

Modellierungsumgebung zur Erzeugung synthetischer Ereignisprotokolle für das Process Mining

Yorck Zisgen¹ Dominik Janssen² Christian Imenkamp Agnes Koschmider³

Abstract: Process Mining hat in verschiedenen Bereichen einen erheblichen praktischen Nutzen erlangt. Der Ausgangspunkt von Process Mining ist ein Ereignisprotokoll, das die Ausführung von Aktivitäten beinhaltet, die einem Geschäftsprozess zugeordnet werden können. Somit hat die Qualität von Ereignisprotokollen erhebliche Auswirkungen auf das Process Mining-Ergebnis. Die Verwendung von Process Mining in neuartigen Anwendungsszenarien oder experimentellen Umgebungen scheitert aktuell daran, weil kaum geeignete Ereignisprotokolle öffentlich zugänglich sind. Dieser Beitrag stellt ein Werkzeug zum Generieren synthetischer (Sensor-)Ereignisprotokolle vor. Im Vergleich zu bestehenden synthetischen Protokollgenerator-Werkzeugen erzeugt der IoT-Prozessprotokollgenerator Daten auf nicht deterministische Weise. Benutzer können Rauschen kontrolliert hinzufügen und Prozesse mit IoT-Daten erweitern. Auf diese Weise ermöglicht das Werkzeug die Generierung synthetischer Daten für IoT-Umgebungen, die individuell konfiguriert werden können. Somit leistet unser Werkzeug einen Beitrag zu einem verstärkten Einsatz von Process Mining in Umgebungen, die auf (IoT-) Sensorereignisdaten angewiesen sind.

Keywords: Prozessmodellierung; Internet of Things; Ereignisprotokoll; Synthetische Daten; Geschäftsprozess-Simulation; Process Mining.

1 Einleitung

Process Mining und Internet-of-Things (IoT) sind zwei Gebiete, die sich sehr gut ergänzen. Denn IoT-Umgebungen produzieren große Datenmengen, die die Methoden des Process Mining für eine gute Prozessanalyse benötigen [Ja20]. Process Mining wiederum erlaubt die effiziente Analyse dieser Daten und ermöglicht Einblicke in IoT-gestützte Prozesse. Allerdings weisen aufgezeichnete IoT-Daten eine Herausforderung im Hinblick auf Datenqualität auf. Durch das Fehlen von Daten, oder eine unvollständige Aufzeichnung wird die direkte Anwendbarkeit von Process Mining Methoden erschwert. Semantisch gesehen sind Sensorereignisdaten auf einem viel niedrigeren Niveau als Ereignisdaten, die üblicherweise auf einem wesentlich höheren Abstraktionsniveau aufgezeichnet werden. Somit können Sensorereignisdaten nicht direkt von Process Mining Werkzeugen verarbeitet werden.

¹ yzi@informatik.uni-kiel.de

² dominik.janssen@informatik.uni-kiel.de

³ ak@informatik.uni-kiel.de

Christian-Albrechts-Universität zu Kiel, AG Process Analytics, Institut für Informatik,
Hermann-Rodewald-Str. 3, Kiel

