as a core analytical tool in trust-aware business process management. Trust Mining analyzes in four steps trust issues in business processes according to the trust tolerance profiles of different trust personas. The outcome of Trust Mining can be used to improve business processes regarding their trust properties. It can also be utilized in a purely illustrative manner to gain confidence in a process by exposing and interpreting the underlying trust relationships.

Trust Mining's evaluation provided evidence regarding its feasibility concerning utility and performance. It showed how the defined metrics can be used to illustrate different trust-related phenomena. Trust personas provide a new way to analyze different perspectives. The performance evaluation has shown that all activities of Trust Mining can be executed in near-instant time for process models of average complexity. Future work in several areas can extend the concept easily. More sophisticated educational approaches are needed to raise awareness about the trust aspect in business processes.

REFERENCES

[1] F. Mazzella, A. Sundararajan, V. B. D'Espous, and M. Möhlmann, "How digital trust powers the sharing economy," *IESE Bus. Rev.*, vol. 26, no. 5, pp. 24–31, 2016.
[2] K. C. Laudon, C. G. Traver, *E-Commerce: Business, Technology, Society*. Boston, MA, USA: Pearson, 2016.
[3] K. Butner, "The smarter supply chain of the future," *Strategy Leadership*, vol. 38, no. 1, pp. 22–31, Jan. 2010.
[4] W. Viriyasitavat, L. Da Xu, Z. Bi, and A. Sapsomboon, "Blockchain-based business process management (BPM) framework for service composition in industry 4.0," *J. Intell. Manuf.*, vol. 31, pp. 1–12, May 2018.
[5] M. Weske, "Business process management architectures," in *Business Process Management*. Berlin, Germany: Springer, 2012, pp. 333–371.
[6] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, *Fundamentals of Business Process Management*, vol. 1. Berlin, Germany: Springer, 2013.
[7] M. Rosemann, "Trust-aware process design," in *Proc. Int. Conf. Bus. Process Manage.* Cham, Switzerland: Springer, 2019, pp. 305–321.
[8] P. Laplante, P. Laplante, B. Amaba, and B. Amaba, "To err is human, to forgive, AI," *IT Prof.*, vol. 21, no. 4, pp. 4–7, Jul. 2019.
[9] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quart.*, vol. 28, pp. 75–105, Mar. 2004.
[10] P. Johannesson and E. Perjons, *Evaluate Artefact*. Cham, Switzerland: Springer, 2014, pp. 137–149, doi: 10.1007/978-3-319-10632-8_9.
[11] S. Castaldo, K. Premazzi, and F. Zerbini, "The meaning(s) of trust. A content analysis on the diverse conceptualizations of trust in scholarly research on business relationships," *J. Bus. Ethics*, vol. 96, no. 4, pp. 657–668, Nov. 2010.
[12] D. Gambetta, "Can we trust trust," *Trust, Making Breaking Cooperat. Relations*, vol. 13, pp. 213–237, Feb. 2000.
[13] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An integrative model of organizational trust," *Acad. Manage. Rev.*, vol. 20, no. 3, pp. 709–734, Jul. 1995.
[14] W. Van Der Aalst, "Data science in action," in *Process Mining*. Berlin, Germany: Springer, 2016, pp. 3–23.
[15] M. Chinosi and A. Trombetta, "BPMN: An introduction to the standard," *Comput. Standards Interfaces*, vol. 34, no. 1, pp. 124–134, Jan. 2012.
[16] W. Van der Aalst, "Loosely coupled interorganizational workflows:: Modeling and analyzing workflows crossing organizational boundaries," *Inf. Manage.*, vol. 37, no. 2, pp. 67–75, 2000.
[17] M. Muller, S. R. Garzon, M. Westerkamp, and Z. A. Lux, "HIDALS: A hybrid IoT-based decentralized application for logistics and supply chain management," in *Proc. IEEE 10th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Oct. 2019, pp. 0802–0808.
[18] M. Müller and S. R. Garzon, "Blockchain-based trusted cross-organizational deliveries of sensor-equipped parcels," in *Proc. Eur. Conf. Parallel Process.* Cham, Switzerland: Springer, 2019, pp. 191–202.
[19] M. Nikolaidou, D. Anagnostopoulos, and A. Tsalgatidou, "Business process modelling and automation in the banking sector: A case study," *Int. J. Simul.*, vol. 2, no. 2, pp. 65–76, 2001.
[20] M. Werner, N. Gehrke, and M. Nuttgens, "Business process mining and reconstruction for financial audits," in *Proc. 45th Hawaii Int. Conf. Syst. Sci.*, Jan. 2012, pp. 5350–5359.
[21] Q. Zeng, C. Liu, H. Duan, and M. Zhou, "Resource conflict checking and resolution controller design for cross-organization emergency response processes," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 10, pp. 3685–3700, Oct. 2020.
[22] H. Duan, C. Liu, Q. Zeng, and M. Zhou, "Refinement-based hierarchical modeling and correctness verification of cross-organization collaborative emergency response processes," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 8, pp. 2845–2859, Aug. 2020.
[23] The Object Management Group (OMG). (Jan. 2014). *Business Process Model and Notation*. https://www.omg.org/spec/BPMN
[24] Q. Zeng, F. Lu, C. Liu, H. Duan, and C. Zhou, "Modeling and verification for cross-department collaborative business processes using extended Petri nets," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 2, pp. 349–362, Feb. 2015.
[25] W. Van Der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, vol. 2. Berlin, Germany: Springer, 2011.
[26] B. F. Van Dongen, A. K. A. de Medeiros, H. Verbeek, A. Weijters, and W. M. van Der Aalst, "The ProM framework: A new era in process mining tool support," in *Proc. Int. Conf. Appl. Theory Petri Nets*. Berlin, Germany: Springer, 2005, pp. 444–454.
[27] Q. Zeng, S. X. Sun, H. Duan, C. Liu, and H. Wang, "Cross-organizational collaborative workflow mining from a multi-source log," *Decis. Support Syst.*, vol. 54, no. 3, pp. 1280–1301, Feb. 2013.
[28] J. C. Recker, "X-aware business process management," *BP Trends*, vol. 8, no. 12, pp. 1–7, 2011.
[29] M. Rosemann and J. C. Recker, "Context-aware process design: Exploring the extrinsic drivers for process flexibility," in *Proc. 18th Int. Conf. Adv. Inf. Syst. Eng., Workshops Doctoral Consortium*. Namur, Belgium: Namur Univ. Press, 2006, pp. 149–158.
[30] S. Suriadi, B. Weiß, A. Winkelmann, A. H. M. ter Hofstede, M. Adams, R. Conforti, C. Fidge, M. La Rosa, C. Ouyang, A. Pika, M. Rosemann, and M. Wynn, "Current research in risk-aware business process management—Overview, comparison, and gap analysis," *Commun. Assoc. Inf. Syst.*, vol. 34, no. 1, p. 52, 2014.
[31] M. T. Wynn, J. De Weerdt, A. H. ter Hofstede, W. M. van der Aalst, H. A. Reijers, M. J. Adams, C. Ouyang, M. Rosemann, and W. Z. Low, "Cost-aware business process management: A research agenda," in *Proc. 24th Australas. Conf. Inf. Syst.*, 2013, pp. 1–10.
[32] M. Heravizadeh, "Quality-aware business process management," Ph.D. dissertation, Fac. Sci. Technol., Queensland Univ. Technol., Brisbane, QLD, Australia, 2009.

