Jochen De Weerdt
Luise Pufahl (Eds.)

LNBIP 492

Business Process Management Workshops

BPM 2023 International Workshops
Utrecht, The Netherlands, September 11–15, 2023
Revised Selected Papers

Springer

MOREMEDIA

Lecture Notes in Business Information Processing 492

Series Editors

Wil van der Aalst, *RWTH Aachen University, Aachen, Germany*
Sudha Ram, *University of Arizona, Tucson, AZ, USA*
Michael Rosemann, *Queensland University of Technology, Brisbane, QLD, Australia*
Clemens Szyperski, *Microsoft Research, Redmond, WA, USA*
Giancarlo Guizzardi, *University of Twente, Enschede, The Netherlands*

LNBIP reports state-of-the-art results in areas related to business information systems and industrial application software development – timely, at a high level, and in both printed and electronic form.

The type of material published includes

- Proceedings (published in time for the respective event)
- Postproceedings (consisting of thoroughly revised and/or extended final papers)
- Other edited monographs (such as, for example, project reports or invited volumes)
- Tutorials (coherently integrated collections of lectures given at advanced courses, seminars, schools, etc.)
- Award-winning or exceptional theses

LNBIP is abstracted/indexed in DBLP, EI and Scopus. LNBIP volumes are also submitted for the inclusion in ISI Proceedings.

Jochen De Weerdt · Luise Pufahl
Editors

Business Process Management Workshops

BPM 2023 International Workshops
Utrecht, The Netherlands, September 11–15, 2023
Revised Selected Papers

Springer

Editors
Jochen De Weerdt
KU Leuven
Leuven, Belgium

Luise Pufahl
TUM School of Computation, Information and Technology
Heilbronn, Germany

ISSN 1865-1348 ISSN 1865-1356 (electronic)
Lecture Notes in Business Information Processing
ISBN 978-3-031-50973-5 ISBN 978-3-031-50974-2 (eBook)
https://doi.org/10.1007/978-3-031-50974-2

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Preface

BPM is an annual international conference that covers all aspects related to Business Process Management. It has become the most prestigious forum for researchers and practitioners in the field. It serves as a melting pot for experts from various disciplines, including Computer Science, Information Systems Engineering, and Management. The BPM conference itself is complemented by a workshop program, where workshops dedicated to more specialized themes, cross-cutting issues, and upcoming trends and paradigms can be easily and conveniently organized with little administrative overhead. This volume collects the proceedings of the BPM 2023 workshops, held on September 11, 2023, in Utrecht, The Netherlands.

BPM 2023 solicited proposals for one-day or half-day workshops to be held before the main conference. In the workshop review and acceptance process, priority was given to proposals that not only addressed an exciting topic but also guaranteed a creative format likely to generate lively interactions and foster new ideas. Examples included panels connecting practitioners and researchers or research-in-progress papers for young researchers. All 11 submitted workshop proposals were selected for co-location with BPM 2023. These workshops include eight established workshops, next to three entirely new workshops on the topics of digital twins for business processes, object-centric business process management, and formal methods in business process management:

- *7th International Workshop on Artificial Intelligence for Business Process Management* (AI4BPM 2023) organized by Chiara Di Francescomarino, Fabrizio Maria Maggi, Andrea Marrella, Arik Senderovich, and Emilio Sulis.

  This workshop aimed to establish a forum for researchers and professionals interested in understanding, envisioning, and discussing the challenges and opportunities of moving from current, largely programmatic approaches for BPM, to emerging forms of AI-enabled BPM.

- *7th International Workshop on BP-Meet-IoT* (BP-Meet-IoT 2023) organized by Francesco Leotta, Massimo Mecella, Estefania Serral, and Victoria Torres.

  BP-Meet-IoT discussed the current state of ongoing research, industry needs, future trends, and practical experiences in the integration between the IoT and BPM fields.

- *19th International Workshop on Business Process Intelligence* (BPI 2022) organized by Johannes De Smedt, Marwan Hassani, and Andrea Burattin.

  This workshop has a long tradition at the BPM conference, and, as usual, it featured presentations of interesting research papers in the BPI domain.

- *16th International Workshop on Social and Human Aspects of Business Process Management* (BPMS2 2023) organized by Rainer Schmidt and Selmin Nurcan.

  The BPMS2 workshop explored how social interactions integrate with BPM and how BPM may profit from this integration. Furthermore, the workshop investigated the human aspects introduced into BPM by involving human users.

- *2nd International Workshop on Data-Driven Business Process Optimization* (BPO 2023) organized by Remco Dijkman, Arik Senderovich, and Willem van Jaarsveld.

  This workshop aimed to bring together researchers from the areas of BPM and Operations Research, as well as other related areas, with the overall goal of developing techniques for optimizing business processes in an organization based on models created from real-world data.

- *11th International Workshop on Declarative, Decision and Hybrid Approaches to Processes* (DEC2H 2023) organized by María Teresa Gómez-López, Claudio Di Ciccio, Tijs Slaats, and Jan Vanthienen.

  DEC2H was interested in the decision- and rule-based modeling and mining of processes, as well as in their hybridization with imperative models in all phases of the BPM lifecycle.

- *1st International Workshop on Digital Twins for Business Processes* (DT4BP 2023) organized by Fabrizio Fornari and Pedro Valderas.

  DT4BP aimed to provide a new forum for researchers and practitioners working at the interplay of business processes and digital twins. Digital process twins can be seen as virtual replicas of the near-time process occurrences in an organization.

- *1st International Workshop on Formal Methods for Business Process Management* (FM-BPM 2023) organized by Claudio Di Ciccio, Alessandro Gianola, and Andrey Rivkin.

  FM-BPM explored techniques, methods, and tools that exploit rigorous mathematical notations for specification, development, and analysis of complex systems.

- *2nd International Workshop on Natural Language Processing for Business Process Management* (NLP4BPM 2023) organized by Han van der Aa, Manuel Resinas, Adela del Río-Ortega, and Henrik Leopold.

  The NLP4BPM workshop aimed to provide a forum for researchers and practitioners to present, discuss, and evaluate how natural language processing (NLP) can be used to establish new or improve existing methods, techniques, tools, and process-aware systems that support the different phases of the BPM lifecycle.

- *1st International Workshop on Object-Centric Processes from A to Z* (OBJECTS 2023) organized by Marco Montali, Andrey Rivkin, and Jan Martijn van der Werf.

  OBJECTS invited contributions presenting novel ideas and advancements of object-centric processes to foster discussions regarding conceptual and technical problems existing in the research domain of object-centric processes.

- *3rd International Workshop on Change, Drift, and Dynamics of Organizational Processes* (ProDy 2023) organized by Bastian Wurm and Jan Mendling.

  ProDy invited conceptual, empirical, and algorithm engineering papers addressing the dynamics of business processes and organizational routines.

All workshops together received a total of 86 submissions. Each workshop had an independent Program Committee, which was in charge of selecting the papers for publication. The workshop papers received at least three single-blind reviews per paper. Of the 86 submissions, 42 papers were selected to be presented at the workshops. Thus, the acceptance rate was 49%.

We thank all workshop proposers and organizers, authors, reviewers, keynote speakers and presenters, and the audience of the BPM 2022 workshops for their contributions to knowledge creation and distribution in the field of Business Process Management. We also thank the organizers and helpers of the BPM 2023 conference, and the University of Utrecht as a great host for a genuinely nice event—lastly, sincere thanks to Springer for their help in publishing the proceedings.

November 2023

Jochen De Weerdt
Luise Pufahl

Contents

7th International Workshop in Artificial Intelligence for Business Process Management (AI4BPM 2023)

A Responsibility Framework for Computing Optimal Process Alignments 5
Matteo Baldoni, Cristina Baroglio, Elisa Marengo, and Roberto Micalizio

A Discussion on Generalization in Next-Activity Prediction 18
Luka Abb, Peter Pfeiffer, Peter Fettke, and Jana-Rebecca Rehse

An Experiment on Transfer Learning for Suffix Prediction on Event Logs 31
Mathieu van Luijken, István Ketykó, and Felix Mannhardt

A Case for Business Process-Specific Foundation Models 44
Yara Rizk, Praveen Venkateswaran, Vatche Isahagian, Austin Narcomey, and Vinod Muthusamy

Using Reinforcement Learning to Optimize Responses in Care Processes: A Case Study on Aggression Incidents 57
Bart J. Verhoef and Xixi Lu

ProtoNER: Few Shot Incremental Learning for Named Entity Recognition Using Prototypical Networks 70
Ritesh Kumar, Saurabh Goyal, Ashish Verma, and Vatche Isahagian

7th International Workshop on Business Processes Meet the Internet-of-Things (BP-Meet-IoT 2023)

Experiences from the Internet-of-Production: Using "Data-Models-in-the-Middle" to Fight Complexity and Facilitate Reuse 87
Wil M. P. van der Aalst

Analyzing Behavior in Cyber-Physical Systems in Connected Vehicles: A Case Study 92
Harry H. Beyel, Omar Makke, Oleg Gusikhin, and Wil M. P. van der Aalst

An Object-Centric Approach to Handling Concurrency in IoT-Aware Processes 105
Florian Gallik, Yusuf Kirikkayis, Ronny Seiger, and Manfred Reichert

Viola: Detecting Violations of Behaviors from Streams of Sensor Data 118
Gemma Di Federico, Giovanni Meroni, and Andrea Burattin

An Event-Centric Metamodel for IoT-Driven Process Monitoring and Conformance Checking 131
Marco Franceschetti, Ronny Seiger, and Barbara Weber

19th International Workshop on Business Process Intelligence (BPI 2022)

Relationships Between Change Patterns in Dynamic Event Attributes 149
Jonas Cremerius, Hendrik Patzlaff, and Mathias Weske

State of the Art: Automatic Generation of Business Process Models 161
Selina Schüler and Sascha Alpers

Navigating Event Abstraction in Process Mining: A Comprehensive Analysis of Sub-problems, Data, and Process Characteristic Considerations 174
Jungeun Lim and Minseok Song

Timed Alignments with Mixed Moves 186
Neha Rino and Thomas Chatain

16th International Workshop on Social and Human Aspects of Business Process Management (BPMS2 2023)

Integrating Social Media and Business Process Management: Exploring the Role of AI Agents and the Benefits for Agility 205
Mehran Majidian Eidgahi, Anne-Marie Barthe-Delanoë, Dominik Bork, Sina Namaki Araghi, Guillaume Macé-Ramète, and Frederick Benaben

Design Principles for Using Business Process Management Systems 217
Sebastian Dunzer, Willi Tang, Nico Höchstädter, Sandra Zilker, and Martin Matzner

Towards a Measurement Instrument for Assessing Capabilities When Innovating Less-Structured Business Processes 229
Joklan Imelda Camelia Goni and Amy Van Looy

2nd International Workshop on Data-Driven Business Process Optimization (BPO 2023)

Timed Process Interventions: Causal Inference vs. Reinforcement Learning 245
Hans Weytjens, Wouter Verbeke, and Jochen De Weerdt

Towards Data-Driven Business Process Redesign Through the Lens of Process Mining Case Studies 259
Zeping Wang, Rehan Syed, and Chun Ouyang

Analyzing the Devil's Quadrangle of Process Instances Through Process Mining 272
Ignacio Velásquez and Marcos Sepúlveda

11th International Workshop on DEClarative, DECision and Hybrid approaches to processes (DEC2H 2023)

Generating Event Logs from Hybrid Process Models 289
Anti Alman, Fabrizio Maria Maggi, Marco Montali, and Andrey Rivkin

Exploring Hybrid Modelling of Industrial Process – Mining Use Case 302
Edyta Brzychczy, Krzysztof Kluza, and Katarzyna Gdowska

Pareto-Optimal Trace Generation from Declarative Process Models 314
Juan F. Diaz, Hugo A. López, Luis Quesada, and Juan C. Rosero