In diesem Beitrag stellen wir den *IoT-Prozessprotokollgenerator* vor, der es erlaubt Ereignisprotokolle aus simulierten IoT-Umgebungen in verschiedenen Qualitätsstufen für Process Mining Verfahren zu erstellen. Das Werkzeug kann sowohl einfache Ereignisprotokolle ohne IoT Bezug generieren, als auch IoT-Sensorereignisprotokolle wie sie für IoT-Umgebungen typisch sind. Um den Erstellungsprozess aus Anwendersicht zu vereinfachen, wurde die Simulations-Engine mit einer benutzerfreundlichen grafischen Oberfläche entwickelt. Benutzer können Prozesse grafisch modellieren und die Aktivitäten im Modell im Bezug auf Dauer und Häufigkeit konfigurieren. Zusätzlich können verschiedene Arten von Rauschen in beliebiger Menge dem Ereignisprotokoll hinzugefügt werden. Optional kann eine IoT-Umgebung mit verschiedenen Arten von IoT-Sensoren dem Prozessmodell zugeordnet werden. Die hieraus resultierenden Ereignisprotokolle können schließlich für verschiedenste IoT-Umgebungen, mit unterschiedlichen Sensor- und Fehlertypen erstellt werden und für experimentelle Zwecke verwendet werden. So können zum Beispiel Bewegungssensoren mit diskreten (d.h., EIN und AUS Zuständen) simuliert werden, aber auch Temperatursensoren mit stetigen Sensorwerten. Eine solche Simulation kann beispielsweise die Frage beantworten, ob das Ausstatten einer Umgebung mit IoT-Sensoren überhaupt empfehlenswert ist. Engpässe in Produktionskapazitäten lassen sich durch das Aufzeigen von Überlastungen in der Simulation aufdecken. Es wurde bereits gezeigt, dass synthetisch generierte Daten nicht nur ein valider Ersatz für echte Daten sind [Ch19, PWV16], sondern auch einen Einblick in domänenspezifische Forschung geben kann [Tr18]. Wir sind davon überzeugt, dass unser IoT-Prozessprotokollgenerator die Anwendung von Process Mining Methoden in Fällen erleichtert, in der es erschwerten Zugang zu Daten gibt, oder die Datenqualität die Aussagekraft der Analyse einschränkt.

Der Beitrag ist folgendermaßen strukturiert: Kapitel 2 fasst die verwandten Arbeiten zusammen. Kapitel 3 stellt die allgemeine Struktur unseres Werkzeuges vor. In Kapitel 4 wird die Implementierung des Werkzeuges vorgestellt. Kapitel 5 demonstriert die Nützlichkeit unseres Werkzeuges am Beispiel von zwei Anwendungsfällen. Der Beitrag endet mit einer Zusammenfassung und einem Ausblick auf zukünftige Forschungsfragen.

2 Verwandte Arbeiten

Die folgenden Forschungsbereiche sind für unseren IoT-Prozessprotokollgenerator relevant: (1) IoT-Log-Generatoren und (2) Ereignisprotokoll-Generatoren.

Im Hinblick auf Kategorie 1) beschränken sich die verfügbaren Arbeiten bzw. Werkzeuge auf einen einzigen Sensortyp [Gi13, Pa13], eine spezielle Anwendungsdomäne [Ah19, Pa13], oder sie stellen eine IoT-Simulation nur konzeptuell vor [AMK18, Ch18, SSN17]. Zum Beispiel ist der Ansatz aus [Gi13, Pa13] beschränkt auf GPS Sensoren und deren Signalstärke und erlaubt keine weiteren Arten von Sensoren. Unser IoT-Prozessprotokollgenerator hingegen erlaubt das Hinzufügen von zusätzlichen Sensortypen, wie zum Beispiel An/Aus-, Bewegungs-, Licht-, oder Temperatursensoren.

IoT-Ereignisprotokoll-Generatoren wurden bereits auf verschiedene Anwendungsdomänen angewendet, zum Beispiel auf Mobilgeräte, Drahtlosnetzwerke oder cyberphysische Systeme. Kertesz et al. [KPG19], zum Beispiel, stellen einen Simulator für die Cloud-Kommunikation von mobilen IoT-Geräten generierten Sensordaten vor. Papadopoulos et al. [Pa13] beschäftigen sich mit der Signalstärke von Drahtlosnetzwerken. Ramprasad et al. [Ra19] schlagen einen Simulator für virtuelle IoT-Architekturen (EMU-IoT) vor, der eine Ende-zu-Ende Evaluation eines IoT Netzwerkes simulieren kann. Giménez et al. [Gi13] testen die Veränderung von Positionsdaten in Lagerhäusern, um Kollisionsszenarien zu testen. Ahmad et al. [Ah19] stellen eine Simulationsarchitektur vor, die sich mit der Kommunikation in Echtzeit IoT-Umgebungen befasst. Verfügbare IoT-Simulatoren sind demnach üblicherweise auf ein sehr enges Anwendungsfeld beschränkt. Eine Übertragbarkeit auf andere Domänen ist nicht vorgesehen. Unser IoT-Prozessprotokollgenerator erlaubt jedoch Simulationen in variablen Umgebungen, zum Beispiel Smart-Homes, Smart-Factories oder etwa Krankenhausumgebungen.