[33] V. Diamantopoulou, N. Argyropoulos, C. Kalloniatis, and S. Gritzalis, "Supporting the design of privacy-aware business processes via privacy process patterns," in *Proc. 11th Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, May 2017, pp. 187–198.
[34] C. Liu, H. Duan, Q. Zeng, M. Zhou, F. Lu, and J. Cheng, "Towards comprehensive support for privacy preservation cross-organization business process mining," *IEEE Trans. Services Comput.*, vol. 12, no. 4, pp. 639–653, Jul. 2019.
[35] C. Liu, Q. Zeng, L. Cheng, H. Duan, M. Zhou, and J. Cheng, "Privacy-preserving behavioral correctness verification of cross-organizational workflow with task synchronization patterns," *IEEE Trans. Autom. Sci. Eng.*, early access, Jun. 4, 2020, doi: 10.1109/TASE.2020.2993376.
[36] A. Ghose, K. Hoesch-Klohe, L. Hinsche, and L.-S. Le, "Green business process management: A research agenda," *Australas. J. Inf. Syst.*, vol. 16, no. 2, Mar. 2010. [Online]. Available: https://journal.acs.org.au/index.php/ajis/article/view/597
[37] J. J. Horton and R. J. Zeckhauser, "Owning, using and renting: Some simple economics of the 'sharing economy,'" Nat. Bureau Econ. Res., Cambridge, MA, USA, Tech. Rep., 2016.
[38] A. Sundararajan, *The Sharing Economy: The End of Employment and the Rise of Crowd-Based Capitalism*. Cambridge, MA, USA: MIT Press, 2016.
[39] L. Bernstein, "Trustworthy software systems," *ACM SIGSOFT Softw. Eng. Notes*, vol. 30, no. 1, pp. 4–5, 2005.
[40] F. Chasin, D. M. Riehle, and M. Rosemann, "Trust management—An information systems perspective," in *Proc. Eur. Conf. Inf. Syst.*, 2019, pp. 1-13.
[41] S. Banerjee, C. A. Mattmann, N. Medvidovic, and L. Golubchik, "Leveraging architectural models to inject trust into software systems," in *Proc. Workshop Softw. Eng. Secure Syst.-Building Trustworthy Appl. (SESS)*, vol. 30, 2005, pp. 1–7.
[42] W. Viriyasitavat and A. Martin, "A survey of trust in workflows and relevant contexts," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 3, pp. 911–940, Aug. 2011.
[43] M. Muller, S. R. Garzon, M. Rosemann, and A. Kupper, "Towards trust-aware collaborative business processes: An approach to identify uncertainty," *IEEE Internet Comput.*, vol. 24, no. 6, pp. 17–25, Nov. 2020.
[44] M. Müller, N. Ostern, and M. Rosemann, "Silver bullet for all trust issues? Blockchain-based trust patterns for collaborative business processes," in *Proc. 18th Int. Conf. Bus. Process Manage. (BPM)*, 2020, pp. 3–18.
[45] N. G. Mohammadi and M. Heisel, "Patterns for identification of trust concerns and specification of trustworthiness requirements," in *Proc. 21st Eur. Conf. Pattern Lang. Programs*, Jul. 2016, pp. 1–20.
[46] N. Gol Mohammadi and M. Heisel, "Enhancing business process models with trustworthiness requirements," in *Trust Management X*, S. M. Habib, J. Vassileva, S. Mauw, and M. Mühlhäuser, Eds. Cham, Switzerland: Springer, 2016, pp. 33–51.
[47] S. Ruohomaa and L. Kutvonen, "Trust management survey," in *Proc. Int. Conf. Trust Manage.* Berlin, Germany: Springer, 2005, pp. 77–92.
[48] R. Farmer and B. Glass, *Building Web Reputation Systems*. Newton, MA, USA, O'Reilly Media, 2010.
[49] T. Grandison and M. Sloman, "Specifying and analysing trust for Internet applications," in *Towards the Knowledge Society*. Boston, MA, USA: Springer, 2003, pp. 145–157.
[50] *International Standard ISO/IEC Information Technology—Security Techniques—Information Security Management Systems*, Standard ISO/IEC TR 27000, 2018.
[51] R. Ross, M. McEvilley, and J. Oren, "Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems," NIST, Gaithersburg, MD, USA, Tech. Rep., 2016. [Online]. Available: https://www.nist.gov/publications/systems-security-engineering -considerations-multidisciplinary-approach-engineering-0
[52] The Object Management Group (OMG). (Apr. 2008). *UMLTM Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms Specification*. Accessed: Jul. 1, 2021. [Online]. Available: https://www.omg.org/spec/QFTP/1.1/PDF
[53] D. Xiang, G. Liu, C. Yan, and C. Jiang, "Detecting data-flow errors based on Petri nets with data operations," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 251–260, Jan. 2018.
[54] R. Khare and A. Rifkin, "Weaving a Web of trust," *World Wide Web J.*, vol. 2, no. 3, pp. 77–112, 1997.
[55] M. Müller, N. Ostern, S. Rodriguez Garzon, and A. Küpper, *Trust Models for Next-Generation Blockchain Ecosystems*. Basel, Switzerland: EAI/Springer, 2021.
[56] L. Pan and C. Li. (2020). *6219 Pairs of BPMN Images and Definition Files*. [Online]. Available: https://dx.doi.org/10.21227/yzvb-0f30
[57] G. website. *Model Repository—Browse UML, BPMN, Database (RDS) and Flowchart Examples*. Accessed: Jan. 12, 2020. [Online]. Available: https://app.genmymodel.com/api/repository
[58] F. Corradini, F. Fornari, A. Polini, B. Re, and F. Tiezzi, "RePROSitory: A repository platform for sharing business PROcess modelS," *BPM (PhD/Demos)*, vol. 2420, pp. 149–153, Sep. 2019.
[59] E. Rolón, F. Ruiz, F. García, and M. Piattini, "Applying software metrics to evaluate business process models," *CLEI Electron. J.*, vol. 9, no. 1, Jun. 2006. [Online]. Available: https://www.mendeley.com/catalogue/6b67ebae-fa8c-34e7-adf8-7b66761d16ee/