Empirical Evidence of DMN Errors in the Wild - An SAP Signavio Case Study 326
Carl Corea, Timotheus Kampik, and Patrick Delfmann

2nd International Workshop on Digital Twins for Business Processes (DT4BP 2023)

Digital Twin of the Organization for Support of Customer Journeys and Business Processes 341
Wolfgang Groher and Uwe V. Riss

Framing the Digital Business Process Twin: From a Holistic Maturity Model to a Specific and Substantial Use Case in the Automotive Industry 353
Markus Rabe and Emre Kilic

1st International Workshop on Formal Methods for Business Process Management (FM-BPM 2023)

Model-Independent Error Bound Estimation for Conformance Checking Approximation 369
Mohammadreza Fani Sani, Martin Kabierski, Sebastiaan J. van Zelst, and Wil M. P. van der Aalst

Repair of Unsound Data-Aware Process Models 383
Matteo Zavatteri, Davide Bresolin, and Massimiliano de Leoni

Non-Automata Based Conformance Checking of Declarative Process Specifications Based on ASP 396
Isabelle Kuhlmann, Carl Corea, and John Grant

Equivalence of Data Petri Nets with Arithmetic 409
Marco Montali and Sarah Winkler

2nd International Workshop on Natural Language Processing for Business Process Management (NLP4BPM 2023)

Abstractions, Scenarios, and Prompt Definitions for Process Mining with LLMs: A Case Study 427
Alessandro Berti, Daniel Schuster, and Wil M. P. van der Aalst

Text-Aware Predictive Process Monitoring of Knowledge-Intensive Processes: Does Control Flow Matter? 440
Katharina Brennig, Kay Benkert, Bernd Löhr, and Oliver Müller

Large Language Models Can Accomplish Business Process Management Tasks 453
Michael Grohs, Luka Abb, Nourhan Elsayed, and Jana-Rebecca Rehse

Collecting Activities and States in German Business Process Models 466
Ralf Laue, Kristin Kutzner, and Martin Läuter

1st International Workshop of Object-centric Processes from A to Z (OBJECTS 2023)

Typed Petri Nets with Variable Arc Weights 483
Irina A. Lomazova, Alexey A. Mitsyuk, and Andrey Rivkin

Addressing Convergence, Divergence, and Deficiency Issues 496
Jan Niklas Adams and Wil M. P. van der Aalst

A Model-Driven Engineering Perspective for the Object-Centric Event Data (OCED) Metamodel 508
Daniel Calegari and Andrea Delgado

Predictive Analytics for Object-Centric Processes: Do Graph Neural Networks Really Help? 521
Riccardo Galanti and Massimiliano de Leoni

3rd International Workshop on Change, Drift, and Dynamics of Organizational Processes (ProDy 2023)

A Method for Creating Process Architecture: Driving Organizational Change 539
Emerson Lima Aredes, Marina Lourenção, Roger Burlton, Hajo Reijers, and Silvia Inês Dallavale de Pádua

Ambidextrous Business Process Management: Unleashing the Dual Power of Innovation and Efficiency 552
Piotr Sliż

Adding Dynamic Simulation to Business Process Modeling via System Dynamics 565
H. A. Proper, Q. Zhu, J. P. P. Ravesteijn, and W. Gielingh

Author Index 577

7th International Workshop in Artificial Intelligence for Business Process Management (AI4BPM 2023)

Preface of the 7th International Workshop on Artificial Intelligence for Business Process Management

With the growing importance of Artificial Intelligence (AI), numerous novel methodologies and techniques are emerging and being applied in a number of different areas. The application of AI methods to Business Process Management (BPM) is one of these areas which is currently attracting the attention of both industry and academia. The use of AI in BPM has been discussed as one of the emerging technologies that will touch upon almost all business process activities performed by humans. In some cases, AI will dramatically simplify human interaction with processes, while in other cases it will enable full automation of tasks that have traditionally required manual labor. We believe that, in the future, AI may have an important impact in all the phases of the BPM lifecycle: modeling, analysis, automation, implementation, and monitoring. Future AI-augmented BPM technologies will enable constant improvement and adaptation based on continuous experiential learning, thus supporting humans in a number of tasks, such as analysis and decision making.

The goal of this workshop is to establish a forum for researchers and professionals interested in understanding, envisioning, and discussing challenges and opportunities of moving from current, largely programmatic approaches for BPM to emerging forms of AI-driven BPM, hence ‘AI4BPM’.

This year, a keynote speech by Ivan Donadello opened the program of the workshop with an in-depth overview of the application of neuro-symbolic approaches to the BPM field. In addition, the workshop attracted 10 international submissions on different topics including predictive and prescriptive process monitoring, explainability, process discovery, anomaly detection, and conformance checking. All submissions were reviewed by at least 3 program committee members (or their sub-reviewers) and eventually 6 papers were accepted. The accepted papers provided an interesting mix of conceptual and technical contributions from both academia and industry that are of interest for the AI4BPM community.

Baldoni et al. proposed an approach to process alignment leveraging contextual information in the form of responsibilities. Abb et al. presented empirical evidence of how predictive process monitoring approaches for next activity prediction support generalization for new data. Van Luijken et al. proposed an experimental study on the effectiveness of transfer learning for suffix prediction using two sequential deep learning architectures: GPT and LSTM. Rizk et al. introduced a process-specific foundation model tackling issues such as data scarcity, multi-modal representation, and privacy. Verhoef and Lu applied reinforcement learning algorithms to derive optimal policies for staff members when faced with aggressive behavior from clients. Finally, Kumar et al. used few-shot incremental learning for the extraction of information from documents exchanged in business processes to enrich the data used for process analysis.

The workshop ended with a final round table involving the authors and the audience of the workshop, focusing on the importance of fostering the development of techniques for the prediction of next events (not only activities), and in particular the delicate role of next resource predictions. Furthermore, it was discussed to what extent the concept of responsibility can allow specification of more robust approaches to multi-perspective trace alignment in conformance checking.

September 2023

Chiara Di Francescomarino
Fabrizio Maria Maggi
Andrea Marrella
Arik Senderovich
Emilio Sulis

Organization

Program Committee

Han van der Aa	University of Mannheim, Germany
Matteo Baldoni	University of Turin, Italy
Patrizio Bellan	FBK, Italy
Marco Comuzzi	Ulsan Institute, Republic of Korea
Francesco Corcoglioniti	Free University of Bozen-Bolzano, Italy
Massimiliano de Leoni	University of Padova, Italy
Claudio Di Ciccio	Sapienza University of Rome, Italy
Ivan Donadello	Free University of Bozen-Bolzano, Italy
Joerg Evermann	Memorial University of Newfoundland, Canada
Stephan Fahrenkrog-Petersen	Humboldt University of Berlin, Germany
Francesco Folino	CNR, Italy
Fabiana Fournier	IBM Research AI, Israel
Krzysztof Kluza	AGH University of Science and Technology, Poland
Henrik Leopold	Kühne Logistics University, Germany
Francesco Leotta	Sapienza University of Rome, Italy
Lior Limonad	IBM Research AI, Israel
Roberto Micalizio	University of Turin, Italy
Luigi Pontieri	CNR, Italy
Jana-Rebecca Rehse	University of Mannheim, Germany
Massimiliano Ronzani	Fondazione Bruno Kessler, Italy
Tijs Slaats	University of Copenhagen, Denmark
Heiner Stuckenschmidt	University of Mannheim, Germany
Niek Tax	Facebook, UK
Daniele Theseider Dupré	University of Eastern Piedmont, Italy
Matteo Zavatteri	University of Padua, Italy

A Responsibility Framework for Computing Optimal Process Alignments

Matteo Baldoni, Cristina Baroglio, Elisa Marengo(✉),
and Roberto Micalizio

Computer Science Department, University of Turin, Turin, Italy
`{matteo.baldoni,cristina.baroglio,elisa.marengo,`
`roberto.micalizio}@unito.it`

Abstract. In this paper we propose a novel approach to process alignment which leverages on contextual information captured by way of *responsibilities*. On the one hand, responsibilities may justify deviations. In these cases, we consider deviations as correct behaviors rather than errors. On the other hand, responsibilities can either be met or neglected in the execution trace. Thus, we prefer alignments where neglected responsibilities are minimized.

The paper proposes a formal framework for responsibilities in a process model, including the definition of cost functions to determine optimal alignments. It also outlines a branch-and-bound algorithm for their computation.

Keywords: Process Alignment · Responsibilities · Responsibility Alignment

1 Introduction

Trace alignment is a fundamental activity of conformance checking in process mining [11]. It aims at relating an intended behavior, described by way of a process model, and an actual execution trace recorded in an event log. Trace alignment highlights where the actual trace deviates from the process model, and provides insights for further investigations such as performance analysis [1], model repair [11], diagnosis and such like.

Many consolidated approaches (see [11] for an overview) focus on trace alignment from the *control flow perspective*. These approaches try to match a logged trace with a sequence of activities from a process model by scanning both stepwise, looking for mismatches. In the classical approaches, all mismatches have the same weight, and an optimal alignment is one that minimizes them. More recent works consider additional aspects besides the control flow, such as contextual information to weight the impact of a mismatch on the alignment considering when the mismatch occurs [2,9]. Other works show the importance of considering additional perspectives besides the control flow, such as data access [4,15]

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 5–17, 2024.
https://doi.org/10.1007/978-3-031-50974-2_1

and temporal aspects on the occurrence of the events [5,12]. In general, the idea is that when additional information is available, leveraging on it in conformance checking leads to more realistic and informative alignments.

In this paper we focus on *responsibilities*, through which organizations gain the flexibility that a prescriptive representation of a process cannot enjoy. Via responsibilities, in fact, organizations are capable of incorporating (internal or external) regulations, laws, policies, refine objectives and such like.

Responsibilities are captured in different ways within a business organization. RACI matrices, for instance, specify which roles are directly responsible for the completion of what tasks. In the *Business Motivation Model* (BMM) by OMG [16] responsibilities are assigned to *Organizational Unit*. Note that this applies not only to simple tasks, but also every *Business Process* (or complex task) must be under the responsibility of some Organizational Unit.

Responsibilities, thus, are an integral part of business organizations, but so far conformance checking techniques, and trace alignment in particular, have neglected their informative power. Responsibilities, in fact, give us useful insights on when an activity should, or should not, be performed. Consider, for instance, a buying and selling process. An employee is not only responsible for sending a payment confirmation after a payment, but (s)he is also responsible for sending the confirmation only if the payment occurred. Therefore, if in an actual execution both payment and payment confirmation are missing, only the first should be considered as an anomaly. Standard approaches considering the activities mismatches would instead count both as non-compliant behaviors. This is also due to the nature of process models, which is often more prescriptive than strictly required. Complementing such a model with responsibilities would allow us to understand that a mismatch in an alignment is not necessarily a misbehavior in the process execution, but an acceptable alternative.

In this paper we propose: *i)* a formalism for responsibility representation which supports the specification of declarative orderings among activities; *ii)* an alignment strategy that accounts for mismatches with the process model as well as responsibilities that are either satisfied or neglected; *iii)* we outline an algorithm to compute all the optimal alignments.

In Sect. 2 we introduce the responsibility relations and their formalization. Section 3 formalizes the concept of process model extended with the responsibilities. Process and responsibilities alignments are presented in Sect. 4. Related work in Sect. 5 and Conclusions in Sect. 6 end the paper.

2 Responsibilities: Definition and Evaluation in a Trace

The term responsibility is associated with multiple shades of meaning [18]. In this paper, as well as in BMM and other business models, responsibility refers to an actor's duty to perform a task in a given *context*, or role responsibility in the terminology by Vincent [18]. Formally, we represent a responsibility relation as $R(x, \mathsf{u}, \mathsf{v})$ where x is a role, u is a context condition, and v is the duty assigned to x. Intuitively, $R(x, \mathsf{u}, \mathsf{v})$ states that any actor playing role x will be receptive to the need of bringing about v if u holds, and hence it will be answerable about v in that specific context. That is, it would be possible to ask x an account about her involvement in the achievement, or not, of duty v. Condition u and duty v can both be simple activities or temporal patterns on activities executions.