Grundsätzlich können synthetische Ereignisprotokolle mit den gängigen Process Mining Werkzeugen wie CPN tool [JKW07], ProM [Va05] oder WoPed [EF08] generiert werden. So wurde auch in [Ci15] ein Log-Generator vorgestellt, der deklarative Prozessmodelle erstellen kann. Diese Werkzeuge erstellen jedoch deterministische Ereignisprotokolle die direkt aus dem Prozessmodell erstellt werden (d.h. es können keine Häufigkeiten oder Wahrscheinlichkeiten für bestimmte Traces spezifiziert werden). ProM erlaubt es zwar, dem Ereignisprotokoll Rauschen hinzuzufügen, eine Analyse der verfügbaren Werkzeuge zum Filtern von Rauschen in Ereignisprotokollen hat jedoch gezeigt, dass die Werkzeuge weder in der Lage waren, Rauschen dem Ereignisprotokoll adäquat hinzuzufügen, noch angemessen das Rauschen herauszufiltern [Ko21]. Unser IoT-Prozessprotokollgenerator hingegen erlaubt es, unterschiedliche Arten und Häufigkeit von Rauschen dem Ereignisprotokoll hinzuzufügen und gleichzeitig ein rauschfreies Ereignisprotokoll zur Verfügung zu stellen.

3 Implementierung

Das in diesem Beitrag vorgestellte Werkzeug lässt sich über einen Webbrowser erreichen. Benutzer können Geschäftsprozesse als Petri-Netze grafisch modellieren oder in Form von .PNML-Dateien importieren. Zusätzlich lassen sich Simulationseinstellungen, wie die Dauer von Aktivitäten, den simulierten Zeitraum oder verschiedene Arten und die Häufigkeit von Fehlern (*Noise*) festlegen. Darüber hinaus lässt sich mit dem IoT-Umgebungsmodellierungswerkzeug die betrachtete Umgebung (Fabrik, Wohnhaus, Stadt) mit ihren vorhandene Sensoren und deren Ausfallwahrscheinlichkeiten per Drag-and-Drop modellieren. Hieraus ergeben sich in der Simulation zwei unterschiedliche Arten von Ereignisprotokollen. Zum einen ein Ereignisprotokoll mit Zeiten und Aktivitäten, wie es üblicherweise im Process Mining betrachtet wird, zum anderen ein Sensorereignislog mit Sensorwerten. Die Simulationskomponente erwartet lediglich als Eingabeparameter ein Prozessmodell, die modellierte IoT-Umgebung und Simulationseinstellungen. Hierbei wird

die Annahme gemacht, dass jeder Prozessschritt an einem oder mehreren möglichen Orten stattfindet. Aus der Verknüpfung der Modelle generiert die Anwendungsschicht ein Ereignisprotokoll, welches online betrachtet oder als .TXT- oder .CSV-Datei heruntergeladen werden kann.