MARCEL MÜLLER received the M.Sc. degree in computer science from TU Berlin, in 2019. He is currently researching with the Telekom Innovation Laboratories, Technische Universität Berlin, Berlin, Germany. His current research interests include business process management, blockchain and distributed ledger technologies, software engineering, and machine learning.

NADINE OSTERN is currently a Deputy Professor of digitization and process management with Philipps-University Marburg. Her research interest includes intersection of novel technologies and process management, with a special interest in harnessing the potentials of digital technologies for explorative process redesign.

DENIS KOLJADA is currently pursuing the B.Sc. degree in business informatics at Technische Universität Berlin, Germany. His current research interests include business process management, decentralized networks, machine learning, and recommendation systems.

KAI GRUNERT is currently a Researcher with the Telekom Innovation Laboratories, Technische Universität Berlin, Berlin, Germany. His research interests include process automation and modeling, distributed systems, and Web technologies.

MICHAEL ROSEMANN is currently the Director of the Centre for Future Enterprise, Queensland University of Technology, Brisbane, Australia. His research interests include management of business processes, trust, and innovation.

AXEL KÜPPER is currently a Professor of service-centric networking with the Telekom Innovation Laboratories, Technische Universität Berlin, Berlin, Germany. His current research interests include related to decentralized systems and applications, mobile computing, cloud computing, privacy, and future Web technologies.