Our proposal is to leverage on responsibilities, and the fact that they can be neglected, as a preference criterion on alignments. In fact, in the real execution of a process, an actor can either meet or neglect her responsibilities. Our goal is to use these events as contextual information for trace alignment, relying on the assumption that role players will act so to be aligned as much as possible with their responsibilities. Let us consider an example inspired by [14].

Example (Alignments and Responsibilities). In a Fine Management Process, a fine is first sent (*Send-Fine*) to the offender, then the offender can either appeal to the judge (*Appeal-Judge*), or pay (*Pay*), in which case a receipt is produced by an employee (*Send-Receipt*). Only two model runs are possible: $\mathit{E1} = \langle \textit{Send-Fine}, \textit{Appeal-Judge} \rangle$ and $\mathit{E2} = \langle \textit{Send-Fine}, \textit{Pay}, \textit{Send-Receipt} \rangle$. Let us consider the observed execution trace: $T = \langle \textit{Send-Fine} \rangle$. The possible alignments with trace T are the following, where $\gg$ represents mismatches (i.e., moves where either the log trace or the model move one step).

$$\mathit{A1} = \begin{array}{c|c} \textit{Send-Fine} & \gg \\ \hline \textit{Send-Fine} & \textit{Appeal-Judge} \end{array} \qquad \mathit{A2} = \begin{array}{c|c|c} \textit{Send-Fine} & \gg & \gg \\ \hline \textit{Send-Fine} & \textit{Pay} & \textit{Send-Receipt} \end{array}$$

Classical approaches would conclude that the model execution $\mathit{E1}$ is closer to the trace because in $\mathit{A1}$ there is one mismatch, while in $\mathit{A2}$ there are two. Let us now assume that the model is complemented with an explicit representation of responsibilities, and that the employee is (always) responsible for producing a receipt *only after* the payment of the fine, and *only in case* it occurs. Assessing the two alignments against such a responsibility allows us to observe that the lack of *Send-Receipt* in $\mathit{A2}$ is actually correct, justified by the fact that the payment did not occur. Therefore, it should not be considered a mismatch and the two alignments can be considered as equivalent in terms of number of mismatches.

Responsibilities provide, in a declarative manner, the expected context of an activity, which is precious for interpreting a logged trace in a way that goes beyond the syntactic distance between strings. Accordingly, we define the cost of an alignment as depending both on the responsibilities that are neglected, and the found mismatches as follows: a neglected responsibility amounts to a cost accumulated by the alignment; a mismatch justified by responsibilities (as for *Send-Receipt* in our example) does not contribute to the cost of an alignment, while other mismatches are considered as misbehavior and contribute to the cost.

To express context conditions and duties in a responsibility relation we rely on precedence logic defined in [17] and summarized in the following.

Precedence Logic. Given a responsibility $R(\mathrm{x}, \mathsf{u}, \mathsf{v})$, we denote the conditions u and v as precedence logic expressions [17], defined over the set of symbols $\Sigma \cup \{0, \top\}$; here, Σ is a set of activity symbols, 0 means false, and $\top$ means true. Precedence logic is an event-based linear temporal logic, obtained from propositional logic augmented with the temporal operator $(\cdot)$ *before*. Such an operator is used to express minimal ordering requirements between events. For instance, $a \cdot b$ expresses the requirement for event a to occur some time before the occurrence of event b (need not be strictly before). Besides the before operator, the logic includes the $\vee$ (choice) and the $\wedge$ (interleaving) operators (capturing that two conditions need to be satisfied but there is no temporal requirements between them). Given a workflow u expressed in precedence logic, the *residual* of u against an event e, denoted as u/e, defines the evolution of u after the occurrence of event e. The residual operator is defined by rules (a – h) below, defined in [7,17]. Here, u is a given workflow, e is an event or $\top$, its complement $\bar{e}$ represents the non-occurrence of e, and Γ_u represents the set of literals in u and their complements (e.g., $\Gamma_{a \cdot b} = \{a, \bar{a}, b, \bar{b}\}$). The residual u/e is defined as:

(a) $0/e \doteq 0$ (b) $\top/e \doteq \top$
(c) $(u_1 \wedge u_2)/e \doteq ((u_1/e) \wedge (u_2/e))$ (d) $(u_1 \vee u_2)/e \doteq ((u_1/e) \vee (u_2/e))$
(e) $(e \cdot u_1)/e \doteq u_1$ if $e \notin \Gamma_{u_1}$ (f) $(u_1/e) \doteq u_1$ if $e \notin \Gamma_{u_1}$
(g) $(e' \cdot u_1)/e \doteq 0$ if $e \in \Gamma_{u_1}$ (h) $(\bar{e} \cdot u_1)/e \doteq 0$

Since 0 amounts to false, and $\top$ to true, the residual operator can be used for assessing whether a workflow expression u is satisfied by a given sequence of events $\sigma = \langle \sigma_1, \ldots, \sigma_m \rangle$ in Σ. Specifically, we denote as u/σ the expression $(((u/\sigma_1)/\sigma_2)\ldots)/\sigma_m$. When u/σ leads to $\top$, σ is a possible execution run of u. When u/σ leads to 0 σ represents a trace not compliant with u. According to [17], it is assumed that *i)* the events in σ are non-repeating (timestamps can be used to differentiate multiple instances of the same event [17]), and *ii)* an event e and its complement $\bar{e}$ are mutually exclusive in every sequence σ.

Evaluate Responsibilities in a Trace. Relying on precedence logic gives us two advantages: 1) *generality*, since we can model both contexts and duties as workflows, and 2) *semantics*, since we can assess the state of a responsibility against a log trace relying on the residual operator. Specifically, we can assess the state of $R(\mathsf{x}, \mathsf{u}, \mathsf{v})$ as either *i)* active, *ii)* discharged *iii)* neglected, or *iv)* satisfied, given an execution trace $\sigma = \langle \sigma_1, \ldots, \sigma_m \rangle$ of events over Σ. Formally, let us denote as $\langle \sigma_1, \ldots, \sigma_i \rangle$ a prefix of σ events with $1 \leq i \leq m$,

- $R(\mathsf{x}, \mathsf{u}, \mathsf{v})$ is *active* at step i (s.t. $i < m$), if $\mathsf{u}/\langle \sigma_1, \ldots, \sigma_i \rangle = \top$ and $\mathsf{v}/\langle \sigma_1, \ldots, \sigma_i \rangle$ is neither $\top$ nor 0;
- $R(\mathsf{x}, \mathsf{u}, \mathsf{v})$ is *discharged* at step i if $\mathsf{u}/\langle \sigma_1, \ldots, \sigma_i \rangle = 0$ (the residual of v is irrelevant);
- $R(\mathsf{x}, \mathsf{u}, \mathsf{v})$ is *satisfied* at step i if $\mathsf{u}/\langle \sigma_1, \ldots, \sigma_i \rangle = \top$ and $\mathsf{v}/\langle \sigma_1, \ldots, \sigma_i \rangle = \top$;
- $R(\mathsf{x}, \mathsf{u}, \mathsf{v})$ is *neglected* at step i if $\mathsf{u}/\langle \sigma_1, \ldots, \sigma_i \rangle = \top$ and $\mathsf{v}/\langle \sigma_1, \ldots, \sigma_i \rangle = 0$, or at step m (the end of the execution) when $\mathsf{u}/\sigma = \top$ and v/σ is not $\top$.

Intuitively, when the responsibility is active there is an expectation on x to bring about v since the context condition u holds. When the responsibility is discharged, instead, the context condition does not hold (and cannot hold along the given σ), and hence no expectation about v can be made. The responsibility is satisfied along σ when both u and v progress to $\top$. Finally, a responsibility is neglected either when, at any execution step, the context condition u holds and the duty v does not, or when, at the end of the trace, u holds and v has not progressed to $\top$, that is, the expectation created with u has not been met.

Example (Responsibilities). Let us consider a set Σ of activity symbols $\{\mathsf{p}, \mathsf{sf}, \mathsf{sr}\}$ standing respectively for *Pay*, *Send-Fine* and *Send-Receipt*. Consider a responsibility relation $R(\mathrm{x},\ \top,\ \ \mathsf{p} \cdot \mathsf{sr})$ expressing that the receipt has to be sent only after the payment. Let us consider the execution $\langle \mathsf{sf}, \mathsf{sr} \rangle$ and apply the residual with respect to it. First, since $\mathsf{sf} \notin \Gamma_{\mathsf{p}\cdot\mathsf{sr}}$, rule (e) applies: $\mathsf{p} \cdot \mathsf{sr}/\mathsf{sf} = \mathsf{p} \cdot \mathsf{sr}$. Then, rule (g) applies to $\mathsf{p} \cdot \mathsf{sr}/\mathsf{sr}$ since $\mathsf{sr} \in \Gamma_{\mathsf{sr}}$, bringing the responsibility to be neglected.

3 Process Model with Responsibilities

In our approach a process model accounts both for the control flow, and for responsibility relations assigned to roles taking part to the process. We distinguish the two parts, defining a *Process Net*, specified as a labeled Petri Net in Definition 1; and complementing it with a set of responsibilities annotating it.

We define a process net as an extension of the process model given in [9] by including a set of roles and assigning them to the activities. A role can be seen as a participant to the process and defined in terms of its function or skills.

Definition 1 (Process Net). *A Process Net is a Labeled Petri Net defined as a tuple $\mathsf{N} = \langle P, T, F, m_0, m_f, \Sigma, \lambda, Z, \zeta \rangle$, where P is the set of places, T is the set of transitions (with $P \cap T = \emptyset$), F is the flow relation $F \subseteq (P \times T) \cup (T \times P)$, m_0 is the initial marking, m_f is the final marking, Σ is the set of activity symbols, $\lambda : T \rightarrow \Sigma \cup \{\tau\}$ labels every transition by an activity or as silent, Z is the set of roles, and $\zeta : \Sigma \rightarrow Z$ assigns a role to every activity in Σ.*

A process net N sets the scope of responsibility relations, since it specifies both the roles Z and the activities Σ over which a responsibility is defined. Responsibilities are defined at design time, and relying on precedence logic allows us to specify that a responsibility be active when a precise execution path occurs. For instance, given a process net N and the activities $\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d} \in \Sigma$, and a role $x \in Z$, to specify that an actor playing role x is responsible for activity d only if activities a, b, and c (in the order but possibly interleaved with other events) occur, one can specify the responsibility relation $R(x,\ \ \mathsf{a} \cdot \mathsf{b} \cdot \mathsf{c},\ \ \mathsf{d})$. Instead, to specify that the responsibility is activated when the three activities occur in any order one can use the relation $R(x,\ \ \mathsf{a} \wedge \mathsf{b} \wedge \mathsf{c},\ \ \mathsf{d})$.

We expect that each responsibility is consistently defined with the process model it refers to. In other terms, both the context and the duty conditions are assumed to be (sub)workflows that can be generated by at least one model run. Therefore, there is always at least a way to satisfy a responsibility.

Activities are part of the context in which responsibilities hold. For instance, by accepting an order, an employee becomes responsible for a number of duties. We "attach" responsibilities to activities, meaning intuitively that a responsibility gets relevant when its corresponding activity is performed. For instance,

$R(x,\ \mathsf{a}\cdot\mathsf{b}\cdot\mathsf{c},\ \ \mathsf{d})$ can be attached to e, to express that it gets relevant when e is executed. The context condition captures that, if e can be reached from more than one path, the responsibility is activated only by the path where $\mathsf{a}\cdot\mathsf{b}\cdot\mathsf{c}$ holds. Definition 2 formally define the responsibility labelling of a process net.