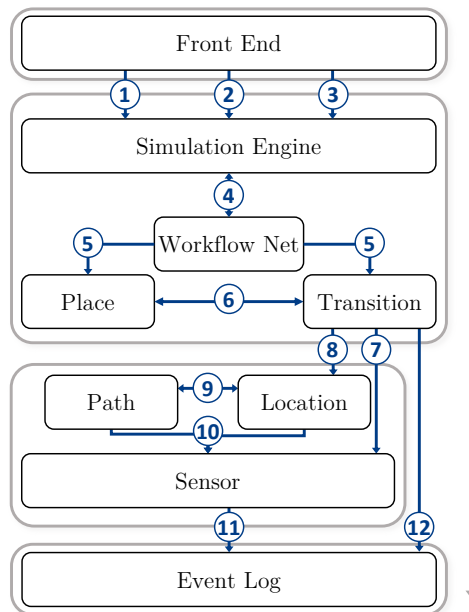


Abb. 1: Von der Modellierung über die Simulation zum Ereignisprotokoll

Zunächst werden die modellierten Geschäftsprozesse, die Simulationseinstellungen und die Wahl des Ausgabeformats an die Simulationsengine übermittelt (Abb. 1, Markierungen 1, 2, 3). Die Simulationsengine erzeugt anschließend Prozessinstanzen aus den Geschäftsprozessmodellen (4). Die Prozessinstanzen (Workflow-Netze bestehend aus Transitionen und Stellen) werden unabhängig voneinander verarbeitet. Die Simulation von Geschäftsprozessen folgt hierbei den Regeln für Petri-Netze (5, 6). Transitionen repräsentieren Aktivitäten, welche Orten in der IoT-Umgebung zugeordnet sind (8) und lösen Sensoren aus (7). Bewegungen innerhalb der IoT-Umgebung (9) können ebenfalls zu Sensorauslösungen führen (10). Sensormesswerte (diskrete, stetige oder zustandsbasierte Werte) werden gemäß den IoT-Einstellungen generiert und zusammen mit Zeitstempel und optionalem Rauschen an das Sensorereignislog übermittelt (11). Soll nur ein Ereignisprotokoll erzeugt werden, können Sensoren und IoT-Umgebung ausgelassen und die Prozessausführungen direkt ausgegeben werden (12).

Unser Werkzeug unterscheidet sich damit von Verarbeitungsskripten durch die Vermeidung deterministischer Vorgehensweisen, beispielsweise bei der Wahl, Ausführung und Reihenfolge modellierter alternativer oder paralleler Prozessschritte. Darüber hinaus werden

Aktivitätendauer und Sensorwerte zufällig gemäß angegebener Verteilung innerhalb eines modellierten Intervalls gesetzt. Daraus resultiert, dass alle möglichen Prozessausführungssequenzen im Ereignisprotokoll beobachtet werden können. Sensorauslösungen können zu unterschiedlichen Zeitpunkten innerhalb der Dauer einer Aktivität stattfinden, sowohl in einer vorgegebenen als auch in einer randomisierten Reihenfolge. Zusätzlich erlaubt unser Werkzeug die Definition von 'Regelfällen' und 'Ausnahmefällen' mittels unterschiedlicher Gewichtung von Wahrscheinlichkeiten an Kanten im Prozessmodell.

4 Modellierung und Simulation

Im Gegensatz zu anderen Werkzeugen zur Geschäftsprozessmodellierung beschränkt sich unser Werkzeug nicht nur auf die Modellierung, sondern ermöglicht, Prozesse benutzerorientiert zu konfigurieren. Somit lassen sich verschiedene Varianten von Ereignisprotokollen erzeugen. Abb. 2 zeigt einen Ausschnitt aus unserem Petri-Netz Editor. Stellen, Transitionen und Marken können per Drag-and-Drop bearbeitet werden. Die hier erstellten Petri-Netze bilden die Grundlage für die spätere Simulation und Erstellung des Ereignisprotokolls.

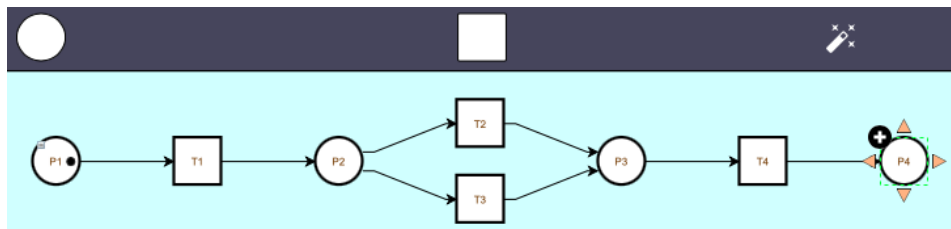


Abb. 2: Oberfläche des Petri-Netz Editors

Durch weitere Benutzereingaben können die zeitliche Dauer von Prozessschritten oder gewichtete Pfadverzweigungen im Prozessmodell definiert werden. Um neben einem Ereignisprotokoll auch ein Sensorereignisprotokoll erstellen zu können, müssen hierfür Sensoren und das IoT-Umfeld modelliert werden. In Abb. 3 ist eine solche IoT-Umgebung dargestellt. Die Ovale stehen für Orte bzw. Geräte und Maschinen an bestimmten Orten in der IoT-Umgebung, die Dreiecke sind verschiedene Sensortypen die mit den betreffenden Orten in Verbindung stehen.