Definition 2 (Responsibility Labelling). *Let* N *be a process net, and let* Z *and* Σ *be, respectively, the set of roles and activity symbols in* N*. A responsibility labeling over* N *is a function* $R:\Sigma\rightarrow\{R_1,\ldots,R_n\}$ *where each* R_i *is a responsibility relation* $R(x_i,\mathsf{u}_i,\mathsf{v}_i)$*, such that:* $x_i\in Z$ *and* u_i *and* v_i *are precedence logic expressions over* $\Sigma\cup\{0,\top\}$*.*

A *process model* is then defined as a pair $\mathsf{M}=\langle\mathsf{N},\mathsf{R}\rangle$ where N is a process net as in Definition 1, and R is a responsibility labelling as in Definition 2.

4 Flow and Responsibility Alignments

An alignment compares a process execution against an execution trace (i.e., a log trace). Generally, the objective is to find, among the possible ones, an alignment which is optimal w.r.t. a criterion of preference. Intuitively, an alignment proceeds step-by-step on the model and on the log: at each step, if the activity in the model and the one in the log match each other, a *synchronous move* is made, and both model and log advance one step. Otherwise, either the *model moves* and the log does not, or the other way around, the *log moves* and the model does not. Usually, to find an optimal matching, a cost function associated with mismatches (i.e., asynchronous moves) is defined. So, an optimal alignment is the one minimizing the cumulative cost of the mismatches. The classical approach is to minimize the number of asynchronous moves [11].

In our approach, an optimal alignment is determined taking into account both the alignment between a log trace and a model run, and the involved responsibilities. We refer to the former as *flow alignment* and to the latter as *responsibility alignment*. Definition 3, adapted from [9], formally introduces the notion of *flow alignment*, capturing the alignment between a process net and an execution trace. The symbol $\gg$ represents a *no-move*, and is used for marking asynchronous moves. More in general, given a process model $\mathsf{M}=\langle\mathsf{N},\mathsf{R}\rangle$, we use the term *model run* for the sequence of activity symbols in Σ produced by a full run of the process net N, where a Petri Net full run is a sequence of firings from the initial marking to the final one [9]. We also assume the process net N to be *easy sound* [11], that is, there exists at least one full run. The term *log trace*, instead, refers to an actual execution of a process instance of M, it is therefore a finite sequence of activity symbols $\sigma\in\Sigma^*$ (i.e., the space of sequences defined over symbols in Σ).

Definition 3 (Flow Alignment). *Let* $\sigma=\langle\sigma_1,\ldots,\sigma_m\rangle$ *be a log trace in* Σ^**, and* $\mathsf{N}=\langle P,T,F,m_0,m_f,\Sigma,\lambda,Z,\zeta\rangle$ *a process net, an alignment of* σ *with the process net* N *is a finite sequence* $\varphi=\langle(\sigma'_1,u'_1),\ldots,(\sigma'_p,u'_p)\rangle$ *of moves such that:*

- *each move is either: a synchronous move (a,t)* $\in \Sigma \times T$ *with* $a = \lambda(t)$*, a log move* $(a, \gg)$*, or a model move* $(\gg, t)$*,*
- *dropping the* $\gg$ *symbols from the left projection* $(\sigma'_1, \ldots, \sigma'_p)$ *of* φ*, yields* σ*,*
- *dropping the* $\gg$ *symbols from the right projection* $(u'_1, \ldots, u'_p)$ *of* φ*, yields a full run* u *of* N*.*

To consider the responsibilities in evaluating the optimal alignment, our approach is to collect all the responsibilities attached to the activities of a model run (i.e., the responsibilities that should be satisfied along a possible, expected execution), and assess them against a log trace (to check if indeed they are satisfied). The cost of an alignment, thus, takes also into account the cost of neglected responsibilities. Moreover, the responsibilities collected along a model run give us a context for assessing whether a model move (i.e., a "skip" on the log side) actually represents an execution error, or a proper behavior.

Given a flow alignment φ, its *responsibility set* is the set of responsibilities attached to the activities in the model run given by the right projection of φ (i.e., the model-side projection). In general, a responsibility set can be computed for any non-empty prefix of φ by considering the alignment up to a given step j.

Definition 4 (Responsibility Set). *Let* $\varphi = \langle(\sigma'_1, u'_1), \ldots, (\sigma'_p, u'_p)\rangle$ *be a flow alignment between a process model* $\mathsf{M} = \langle \mathsf{N}, \mathsf{R}\rangle$ *and a log trace* $\sigma \in \Sigma^*$*, the responsibility set* $\mathscr{R}^{\varphi,j}$ *for the alignment* φ *at step* j $(1 \leq j \leq p)$ *is defined as* $\mathscr{R}^{\varphi,j} = \cup_{i=1}^{j} \mathsf{R}(\lambda(u'_i))$*.*

It holds $\mathsf{R}(\gg) = \emptyset$. As a shortcut, we denote as $\mathscr{R}^{\varphi}$ the set $\mathscr{R}^{\varphi,p}$, that is the set of responsibilities computed considering all the steps in the alignment φ. The responsibilities in $\mathscr{R}^{\varphi}$ are actually satisfied or neglected depending on the activities that are included in the log trace (i.e., log-side projection of φ). Thus, we first extend the notion of residuation of the precedence logic to responsibility relations, and then to a responsibility set.

Given a responsibility set $\mathscr{R}^{\varphi} = \{\mathsf{R}_1, \ldots, \mathsf{R}_k\}$ with $\mathsf{R}_i = R(\mathrm{x}_i, \mathsf{u}_i, \mathsf{v}_i)$, let $\sigma' = \langle\sigma'_1, \ldots, \sigma'_p\rangle$ be the log-side projection of φ. Then, the notation R_i/σ' is a shorthand for $R(\mathrm{x}_i, \mathsf{u}_i/\sigma', \mathsf{v}_i/\sigma')$ and $\mathscr{R}^{\varphi}/\sigma'$ is a shorthand for $\{\mathsf{R}_1/\sigma', \ldots, \mathsf{R}_k/\sigma'\}$. Additionally, the residuation of any expression u with $\gg$ has no effect on the expression, namely $\mathsf{u}/\gg = \mathsf{u}$.

Proposition 1. *[Consistency] Let* $\mathscr{R}^{\varphi,j}$ *be the responsibility set computed at step* j $(1 \leq j \leq p)$ *of an alignment* $\varphi = \langle(\sigma'_1, u'_1), \ldots, (\sigma'_p, u'_p)\rangle$*, let* σ' *be the projection log-side of* φ*, then for each* $\mathsf{R}_i \in \mathscr{R}^{\varphi,j}$ *the following conditions hold:*

1. *if* $\mathsf{R}_i/\langle\sigma'_1, \ldots, \sigma'_j\rangle$ *is satisfied, then also* R_i/σ' *is satisfied;*
2. *if* $\mathsf{R}_i/\langle\sigma'_1, \ldots, \sigma'_j\rangle$ *is neglected, then also* R_i/σ' *is neglected;*
3. *if* $\mathsf{R}_i/\langle\sigma'_1, \ldots, \sigma'_j\rangle$ *is discharged, then also* R_i/σ' *is discharged;*
4. *if* $\mathsf{R}_i/\langle\sigma'_1, \ldots, \sigma'_j\rangle$ *is active, then* R_i/σ' *is either satisfied or neglected.*

This proposition, that follows directly from the rewriting rules of the precedence logic, guarantees a consistent evaluation of the responsibilities against a log trace. In fact, whenever a responsibility progresses from the active state to either satisfied, neglected, or discharged, such a second state is final: the state of the responsibility can no longer evolve along the same trace. That is, further events along the trace cannot satisfy a neglected responsibility nor vice versa. At the same time, a responsibility activated along a log trace must necessarily evolve to either satisfied or neglected by the end of the same trace.

Example (Responsibility Set). Consider the example in Sect. 2, the responsibility set $\mathscr{R}^{A1}$ of alignment A1 will be the union of the responsibilities associated with the activities *Send-Fine* and *Appeal-Judge*, while the set $\mathscr{R}^{A2}$ for A2 is given by the responsibilities associated to *Send-Fine*, *Pay* and *Send-Receipt*. The resulting sets will be residuated with respect to the log trace (i.e., activity *Send-Fine*) and the cost of the alignment computed as describe in the following.

4.1 Cost Functions for Optimal Alignments

In general, several alignments of a log trace w.r.t. a model exist. To compare them and determine the optimal one we define a cost function that considers both the cost of the mismatches between the model run and the log trace, which we call *Flow Alignment Cost* $\mathscr{C}_{\mathsf{N},\varphi}$, and the cost for the neglected responsibilities, which we call *Responsibility Alignment Cost* $\mathscr{C}_{R,\varphi}$.

Let us start from the latter one. The responsibility cost $\mathscr{C}_{R,\varphi}$ is computed for a flow alignment $\varphi = \langle(\sigma'_1, u'_1), \ldots, (\sigma'_p, u'_p)\rangle$ between a process model $\mathsf{M} = \langle \mathsf{N}, \mathsf{R}\rangle$ and a log trace $\sigma \in \Sigma^*$. The cost $\mathscr{C}_{R,\varphi}$ corresponds to the number of responsibilities that are neglected in φ. To compute them, first the responsibility set $\mathscr{R}^{\varphi}$ for φ is determined (Definition 4). Then, $\mathscr{R}^{\varphi}$ is residuated with respect to the projection log-side of φ. Neglected responsibilities are then those that are active, but not satisfied at the end of the trace. Note that the approach can be easily extended to consider different responsibility costs (for instance to capture that some responsibilities are more important to be satisfied than others).

Example (Responsibility Alignment Cost). Let us consider the fine process scenario, and assume that the employee is responsible for archiving (ar) any sent fine (sf) after 60 days, independently of whether the offender has appealed to the judge or paid for the fine; that is, responsibility $R(\mathrm{x}, \mathsf{sf}, \mathsf{ar})$ is attached to sf. Now, in both alignments A1 and A2 (see Sect. 2), the responsibility is activated but, since it is not satisfied by the end of both alignments, it is marked as neglected in both cases, thus bringing a cost in both alignments.

The second component of our cost function, the flow cost $\mathscr{C}_{\mathsf{N},\varphi}$, calculates the cost of every mismatch (i.e., either model or log moves) included within a given alignment. Notably, this calculation takes into account responsibilities as a sort of context. By using them, in fact, we are able to identify some model moves as correct, and not as mismatches. To this end, $\mathscr{C}_{\mathsf{N},\varphi}$ is computed with respect to a flow alignment $\varphi = \langle(\sigma'_1, u'_1), \ldots, (\sigma'_p, u'_p)\rangle$ between a process model $\mathsf{M} = \langle \mathsf{N}, \mathsf{R}\rangle$ and a log trace $\sigma \in \Sigma^*$. To compute $\mathscr{C}_{\mathsf{N},\varphi}$, each alignment step (σ'_j, u'_j) is considered. A step (σ'_j, u'_j) is a mismatch, and hence to be counted as a cost, when it is either a log move (i.e., the label $\lambda(u'_j)$ assigned to transition

u'_j is $\gg$), or it is a model move (i.e., σ'_j equals $\gg$) which is not justified by any responsibility. A model move is justified by a responsibility if there is at least one $R(x, \mathsf{u}, \mathsf{v}) \in \mathscr{R}^{\varphi,j}$ which is not neglected at step j, but it would be if the log event σ'_j were substituted by the corresponding model activity $\lambda(u'_j)$. In other words, executing the activity corresponding to a synchronous move would have lead an active responsibility $R(x, \top, \mathsf{v})$ to progress to neglected i.e., $R(x, \top, 0)$. This means that skipping activity $\lambda(u'_j)$ is consistent with at least one responsibility, and hence $\gg$ does not represent a misbehavior.

Note that this approach is compatible with other approaches in the literature where the cost of a mismatch is not necessarily constant nor one (e.g., [2,9]).

Example (Flow Alignment Cost). Let us consider again alignment A2 in Sect. 2, and let us assume that $R(\mathrm{x}, \top, \ \mathsf{p} \cdot \mathsf{sr})$ is associated with activity *Pay* to indicate that, if the expected execution is the one that goes through *Pay* rather than *Appeal-Judge*, then the receipt has to be sent only after and only in case of a payment. Concerning the Flow Alignment Cost, albeit A2 has two asynchronous model moves, only the first one actually contributes to the flow cost. The log skip on *Send-Receipt* (sr), instead, is justified since the occurrence of sr in the log would lead to the violation of $R(\mathrm{x}, \top, \ \mathsf{p} \cdot \mathsf{sr})$.

The total cost of an alignment is computed as the weighted sum of the flow and the responsibility costs.

Definition 5 (Alignment Cost, Optimal Alignment). *Let* $\varphi = \langle(\sigma'_1, u'_1), \ldots, (\sigma'_p, u'_p)\rangle$ *be the flow alignment between a process model* $\mathsf{M} = \langle \mathsf{N}, \mathsf{R}\rangle$ *and a log trace* $\sigma \in \Sigma^*$*, the cost function of the alignment* φ *is*

$$\mathscr{C}_\varphi = \gamma \cdot \mathscr{C}_{\mathsf{N},\varphi} + \delta \cdot \mathscr{C}_{R,\varphi}$$

An alignment between a model M *and a log trace* σ *is optimal if* $\mathscr{C}_\varphi$ *is minimal.*

Coefficients γ and δ are domain-dependent weights that can be tuned for penalizing more either neglected responsibilities or asynchronous moves. A greater weight for δ (responsibility coefficient) prefers asynchronous moves to neglected responsibilities. On the other hand, a greater weight for γ (flow coefficient), prefers neglected responsibilities to mismatches, thus the found alignments would be characterized by as many synchronous moves as possible even though this could lead to violate responsibilities.

4.2 Computation of the Optimal Alignment

To compute the optimal alignment, many approaches in the literature adopt A* or branch-and-bound algorithms. The approach we outline in this section exploits a branch-and-bound strategy, and relies on the Synchronous Product Net (SN) between the process net and a sequential Petri Net representing the log trace. Intuitively, the SN combines the two nets representing the synchronous and asynchronous moves. The SN is built as follows: every model (resp. log) transition is augmented with the $\gg$ symbol to represent asynchronous moves. Synchronous moves are represented with additional transitions, each labeled with the activity synchronously performed. We rely on the formal definition of SN as in [11]. In this setting, an alignment corresponds to a *full run* in the SN, that is a firing sequence bringing from the initial marking to a final marking, where the final markings are those where both the process net and the log trace reach one of their final markings.

Given in input the SN, it is possible to implement a *branch and bound* search strategy which finds the optimal full runs, i.e., those runs having minimal cost computed according to Definition 5. The idea is as follows. The search space is given by the set of possible markings of the SN; for each marking, a suitable data structure keeps *i)* the marking, *ii)* the sequence of transitions leading to it (each transition is an alignment step), *iii)* the cost of the alignment up to that point, *iv)* the set of responsibilities collected along this (partial) alignment. The algorithm keeps in a queue the frontier of the search, i.e., the markings still to be considered. The frontier is ordered in a non-decreasing way with respect to the marking cost, and it is initialized with the initial marking.

At each step of the search, the algorithm removes the top marking from the frontier (i.e., the one with the lowest cost). If the top marking is a final marking, then the cost of the responsibilities that are neglected because not satisfied at the end of the alignment (and not counted in the previous steps), need to be added to the alignment cost. Then the cost of the found solution is compared with the best cost found up to that point and, if it is equally good is added to the set of optimal solutions while if it is better it replaces the current best. Otherwise, it is discarded.

On the other hand, if the top marking is not a final marking, the algorithm computes, for each enabled transition, the reachable markings from this top marking (the top marking is now visited and will not be visited again in the future). For each reachable marking, if not yet visited, the algorithm computes the responsibility set and the cost up to this point, and then the marking is stored in the frontier in the proper order. The responsibility set is computed incrementally, by adding to the set of responsibilities collected up to that point, the responsibilities associated to the last transition. The cost associated with the marking is computed by combining flow and responsibilities costs as in Definition 5. To compute the flow cost, we count the asynchronous moves along the current alignment; in case of a model move, a cost is added only if the skip in the log is not justified by at least one responsibility associated with this marking (as explained in Sect. 4.1). On the other side, the responsibility cost counts all the responsibilities that, associated with this marking, are neglected by a synchronous or model move. It is worth noting that, this cost function is monotone, in fact, by Proposition 1, there is no case when the cost for a neglected responsibility needs to be reverted since the very same responsibility gets satisfied.

5 Related Work

Several proposals in the literature focus on extending process alignment considering other perspectives besides the control flow. Among these, several consider data aspects [4,13,15], time [12], costs and such like. Interestingly, in [14] the authors propose a cost function which, similarly to our proposal, combines the costs from different perspectives.

Approaches focusing on the control flow differentiate the cost of a misalignment based on when it occurs [2,8,9]. In [9] mismatches occurring at the early stages of an execution have more impact than those occurring at later stages. Authors in [2] also consider which activities occurred before and after that mismatch. The proposal in [8], instead, is to favor alignments where the number of synchronous moves is maximized. This is achieved by defining a cost function which penalizes log moves only.

Considering the technical aspects, instead, many proposals rely on Petri Nets and specifically on the Synchronous Product Net. In general, the cheapest path search techniques rely on Dijkstra- or A^*-based algorithms [3,9]; possibly, pruning techniques are applied to speed up the overall search [9,11]. Approaches based on planning [2] and SAT algorithms [10] have also been proposed.

To the best of our knowledge, no existing approaches consider the perspective of responsibilities. In our proposal, we consider it in two ways: as an additional perspective compared to the control flow and to evaluate model moves. One similarity with [8], is that we propose a strategy for not counting the cost of model moves. However, while in [8] model moves are ignored systematically, in our approach we do not consider a model move as a cost only when performing the corresponding synchronous move would lead to neglect some responsibilities.

6 Conclusions and Future Work

We presented a novel methodology of process alignment which takes into account responsibilities (see Sect. 2) during the search for optimal alignments. The paper contributes with a formal framework for representing responsibilities and using them for complementing a process model. A branch and bound algorithm is also discussed whose implementation is still in progress.

An explicit representation of responsibilities as part of a process model opens several future directions. First, each alignment found by our algorithm is associated with a set of met and unmet responsibilities. In first lieu, these two sets provide a sort of justification why a specific alignment has been selected as optimal. More generally, however, by considering the set of unmet responsibilities for a number of log traces, one could reason about possible inefficiencies and flaws in the process model, enabling a responsibility-driven procedure for re-engineering a process, where the responsibilities themselves could be redefined. In addition, role responsibilities designate the actors playing a specific role as "in charge" of some job, and hence capable of providing accounts about the accomplishment, or failure, of that job. This permits the creation of accountability relationships between the actor who is responsible for a job, and another actor who has some interest in that job (e.g., for her decision process) [6]. By means of accountability, problems can be detected and examined with the objective to understand why the problem has occurred. An interesting future direction, thus, is to complement our responsibility framework with accountability relationships as a way for improving both the computation of alignments and their understanding in the context of a business organization.

Acknowledgments. This work is part of the projects NODES, which has received funding from the MUR-M4C2 1.5 of PNRR with grant agreement no. ECS00000036, and "APE: Accountability and Processes from Events", financed by MUR in the context of the "Grant for Internationalization" call.

References

1. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. WIREs Data Mining Knowl. Discov. **2**(2), 182–192 (2012)
2. Acitelli, G., Angelini, M., Bonomi, S., Maggi, F.M., Marrella, A., Palma, A.: Context-aware trace alignment with automated planning. In: ICPM 2022 (2022)
3. Adriansyah, A.: Aligning observed and modeled behavior. Ph.D. thesis, Technische Universiteit Eindhoven (2014)
4. Alizadeh, M., Lu, X., Fahland, D., Zannone, N., van der Aalst, W.M.P.: Linking data and process perspectives for conformance analysis. Comput. Secur. **73**, 172–193 (2018)
5. Baldoni, M., Baroglio, C., Capuzzimati, F., Marengo, E., Patti, V.: A generalized commitment machine for `2CL` protocols and its implementation. In: Baldoni, M., Dennis, L., Mascardi, V., Vasconcelos, W. (eds.) DALT 2012. LNCS (LNAI), vol. 7784, pp. 96–115. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37890-4_6
6. Baldoni, M., Baroglio, C., Micalizio, R., Tedeschi, S.: Robustness based on accountability in multiagent organizations. In: AAMAS. IFAAMAS (2021)
7. Baldoni, M., Baroglio, C., Micalizio, R., Tedeschi, S.: Accountability in multi-agent organizations: from conceptual design to agent programming. Auton. Agents Multi Agent Syst. **37**(1), 7 (2023)
8. Bloemen, V., van Zelst, S.J., van der Aalst, W.M.P., van Dongen, B.F., van de Pol, J.: Aligning observed and modelled behaviour by maximizing synchronous moves and using milestones. Inf. Syst. **103**, 101456 (2022)
9. Boltenhagen, M., Chatain, T., Carmona, J.: A discounted cost function for fast alignments of business processes. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 252–269. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_17
10. Boltenhagen, M., Chatain, T., Carmona, J.: Model-based trace variant analysis of event logs. Inf. Syst. **102**, 101675 (2021)
11. Carmona, J., van Dongen, B.F., Solti, A., Weidlich, M.: Conformance Checking - Relating Processes and Models. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-99414-7
12. Di Federico, G., Burattin, A.: Do you behave always the same? - a process mining approach. In: Montali, M., Senderovich, A., Weidlich, M. (eds.) ICPM 2022. LNBIP, vol. 468, pp. 5–17. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-27815-0_1
13. Felli, P., Gianola, A., Montali, M., Rivkin, A., Winkler, S.: CoCoMoT: conformance checking of multi-perspective processes via SMT. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 217–234. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_15

14. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Balanced multi-perspective checking of process conformance. Computing **98**, 407–437 (2016)
15. Mozafari Mehr, A.S., de Carvalho, R.M., van Dongen, B.F.: Detecting complex anomalous behaviors in business processes: a multi-perspective conformance checking approach. In: Montali, M., Senderovich, A., Weidlich, M. (eds.) ICPM 2022. LNBIP, vol. 468, pp. 44–56. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-27815-0_4
16. OMG: BMM Model (2015). https://www.omg.org/spec/BMM/1.3/PDF
17. Singh, M.P., Huhns, M.N.: Service-Oriented Computing - Semantics, Processes, Agents. Wiley, Hoboken (2005)
18. Vincent, N.: A structured taxonomy of responsibility concepts. In: Vincent, N., van de Poel, I., van den Hoven, J. (eds.) Moral Responsibility: Beyond Free Will and Determinism. Library of Ethics and Applied Philosophy, vol. 27, pp. 15–35. Springer, Dordrecht (2010). https://doi.org/10.1007/978-94-007-1878-4_2

A Discussion on Generalization in Next-Activity Prediction

Luka Abb[1(✉)], Peter Pfeiffer[2], Peter Fettke[2], and Jana-Rebecca Rehse[1]

[1] University of Mannheim, Mannheim, Germany
{luka.abb,rehse}@uni-mannheim.de
[2] German Research Center for Artificial Intelligence (DFKI) and Saarland University, Saarbrücken, Germany
{peter.pfeiffer,peter.fettke}@dfki.de

Abstract. Next activity prediction aims to forecast the future behavior of running process instances. Recent publications in this field predominantly employ deep learning techniques and evaluate their prediction performance using publicly available event logs. This paper presents empirical evidence that calls into question the effectiveness of these current evaluation approaches. We show that there is an enormous amount of example leakage in all of the commonly used event logs, so that rather trivial prediction approaches perform almost as well as ones that leverage deep learning. We further argue that designing robust evaluations requires a more profound conceptual engagement with the topic of next-activity prediction, and specifically with the notion of generalization to new data. To this end, we present various prediction scenarios that necessitate different types of generalization to guide future research.