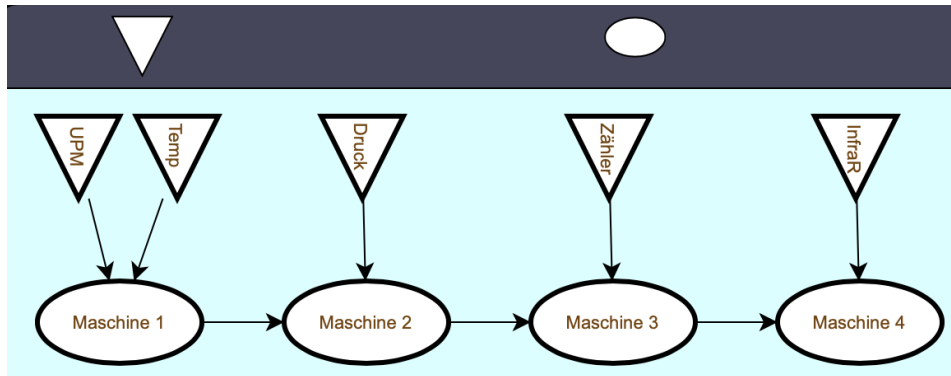


Abb. 3: Oberfläche des IoT-Editors

Für jeden Sensor kann die Art des Sensors festgelegt und eine Wahrscheinlichkeitsverteilung für die auftretenden Werte angegeben werden (Abb. 4).

Name	Timer
Description	Misst die Zeit, die ein Werkstück in dieser Maschine verbleibt. <small>Give your sensor a description.</small>
Your current sensor is the Continuous Sensor	
<input type="radio"/> Unary Sensor <input type="radio"/> Binary Sensor <input type="radio"/> Rotary Sensor <input checked="" type="radio"/> Continuous Sensor <input type="radio"/> Discrete Sensor <input type="radio"/> Timed Sensor <input type="radio"/> Workflow Counter <input type="radio"/> Total Counter <input type="radio"/> Duration Sensor	
Duration Distribution Type	Gaussian distributed duration
Mean Time	120
Standard Deviation	30

Abb. 4: Festlegung der Sensoreigenschaften

Auf Basis der modellierten Geschäftsprozesse und des IoT-Umfelds kann nun eine Zuweisung von Aktivitäten an Orte erfolgen. Hierzu kann der Nutzer aus einer Vielzahl vordefinierter Sensorarten wählen, um beispielsweise eine Smart-Home oder eine Smart-Factory Umgebung digital abzubilden.

Abhängig vom beabsichtigten Simulationszweck kann der Benutzer die Anzahl der Prozessdurchläufe, die benötigten Ausgabefelder, das z.B. Auftreten von Rauschen im Protokoll steuern. Abb. 5 und 6) zeigen die Konfiguration der Simulation und des Rauschens.

5 Beispielhafte Anwendung des IoT-Prozessprotokollgenerators

Im folgenden Abschnitt wird die Anwendbarkeit des Werkzeugs anhand zweier Anwendungsbeispiele demonstriert. Hierzu werden sowohl Krankenhausprozesse als auch Smart-Home