Keywords: Predictive Process Monitoring · Process Prediction · Generalization · Leakage

1 Introduction

Predictive process monitoring (PPM), or process prediction, is a branch of process mining that is concerned with the forecasting of how a running process instance will unfold in the future [3]. For example, PPM approaches may predict what the outcome of the process instance will be, how long it will take to complete, or which activities will be executed next. In contrast to techniques like process discovery or conformance checking, process prediction is forward-facing, and aims to identify process execution problems like delays or compliance violations *before* they occur, thus enabling an organization to preemptively take preventive counteractions [3].

Whereas older approaches to process prediction relied on explicit models of process behavior, such as transition systems or probabilistic automata [1], recent

---

L. Abb and P. Pfeiffer—Equal contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 18–30, 2024.
https://doi.org/10.1007/978-3-031-50974-2_2

research has almost exclusively tackled the problem with neural networks [4]. The majority of this research has also focused on control-flow predictions, specifically the prediction of the *next activity* in a trace [6]. At a high level, all existing contributions approach next-activity prediction as a self-supervised machine learning problem [7,9,11]: An existing event log is randomly split into a training and a test set. A machine learning model, typically a deep neural network, is shown incomplete traces from the training set, such that it learns to predict the next activity in that trace. The performance of the trained model is then evaluated by predicting the next activity for incomplete traces of the unseen test set and computing performance measures. Almost all existing publications train and evaluate their models on a relatively small collection of event logs which include the Helpdesk event log [14] and the logs from the Business Process Intelligence Challenges (BPIC) 2012, 2013, and 2017. In the pursuit of always higher accuracy values, most papers have introduced new architectures while the suitability of the data and evaluation procedures for this task has been discussed little.

In this paper, we argue that this current way of training and evaluating next-activity prediction models is biased in the sense that it does not evaluate how well these models would generalize to unseen data. We argue that, in order to design reliable evaluation procedures, it is necessary to first engage with the topic of next-activity prediction on a more conceptual level. Our line of argument is based on several observations about the aforementioned event logs: First, in most cases, the next-activity label can already be correctly predicted when only the control-flow of the prefix is known. Second, when only considering the control-flow perspective, there is an enormous amount of example leakage in all logs, so that most predictions are made on prefixes that were already seen during training. Third, as other research has already shown [10], incomplete traces can often continue in different ways, so that the maximal achievable accuracy in this evaluation setting is unknown but, by design, lower than 100%.

After introducing basic concepts in Sect. 2, we provide empirical evidence for each of these observations and demonstrate that the next-activity prediction task in these event logs is a rather trivial one that can be solved by a naive baseline (Sect. 3). Section 4 presents various scenarios for generalization in process prediction which are grouped into three types of generalization. Finally, we discuss related work in Sect. 5 and conclude the paper in Sect. 6.

2 Background

Event Log Data. PPM works on *event log data*, gathered from the execution of business processes in information systems. An event log is a collection of cases. A case is represented by a *trace* t, i.e., a sequence of events $\langle e_1, \ldots, e_n \rangle$ of length n. Each event e has two mandatory attributes: the *activity* and the *case ID*. In addition, events can have additional attributes, such as a timestamp or an executing resource, which describe the context in which the event has occurred. Similar to events, traces can also have additional attributes, such as an allocated project. A case represents a completed process execution. For PPM, we are interested in predicting the future behavior of running cases, which are represented by trace prefixes. A *trace prefix* of a trace t of length p is defined as a subsequence $\langle e_1, \ldots, e_p \rangle$, with $1 \leq p < n$.

Next-Activity Prediction. The goal of next-activity prediction is to predict which activity is performed next in a running case. Formally, this problem is framed as multi-class classification, where each class represents one activity. For each trace t in a given event log, pairs (x, y) of features x and labels y are created. x is a prefix of t with length p, which represents the running case. y, which is often called the label of x, represents the activity at position $p+1$ of t, i.e., the next activity, which should be predicted. These pairs (x, y) are provided to a machine learning model, typically a deep neural network, such that it learns a function f that predicts for each prefix the correct next activity, i.e., the class to which the prefix belongs. To learn and evaluate f, the event log is split into two parts, the training set and the test set. The model is trained on the prefix-label pairs from the training set and evaluated on those from the test set. Therefore, for each prefix x, its prediction $\hat{y} := f(x)$ is compared with the ground truth label y and performance measures like accuracy and F1 score can be computed.

3 Validity Issues in Existing Research

In this section, we examine various phenomena that pose threats to the validity of next-activity prediction research. To substantiate our discussion, we present empirical evidence that was generated in a setting that is representative of the typical evaluation setup used in the field. We employ five commonly used event logs (Helpdesk, BPIC12, BPIC13 Incidents, BPIC17 Offer, and MobIS [12]) and generate six splits for each log: five in which we randomly allocate traces so that 80% of them are part of the training set and 20% are part of the test set, and one in which the split is time-based so that the 20% of traces with the most recent start timestamps end up in the test set. We then generate $n-1$ prefix-label pairs (x, y) from each trace with lengths $p \in [1, n-1]$ and calculate prediction accuracy as the percentage of prefixes in the test set for which the correct next-activity label was predicted, i.e., $\hat{y} = y$. We do not apply log preprocessing or make any other changes to the data. The code and data needed to reproduce our results are available online[1].

3.1 Example Leakage

Leakage in machine learning refers to information being made available to a model during training that it would not have access to when classifying unseen data [5]. This can lead to an unrealistic assessment of the model's performance with respect to the classification task at hand. One particular type of leakage is *example leakage*, which occurs when the same example (more specifically, the same feature vector) is present in both the training and the test set. In this

---

[1] https://gitlab.uni-mannheim.de/jpmac/ppm-generalization.

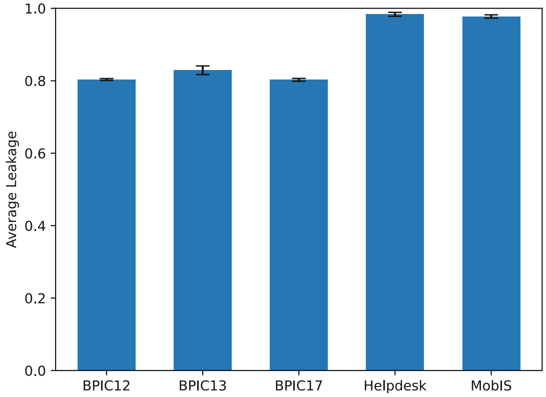


Fig. 1. Example leakage percentage for each event log, averaged over the splits.

case, the classification is a trivial one, as the model is not required to learn generally-valid relationships between features and labels. Example leakage can be a considerable problem when doing predictions on event logs, due to the repetitive nature of the process executions recorded in them [15].

In order to quantify example leakage in next-activity prediction, we first need to establish when two prefixes can be considered identical. We can limit the set of features that need to be considered when establishing equality to those that are actually relevant for predicting the next activity. Previous research has already examined the extent to which context attributes, such as resource or time, enhance prediction performance compared to solely considering the previous control-flow recorded in a prefix [2]. They have found that, in most cases, including context attributes does improve predictions compared to only considering control-flow features, but that these improvements are rather insignificant (low single-digit percentage increases in accuracy). Based on these findings, we can conclude that, in most cases, the next-activity label can be correctly predicted when only the control-flow of the prefix is known. In the following, we therefore consider two prefixes to be identical if they exhibit the same control-flow, i.e., if they have the same activities in the exact same order.

With this equality criterion, we can now quantify example leakage by calculating the percentage of prefixes in the test set that is also included in the training set. The amount of example leakage in the event logs commonly used for the evaluation of next-activity prediction techniques is shown in (Fig. 1). We observe that, across all datasets and splits, example leakage is above 80%, and even close to 100% in the Helpdesk and MobIS event logs. This means that most of the predictions made on the test set are trivial ones, and consequently, that one cannot draw valid conclusions about how well a prediction model would perform on unseen data from this evaluation setting.

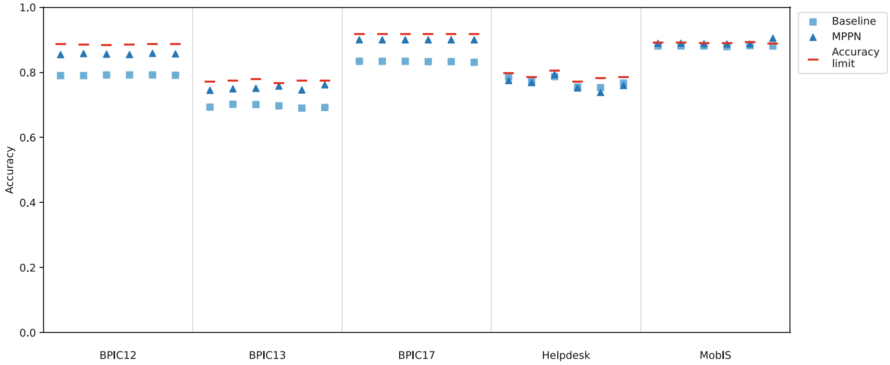


Fig. 2. Prediction accuracy of the naive baseline and the MPPN neural network, along with the accuracy limit in the test set. Each split plotted separately.

3.2 Baseline and Accuracy Limit

We can further illustrate this issue by demonstrating that the prediction accuracy of state-of-the-art models lies in a relatively narrow corridor that is bounded by a naive, frequency-based baseline on the lower end, and by the maximal accuracy that can be achieved with only control-flow features on the upper end. We construct the baseline as follows: for each unique prefix in the training set $x := \langle e_1, \ldots, e_p \rangle$, where e represents the activity only, it simply predicts the most common next activity. If an unknown prefix is encountered (i.e., an example that has not leaked), it instead predicts the most common next activity associated with only the last activity e_p in the prefix, similar to a bigram model, i.e., $x := e_p$.

The upper bound is based on the observation that a common implicit assumption in supervised learning, that each unique combination of feature values maps to exactly one label, does not hold in the process mining domain. Event logs nearly always contain traces that have identical control-flow up to a point but diverge afterwards, for example due to exclusive continuation paths or concurrent activity execution. In the context of next-activity prediction, this means that a prefix exhibits *label ambiguity* [10]. If a prediction model that predicts a single next-activity label is tasked with classifying a label-ambiguous prefix, the best prediction in terms of the resulting overall accuracy it can make is the activity that is most frequently associated with that prefix. All other activities will never be predicted.

From this, we can derive that there is an *accuracy limit* that a prediction model can achieve on a given (test) dataset when it only makes predictions based on the control-flow of the prefix. This accuracy limit is simply calculated as the percentage of examples in the test set in which the label is the most common label for the corresponding prefix. Figure 2 shows the prediction accuracy achieved by the baseline prediction model described above and the MPPN [9], a state-of-the-art neural network predictor that includes contextual attributes for its prediction. The accuracy limit for each test split is also included. Of course, this comparison is limited since it only includes a single state-of-the-art model. However, given that benchmark experiments in previous research have consistently shown that many next-activity prediction models achieve almost the same accuracy when evaluated on the same data (e.g., [7,9,11]), our observations are likely to apply to other models as well.

In the Helpdesk and MobIS logs, the training and test set almost completely overlap. Reaching high accuracies on the test set is therefore trivial for these logs, and consequently, both models achieve the same prediction accuracy. In fact, the only reason that they do not reach 100% accuracy might be label ambiguity, which is why the observed accuracy for these models is almost identical to the accuracy limit. In other event logs, which exhibit slightly less example leakage, the accuracy of the naive baseline is still very close to the one of the state-of-the-art model, although there is a notable gap of a few percentage points. However, it is unclear where this gap comes from. One explanation would be that MPPN's ability to take context information into account help to make more accurate predictions whereas the baseline only considers control-flow features; this would be consistent with the findings of [2], i.e., that incorporating context slightly improves prediction accuracy.