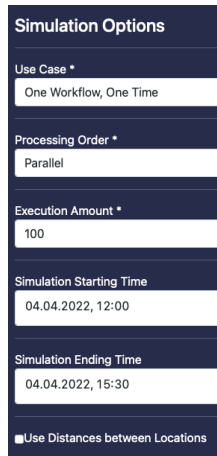


Abb. 5: Konfiguration: Simulation

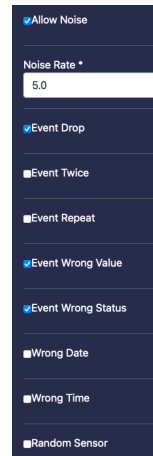


Abb. 6: Konfiguration: Rauschen

Prozesse simuliert. Um ein Ereignisprotokoll für Krankenhausprozesse zu generieren, wurde der Prozess von Elkoumy et al. [El22] angepasst. Das BPMN-Modell wurde in ein Petri-Netz übersetzt, Konfigurationen bezüglich der Dauer von Aktivitäten hinzugefügt und mit Rauschen erweitert. Anschließend wurde der Prozess simuliert. Basierend auf dieser Simulation wurde ein Ereignisprotokoll, siehe Tab. 1, erstellt. Die linke Tabelle zeigt hierbei das tatsächliche Ereignisprotokoll („Ground Truth“), während die rechte Tabelle ein um Rauschen verfälschtes Ereignisprotokoll darstellt.

Log - Clean			Log - Noise			
Case ID	Date Time	Activity	Case ID	Date Time	Activity	Noise Type
846	2022-02-24 08:23	Register	846	2022-02-24 20:23	Register	Wrong Time
846	2022-02-24 09:07	Hospitalize	846	2022-02-24 09:07	Hospitalize	Event Twice
			846	2022-02-24 09:07	Hospitalize	Event Twice
846	2022-02-24 10:46	Blood Test	846	2022-24-02 10:46	Blood Test	Wrong Date
846	2022-02-24 11:18	Blood Test				Event Lost
846	2022-02-24 12:18	Visit	846	2022-02-24 12:18	Visit	Multi Recordings
			846	2022-02-24 12:21	Visit	Multi Recordings
			846	2022-02-24 12:22	Visit	Multi Recordings
846	2022-02-24 13:12	Discharge	846	2022-02-24 13:12	Register	Wrong Event

(a) Ursprüngliches Ereignisprotokoll

(b) Durch Rauschen verfälschtes Ereignisprotokoll

Tab. 1: Synthetisches Ereignisprotokoll zu Krankenhausprozessen

Das zweite Anwendungsbeispiel bezieht sich auf Smart-Home Umgebungen. Zu diesem Zweck wurde auf ein Sensorereignisprotokoll aus [CSE09] zurückgegriffen, welches

Aktivitäten von Bewohnern in einer Smart-Home Umgebung aufgezeichnete. Es wurde eine entsprechende Umgebung modelliert, Sensoren hinzugefügt und alltägliche Aktivitäten im Haushalt wie *kochen*, *aufräumen*, oder *Frühstück zubereiten* simuliert. Die resultierenden Ereignisprotokolle mit und ohne Rauschen sind in Tab. 2a und 2b dargestellt.

Log - Clean				Log - Noise				
Sensor ID	Date	Time	Value	Sensor ID	Date	Time	Value	Noise Type
S1	2022-03-04	08:13	Off	S1	2022-03-04	20:13	Off	Wrong Time
S2	2022-03-04	08:17	On	S2	2022-03-04	08:17	On	Event Twice
S3	2022-03-04	08:25	Off	S2	2022-03-04	08:17	On	Event Twice
S4	2022-03-24	08:36	On					Event Lost
S5	2022-03-04	08:58	Off	S4	2022-03-24	08:36	On	Multi Recordings
F26	2022-03-04	09:33	96.22	S4	2022-03-24	08:36	On	Multi Recordings
S6	2022-03-04	09:42	On	S4	2022-03-24	08:37	On	Multi Recordings
F27	2022-03-04	09:56	0.493	S5	2022-04-03	08:58	Off	Wrong Date
				S2	2022-03-04	09:33	On	Wrong Sensor
				S6	2022-03-04	09:42	Off	Wrong Status
				F27	2022-03-04	09:56	0.557	Wrong Value