Given that this evaluation setting has been widely employed in existing publications on next-activity prediction, our findings suggest that some of the perceived advancements in the field may be – in a sense – illusory. As a research community, we now have a large number of proposed next-activity prediction techniques that employ several different neural network architectures, inductive biases, and strategies to incorporate different types of features. However, these techniques have only ever been evaluated in a setting that suffers from high example leakage and label ambiguity. Therefore, we cannot say with certainty to what extent these approaches would be able to generalize well enough to make good predictions on unseen data – and consequently, if they would be able to provide value in a real-world application.

Although it would be possible to address the issues that we have pointed out in this section on a technical level, we argue that they are symptomatic of a broader problem in process prediction research, namely that the topic is not well defined and understood on a conceptual level. In particular, we believe that our findings call for a discussion of what *generalization* means in a process prediction context.

4 Generalization in Process Prediction

In machine learning, generalization refers to the ability of a trained model to make correct predictions on samples that it has not seen during training, i.e., to reach a low error rate on unseen samples. If the model is able to make correct predictions on such samples, it is expected to be robust enough to work well in the respective application context. Furthermore, it ensures that the model has learned generally valid features for solving the task instead of memorizing the samples.

In order to test generalization capabilities, one must ensure that the samples which the trained model is evaluated on are different to the samples is has been trained on. Although generalization is a characteristic of interest for machine learning in general and process prediction in particular, the generalization capabilities of PPM algorithms have so far not been explicitly evaluated, in the sense of applying an algorithm on a test log that has little to no overlap with the training data[2]. As pointed out in the previous section, splitting an event log into training and test sets randomly or on a temporal basis does not ensure this.

In this section, we presents several exemplary prediction scenarios, classified into different generalization types, and discusses which predictions a PPM algorithm should reasonable make in each. These scenarios are inspired by real life event logs and tailored for the use case of generalizing on event log data, i.e., account for the characteristics of it. However, they are synthetic and not meant to be complete. Rather, they are intended to serve as a starting point for evaluating generalization in process prediction and on creating test splits, which allow to assess generalization. Each scenario presents a situation in which we expect a PPM algorithms to make reasonable predictions on unseen samples to explicitly evaluate generalization capabilities, according to the understanding of generalization in machine learning.

4.1 Prediction Scenarios

In the following scenarios, we suppose to train a prediction model on the mentioned log L, i.e., we create all prefixes for all traces t in the log L and train the model on the resulting samples (x, y). For each scenarios we show new prefixes x that are not seen so far, i.e., $x \notin L$. Given the unseen prefix x as input to the model, we explain which predictions are plausible to be made. If the model is able to make this prediction on the unseen prefix, we say that it can generalize in this scenario.

Note that we focus on the problem of predicting the next activity only. Predicting attributes like resource, time or properties like the process outcome are related problems, but the correct predictions differ, so they require a separate discussion. Furthermore, we assume that we do not have access to additional information like a process model; only the observations in the event log are given.

Unseen Control-Flow. Log $L1$ in Table 1 shows the scenarios where activities $C1$, $C2$ and $C3$ can occur in any order. $L2$ in Table 2 shows a similar, yet more complex scenario with C, D, E, F, G, H in any order. This can be caused, e.g., by concurrent activities and is a common phenomenon in real-world event logs. Another common scenario is the appearance of activities that can be executed multiple times after another as shown in $L3$. For event logs with such patterns, four interesting scenarios can occur:

---

[2] A notable exception to this is [8], which focuses on process model structures.

Table 1. Concurrency

Event Log $L1$
⟨A, B, C1, C2, C3, D, E⟩
⟨A, B, C2, C1, C3, D, E⟩
⟨A, B, C2, C3, C1, D, E⟩
⟨A, B, C3, C1, C2, D, E⟩
⟨A, B, C3, C2, C1, D, E⟩

Table 2. Concurrency with label ambiguity

Event Log $L2$
⟨A, B, C, D, E, F,G, H⟩
⟨A, B, C, F, D,G, E, H⟩
⟨A, B, C, D, F, E,G, H⟩
⟨A, B, F, C, D,G, H, E⟩

Table 3. Loops

Event Log $L3$
⟨A, B, C, D⟩
⟨A, B, B, C, D⟩

1. $L1$ and prefix ⟨A, B, C1, C3, C2, D⟩. Expected prediction: E. Although the model has not seen this prefix due to a new order of $C1$, $C2$ and $C3$, it should have learned that the case always continues with E after D, regardless of the order of the previous activities.
2. $L1$ and prefix: ⟨A, B, C1, C3, C2⟩. Expected prediction: D. Again, the prediction model should have learned that regardless of the order of $C1$, $C2$ and $C3$, D always follows.
3. $L2$ and prefix: ⟨A, B, C, D, F, G⟩. As seen in $L2$, both E and H have happened after G. However, in each trace, either E or D directly follows G. This is the situation of label ambiguity described in [10]. Both options, E and D are valid continuations and thus valid predictions.
4. $L3$ and prefix: ⟨A, B, B, B, C⟩. Expected prediction: D. The model should have learned that the case always continues with D after C, no matter how often B has happened.

Table 4. Example Log with different resources R performing B

Event Log $L4$
⟨(A, R1), (B, R100), (C, R2)⟩
⟨(A, R1), (B, R101), (C, R2)⟩
⟨(A, R1), (B, R101), (C, R2)⟩

Table 5. Example Log with decision depending on cost after B.

Event Log $L5$
⟨(A, 2€), (B, 2€), (C, 2€)⟩
⟨(A, 499€), (B, 499€), (C, 499€)⟩
⟨(A, 501€), (B, 501€), (D, 501€)⟩

Table 6. Example Log with concept drift in 2023.

Event Log $L6$
⟨(A, May 2022), (B, June 2022), (C, June 2022)⟩
⟨(A, July 2022), (B, July 2022), (C, July 2022)⟩
⟨(A, April 2023), (B, May 2023), (D, May 2023)⟩

Unseen Attribute Value Combinations. In certain scenarios, the context attributes like involved resources, timestamp or cost carry important information to determine the continuation of the process instance [2,12]. Considering the contextual information is an important capability when dealing with event logs which distinguishes next step prediction from other sequential prediction tasks. As an example, we show three scenarios where we expect the prediction model to generalize in presence of context attributes. Note that in these scenarios, the models have seen the context attribute values before, i.e., they are not completely new. Just the combination of activity and context has not been seen so far. The first example, $L4$ in Table 4, shows a situation in which different resources are involved in the activities. Log $L5$ in Table 5 gives an example where the next activity to execute depends on the amount of Euro [12]. Lastly, log $L6$ in Table 6 shows an example where timestamps are involved.

1. $L4$ and prefix ⟨(**A**, R1), (**B**, R1)⟩. Expected prediction: C. In $L4$, different resources are involved in activity B. However, C follows B every time. Thus, the prediction model should know that regardless of the resource R in activity B, C always follows.
2. $L5$ and prefix ⟨(**A**, 2€), (**B**, 499€)⟩. Expected prediction: C. The value of Euro has changed to 499€. However, the model should have learned that with 499€ C still follows.
3. $L6$ and prefix: ⟨(**A**, July 2022), (**B**, May 2023)⟩. Expected prediction: D. In 2023, a drift happened causing activity D to follow B instead of C, which the prediction model should be able to express.

Unseen Attribute Values. Sometimes, the training log might not be complete with respect to the activities or other attributes contained. For instance, a new activity (e.g. due to new requirements in the process) or a new resource (e.g. a new person joining the process/company) might occur. To demonstrate these scenarios, we use the logs $L4$, $L5$ and $L6$ from the previous section but discuss other prefixes.

1. $L4$ and prefix ⟨(A, R1), (F, R100)⟩. As F is an activity the prediction model has never seen before, there is no evidence from the event log how to continue. One option is to indicate that the model does not know. For instance by making a special prediction, e.g., *UNKNOWN*. Another option would be to predict any label from the event log that could follow potentially, e.g., C as this has happened in the third position in all traces in the log.
2. $L4$ and prefix ⟨(A, R1), (B, R37)⟩. This scenario is similar to the previous one but with resource $R37$ never seen before. Again, the model could indicate that it does not know or predict any label on positional basis, e.g., C.
3. $L5$ and prefix ⟨(**A**, 200€), (**B**, 200€)⟩. 200€ is between the seen values 2€ and 499€. Thus, we argue that the prediction model should predict C.
4. $L6$ and prefix ⟨(**A**, June 2024), (**B**, June 2024)⟩. The model should know that the process has changed in 2023. If tasked with 2024, the most probable next activity is D.

4.2 Implications

Generalization over unseen control-flow constructs involves dealing with unseen control-flow variants in the prefix as shown in the scenarios in event logs $L1$, $L2$

and $L3$ in Table 1, Table 2 and Table 3. We assume that all activities in prefix and label are known but the specific prefix has not been seen so far. The event log $L2$ in Table 2 is a special scenario as it is linked to label ambiguity [10]. In such a situation, both options, E and H are valid prediction. However, a deterministic model will always make the same prediction when tasked with the same prefix. As H has the higher frequency, the prediction model will most likely always predict H although it should have – and probably has – learned that E can also follow. When evaluating process prediction methods with point-measures like top-1 accuracy, which consider only the single most probable prediction, one cannot assess generalization properly as it does not take into account whether the model has learned that more than one option can follow. To do so, probabilistic measures can be used that assess how much probability is given for each option.

For generalization over unseen context combinations, the prediction model must be able to interpret the context attributes and to distinguish between those scenarios where the context attributes have influence on the next activity to be predicted and those scenarios where they do not. This involves scenario as shown in logs $L4$, $L5$ and $L6$ in Table 4, Table 5 and Table 6. There can be much more complex scenarios with other context attributes where the next activity depends not only on one but the combination of multiple attribute values.

Generalization over new and unknown attribute values are scenarios where a new attribute value like a completely unknown activity or resource occur. In such scenarios, defining plausible predictions is often not trivial and might depend one the use-case. For numerical and temporal attributes, unseen attribute values are more diverse. For instance, the number of unique values for cost in Table 5 can be very large and the chance that all values have been seen is rather low. Similarly, as dates are continuous the prediction model might in practice be tasked with prefixes with year 2024 or 2025.

Especially in this, but also in other situations, the most reasonable approach is to make a decently confident prediction for the most likely next activity and to indicate whether the model knows the correct answer or whether it does not know. For instance, the model might predict a certain activity which usually occurred in this position in the trace but at the same time indicate that it did predict this activity only on positional basis as it has never seen this attribute value in the trace. In practice, these scenarios might not occur in isolation. For instance, an unseen sequence of activities in the prefix can also come with unseen combination of context attributes or new attribute values which makes generalization in process prediction a challenging task.

5 Related Work

So far, the conceptual flaws of process prediction and realistic evaluation procedures beside label ambiguity [10] have been discussed little. Weytjens et al. [15] introduce a pre-processing algorithms to prevent leakage in process prediction focusing on the remaining time prediction problem. Their approach splits the traces on a temporal basis such that there is no temporal overlap between the prefixes used for training and test. However, this does not prevent example leakage on prefix-level.

In [13], the authors compare discovery-based algorithms to sequence-learning algorithms in terms of their accurateness and generalization capabilities. The event logs are split into training and test sets. However, as the paper does not mention any technique to prevent example leakage, it is very likely that the splits used in the experiments face a similar high portion of leaked prefixes which limits the validity of generalization capabilities measured.

Peeperkorn et al. [8] propose an evaluation strategy to leave certain variants out of the training set and only have them in the test set. They used this splitting strategy to evaluate whether prediction models can learn process model structure of the unknown system behind the log, focusing mainly on concurrent activities in process models. Thus, they did not systematically cover all generalization scenarios introduced in this paper. They found that the generalization capabilities of LSTM prediction models are inversely correlated with the number of variants left out. However, as they measured with accuracy, it is unclear how label ambiguity affected the experiments. In comparison to their work, we propose several generalization scenarios.