(a) Ursprüngliches Sensorereignisprotokoll

(b) Durch Rauschen verfälschtes Sensorereignisprotokoll

Tab. 2: Synthetisches Sensorereignisprotokoll einer Smart Home-Umgebung

6 Ausblick

Dieser Beitrag stellte den IoT-Ereignisprotokollgenerator vor. Das Werkzeug ermöglicht die Erzeugung von synthetischen (Sensor)Ereignisprotokollen für das Process Mining. Die synthetisch erzeugten Ereignisprotokolle können als "Ground-Truth" für z.B. die Kombination von Process Mining und maschinellen Lernverfahren verwendet werden. Nutzer können außerdem festlegen, wie viel Rauschen sie einem Ereignisprotokoll hinzufügen möchten. Auf diese Weise können die synthetisch generierten Ereignisprotokolle dazu verwendet werden Process Discovery Algorithmen zu validieren, die Qualität von entdeckten Prozessmodellen zu erhöhen und neue Anwendungsszenarien für IoT-Sensorereignisdaten zu nutzen.

Bisher ist unser Generator auf die Erzeugung von Ereignisprotokollen auf Ein-Personen-Umgebung beschränkt. Wir planen den IoT-Modellierer dahingehen zu erweitern, dass dieser Mehr-Personen-Umgebungen unterstützt und eine rollenbasierte Aufgabenzuweisung ermöglicht. Außerdem ist es geplant, weitere Prozessmodellierungssprachen und Ausgabeformate zu unterstützen und die Prozessvisualisierung zu verbessern. Die nächste Version des Werkzeugs wird es ermöglichen, Prozesse mit BPMN 2.0 zu modellieren. Neben den Ausgabeformaten .csv und .txt, planen wir die Ereignisprotokolle auch als .xes-Dateien auszugeben. Ebenfalls planen wir die Integration sogenannter Seeds, um eine kontrollierte

Reproduzierbarkeit von Ereignisprotokollen zu ermöglichen. Um die visuelle Darstellung der Prozessmodelle zu verbessern, planen wir einen 3D-Modellierungsumgebung zu integrieren, um synthetische Daten für 3D-Umgebungen zu generieren. Die entsprechende 3D-Umgebung wurde in [WK22] vorgestellt.

Literaturverzeichnis

- [Ah19] Ahmad, Shabir; Malik, Sehrish; Ullah, Israr; Park, Dong-Hwan; Kim, Kwangsoo; Kim, DoHyeun: Towards the Design of a Formal Verification and Evaluation Tool of Real-Time Tasks Scheduling of IoT Applications. *Sustainability*, 11(1):204, Januar 2019.
- [AMK18] Ahmad, Shabir; Malik, Sehrish; Kim, Do-Hyeun: Comparative Analysis of Simulation Tools with Visualization based on Realtime Task Scheduling Algorithms for IoT Embedded Applications. *International Journal of Grid and Distributed Computing*, 11:1–10, Februar 2018.
- [Ch18] Chernyshev, Maxim; Baig, Zubair; Bello, Oladayo; Zeadally, Sherali: Internet of Things (IoT): Research, Simulators, and Testbeds. *IEEE Internet of Things Journal*, 5(3):1637–1647, Juni 2018. Number: 3.
- [Ch19] Chen, Junqiao; Chun, David; Patel, Milesh; Chiang, Epson; James, Jesse: The validity of synthetic clinical data: a validation study of a leading synthetic data generator (Synthea) using clinical quality measures. *BMC Medical Informatics and Decision Making*, 19(1):44, März 2019.
- [Ci15] Ciccio, Claudio Di; Bernardi, Mario Luca; Cimitile, Marta; Maggi, Fabrizio Maria: Generating event logs through the simulation of declare models. In: *Workshop on Enterprise and Organizational Modeling and Simulation*. Springer, S. 20–36, 2015.
- [CSE09] Cook, Diane; Schmitter-Edgecombe, Maureen: Assessing the Quality of Activities in a Smart Environment. *Methods of information in medicine*, 48:480–5, Juni 2009.
- [EF08] Eckleder, A; Freytag, T: Woped a tool for teaching, analyzing and visualizing workflow nets. *Petri Net Newsletter*, 75:3–8, 2008.
- [El22] Elkoumy, Gamal; Fahrenkrog-Petersen, Stephan A.; Sani, Mohammadreza Fani; Koschmider, Agnes; Mannhardt, Felix; von Voigt, Saskia Nuñez; Rafiei, Majid; von Waldthausen, Leopold: Privacy and Confidentiality in Process Mining – Threats and Research Challenges. *ACM*, 13(1):1–17, März 2022. Number: 1 arXiv: 2106.00388.
- [Gi13] Gimenez, Pablo; Molina, Benjamin; Palau, Carlos E.; Esteve, Manuel: SWE Simulation and Testing for the IoT. In: *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, Manchester, S. 356–361, Oktober 2013.
- [Ja20] Janiesch, Christian; Koschmider, Agnes; Mecella, Massimo; Weber, Barbara; Burattin, Andrea; Di Ciccio, Claudio; Fortino, Giancarlo; Gal, Avigdor; Kannengiesser, Udo; Leotta, Francesco; Mannhardt, Felix; Marrella, Andrea; Mendling, Jan; Oberweis, Andreas; Reichert, Manfred; Rinderle-Ma, Stefanie; Serral, Estefanía; Song, Wenzhan; Su, Jianwen; Zhang, Liang: The Internet of Things Meets Business Process Management: A Manifesto. *IEEE Systems, Man, and Cybernetics Magazine*, 6:34–44, 10 2020.

- [JKW07] Jensen, Kurt; Kristensen, Lars Michael; Wells, Lisa: Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9(3):213–254, 2007.
- [Ko21] Koschmider, Agnes; Kaczmarek, Kay; Krause, Mathias; van Zelst, Sebastiaan J.: Demystifying Noise and Outliers in Event Logs: Review and Future Directions. In: *Business Process Management Workshops*. Jgg. 436 in *Lecture Notes in Business Information Processing*. Springer, S. 123–135, 2021.
- [KPG19] Kertesz, A.; Pflanzner, T.; Gyimothy, T.: A Mobile IoT Device Simulator for IoT-Fog-Cloud Systems. *Journal of Grid Computing*, 17:529–551, September 2019. Number: 3.
- [Pa13] Papadopoulos, Georgios Z.; Beaudaux, Julien; Gallais, Antoine; Noël, Thomas; Schreiner, Guillaume: Adding value to WSN simulation using the IoT-LAB experimental platform. In: *2013 IEEE 9th WiMob*. S. 485–490, Oktober 2013. ISSN: 2160-4894.
- [PWV16] Patki, Neha; Wedge, Roy; Veeramachaneni, Kalyan: The Synthetic Data Vault. In: *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. S. 399–410, Oktober 2016.
- [Ra19] Ramprasad, Brian; Fokaefs, Marios; Mukherjee, Joydeep; Litoiu, Marin: EMU-IoT - A Virtual Internet of Things Lab. In: *2019 IEEE International Conference on Autonomic Computing (ICAC)*. S. 73–83, Juni 2019.
- [SSN17] Sharif, Mohammad; Sadeghi-Niaraki, Abolghasem: Ubiquitous sensor network simulation and emulation environments: A survey. *Journal of Network and Computer Applications*, 93:150–181, September 2017.
- [Tr18] Tremblay, Jonathan; Prakash, Aayush; Acuna, David; Brophy, Mark; Jampani, Varun; Anil, Cem; To, Thang; Cameracci, Eric; Boochoon, Shaad; Birchfield, Stan: Training Deep Networks With Synthetic Data: Bridging the Reality Gap by Domain Randomization. In: *2018 IEEE/CVF CVPRW*. S. 969–977, 2018.
- [Va05] Van Dongen, Boudewijn F; de Medeiros, Ana Karla A; Verbeek, HMW; Weijters, AJMM; van Der Aalst, Wil MP: The ProM framework: A new era in process mining tool support. In: *International conference on application and theory of petri nets*. Springer, S. 444–454, 2005.
- [WK22] Wetzel, Manuel; Koschmider, Agnes: Entwicklung einer VR-Umgebung zur Exploration von Process-Mining. *HMD Prax. Wirtsch.*, 59(1):37–53, 2022.