6 Conclusion

In this paper, we have critically analyzed the current procedure of evaluating PPM algorithms in research and found that little to no generalization capabilities can be tested that way due to a high percentage of label leakage. This phenomena in combination with label ambiguity lead to the situation that the accuracy values of naive, frequency-based next step prediction approaches and more complex neural-network-based approaches lie in a relatively narrow corridor which is bounded by label ambiguity on the upper end. In conclusion, this raises the question what the accuracy values are worth and how well existing approaches would actually perform in the wild.

As a initial step towards a more reliable evaluation of next step prediction approaches, we presented several scenarios that can be used to discuss, measure and test generalization in process prediction. For instance, they can be used to measure how much difference between train and test set there is and which generalization capabilities are required for which log, i.e., which scenarios are present and which not. Furthermore, synthetic event logs containing these pattern can be simulated and existing one split accordingly to test for generalization. Guided by the plausible predictions, new prediction algorithms can be developed that specifically account for these. In addition, probability-based interpretations of process prediction algorithms could help to overcome some limitations of existing approaches and evaluation procedures.

While the generalization scenarios are inspired by real-world situations, real event logs are required for setting the ground truth label of unseen prefixes. In the scenarios presented, we assumed a ground truth label and argued whether such a prediction will show generalization. In some scenarios, the expected label is more clear than in other scenarios. However, these are only plausible predictions. Real generalization can only be tested if the ground truth label is not assumed but determined by the data. Nevertheless, one should always have label ambiguity in mind when evaluating with event log data.

Although we have focused on next-activity prediction and other prediction situations were out of scope for this work, there might be more scenarios in next-activity prediction that are not yet covered. Furthermore, the high percentage of example leakage between train and test set raises the question whether generalization capabilities are actually required if the behaviour in both sets is that similar when considering the control-flow only. Following that, prediction models that take context information into account might actually be able to generalize with respect to the scenarios of unseen attribute value combination, as they reach comparable or higher accuracy as control-flow only models. Nevertheless, this has yet not been shown explicitly.

References

1. Breuker, D., Matzner, M., Delfmann, P., Becker, J.: Comprehensible predictive models for business processes. MIS Q. **40**(4), 1009–1034 (2016)
2. Brunk, J., Stottmeister, J., Weinzierl, S., Matzner, M., Becker, J.: Exploring the effect of context information on deep learning business process predictions. J. Decis. Syst. **29**(sup1), 328–343 (2020)
3. Di Francescomarino, C., Ghidini, C.: Predictive process monitoring. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. Lecture Notes in Business Information Processing, vol. 448, pp. 320–346. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_10
4. Evermann, J., Rehse, J.R., Fettke, P.: Predicting process behaviour using deep learning. Decis. Support Syst. **100**, 129–140 (2017)
5. Kaufman, S., Rosset, S., Perlich, C.: Leakage in data mining: formulation, detection, and avoidance. In: KDD Conference, vol. 6, pp. 556–563. ACM, New YOrk (2011)
6. Neu, D., Lahann, J., Fettke, P.: A systematic literature review on state-of-the-art deep learning methods for process prediction. Art. Int. Rev. **55**, 1–27 (2022)
7. Pasquadibisceglie, V., Appice, A., Castellano, G., Malerba, D.: A multi-view deep learning approach for predictive business process monitoring. IEEE Trans. Serv. Comp. **15**(04), 2382–2395 (2022)
8. Peeperkorn, J., Broucke, S.V., De Weerdt, J.: Can recurrent neural networks learn process model structure? J. Intell. Inf. Syst. **61**, 1–25 (2022)
9. Pfeiffer, P., Lahann, J., Fettke, P.: Multivariate business process representation learning utilizing Gramian angular fields and convolutional neural networks. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 327–344. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_21
10. Pfeiffer, P., Lahann, J., Fettke, P.: The label ambiguity problem in process prediction. In: Cabanillas, C., Garmann-Johnsen, N.F., Koschmider, A. (eds.) BPM 2022. LNBIP, vol. 460, pp. 37–44. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-25383-6_4

11. Rama-Maneiro, E., Vidal, J., Lama, M.: Deep learning for predictive business process monitoring: review and benchmark. IEEE Trans. Serv. Comp. **16**(1) (2021)
12. Scheid, M., Rehse, J.R., Houy, C., Fettke, P.: Data set for MobIS challenge 2019 (2018). https://doi.org/10.13140/RG.2.2.11870.28487
13. Tax, N., Teinemaa, I., van Zelst, S.J.: An interdisciplinary comparison of sequence modeling methods for next-element prediction. Softw. Syst. Model. **19**(6), 1345–1365 (2020)
14. Verenich, I.: Helpdesk event log. https://doi.org/10.17632/39bp3vv62t.1
15. Weytjens, H., De Weerdt, J.: Creating unbiased public benchmark datasets with data leakage prevention for predictive process monitoring. In: Marrella, A., Weber, B. (eds.) BPM 2021. LNBIP, vol. 436, pp. 18–29. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94343-1_2

An Experiment on Transfer Learning for Suffix Prediction on Event Logs

Mathieu van Luijken, István Ketykó, and Felix Mannhardt(✉)

Eindhoven University of Technology, Eindhoven, The Netherlands
`m.f.a.c.v.luijken@student.tue.nl`, `{i.ketyko,f.mannhardt}@tue.nl`

Abstract. Predicting future activity occurrences for a process instance is a key challenge in predictive process monitoring. Sequential deep learning models have been improving the prediction accuracy for this suffix prediction task. Training such models with many parameters on large event logs requires expensive hardware and is often time consuming. Transfer learning addresses this issue by starting from a pre-trained model to be used as starting point for the training on other data sets thereby reducing training time or improving accuracy in a given time budget. Transfer learning has shown to be very effective for natural language processing and image classification. However, research on transfer learning for predictive process monitoring is scarce and missing for suffix prediction. This paper contributes an experimental study on the effectiveness of transfer learning for suffix prediction using two sequential deep learning architectures (GPT and LSTM). Base models are trained on two public event logs and used as starting point for transfer learning on eight event logs from different domains. The experiments show that even with half of the available training budget and without using very large event logs for the base model, the results obtained in the transfer learning setting are often better and in some cases competitive to when trained using random initialization. A notable exception is an event log with a very large vocabulary of activity labels. This seems to indicate dependence of transfer learning on specific data properties such as vocabulary size and warranting further research.

Keywords: Transfer learning · Suffix prediction · Deep learning

1 Introduction

Predictive process monitoring promises to predict future states of business processes to improve the handling of process instances [1]. Being able to accurately predict how process instances (cases) will unfold has many applications and can help organizations to act on possible problems with their processes before they cause damage. Several predictive tasks have been defined: outcome prediction

---

This work is part of the *Smart Journey Mining* project, funded by the Research Council of Norway (grant no. 312198).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 31–43, 2024.
https://doi.org/10.1007/978-3-031-50974-2_3

focusing on the final state after process execution [2], next event prediction looking at what activity occurs next [3], and predicting the entire continuation of a process instance [4]. The latter task is denoted as suffix prediction and takes as input a sequence of events already occurred and predicts the remaining events for the running process instance.

Deep Learning (DL) has shown to be competitive for this suffix prediction task [5,6] and the performance of various DL architectures was explored [4–6]. Training DL-based models is computationally expensive. Successfully training large models on large data is feasible only due to expensive hardware that allows the DL training to be performed in parallel on large batches of cases. Often the training needs to be repeated several times to find a good configuration of hyperparameters. This problem is not specific to predictive process monitoring but also occurs in other domains such as computer vision and natural language processing. For example, the training of state-of-the-art language models with billions of parameters is impossible without access to large clusters of purpose-made hardware. Similarly, training a large DL model on event logs from real-life and high-volume processes can be challenging.

A solution to avoid expensive training that was successful in other domains is *transfer learning*. Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a related second task [7,8]. Reasoning for performing transfer learning are two-fold: Data pooling and lowering computing costs. Data pooling is when data from two different sources are used to train a single model. The technique is used when there is not enough specific data to solve one problem or if the addition of data from a different domain is likely to increase model performance. Transfer learning can lower computing costs when the weights from a pre-trained model are closer to the optimum required for a new dataset rather than random initialization. In this case the number of training epochs necessary to reach a good performance can decrease rather drastically. Often a pre-trained model is used as starting point and further trained (fine tuned) to specialise on a new dataset. Transfer learning has been shown to perform well on different data type and in many domains, e.g., in image classification [9,10], text sentiment classification [11], and multi-language text classification [12]. Research on the effectiveness of transfer learning in the domain of process mining is scarce with only three papers [13–15] mentioning transfer learning. None of the existing work investigates transfer learning across different processes none considers suffix prediction as investigated in our work. Instead, all of the studies focus on training a base model on a subset of the original event log, e.g., only short traces, and then leverage this base model as starting point for further training on longer traces. Thus, a transfer only happens within the same process.

This paper contributes a systematic study of transfer learning for suffix prediction for transferring between different processes. The experiment performed uses two sequential DL models that are tested on eight real-life event logs. The basis is an existing framework for comparing sequential DL suffix prediction approaches [4] from which we select Long Short Term Memory (LSTM) and

Generative Pre-trained Transformer (GPT) as commonly used DL approaches. The models and the framework are further described in Sect. 2. In our transfer learning procedure, described in Sect. 3, we train base models on two event logs and investigate the effectiveness of *transfer learning* by comparing the prediction performance reached in a transfer learning setting with half of the training budget to that for a baseline of training with random initialization. We explore several scenarios by varying which parts of the model to fine tune. In our evaluation (Sect. 4) we compare in total 224 training runs that result from taking two base event logs and for each of them comparing eight target event logs over two architectures (LSTM, GPT) with seven different fine-tuning scenarios.

Our results show that for the majority of the event logs, i.e., five out of eight logs, transfer learning increases the accuracy of activity prediction and time-until-next-event prediction when restricted to half of the training budget for the random initialization baseline. For two logs, the activity prediction accuracy is similar to the baseline and for one log the result on activity prediction is worse. So, in many cases and with proper fine tuning configuration transfer learning would be a viable solution to lower computational cost. However, more research is required to understand the impact of process characteristics on transfer learning, e.g., a much larger alphabet of activities may have been contributed to the one negative result. An extension of our experiment towards investigating whether DL models can truly learn certain control-flow constructs [16] and investigating the impact of process and event sequence complexity [17] could shed more light on the utility of transfer learning for different settings.

2 Preliminaries

We define the suffix prediction task on event logs and briefly review the used DL framework as well as the LSTM and GPT architectures. We keep this part brief since we leverage the existing framework of Keyko et al. [4].

2.1 Suffix Prediction on Event Logs

An *event log* contains a number of traces and is denoted as $D = \{\sigma^{(1)}, \dots, \sigma^{(d)}\}$ where d is the number of traces. These traces are sequences consisting of n different events where each individual *trace* is denoted as $\sigma^{(i)} = \langle e_1, \dots, e_n \rangle$. In our setting, we consider events $e_j = (a_j, t_j)$ to only contain two different attributes: an activity a_j and an execution time t_j, which we pre-process to store the relative time or duration since the previous event in a trace. Any given trace $\sigma^{(i)} = \langle (a_1, t_1), \dots, (a_k, t_k) \rangle$ can be split up into a prefix $\sigma^{(i)}_{\leq k} = \langle (a_1, t_1), \dots, (a_k, t_k) \rangle$ and its corresponding suffix $\sigma^{(i)}_{>k} = \langle (a_{k+1}, t_{k+1}), \dots, (a_n, t_n) \rangle$. The problem of *suffix prediction* is to take a prefix $\sigma^{(i)}_{\leq k}$ and use the information in this prefix to predict its corresponding suffix $\sigma^{(i)}_{>k}$ consisting out of predicted activities and corresponding duration's.

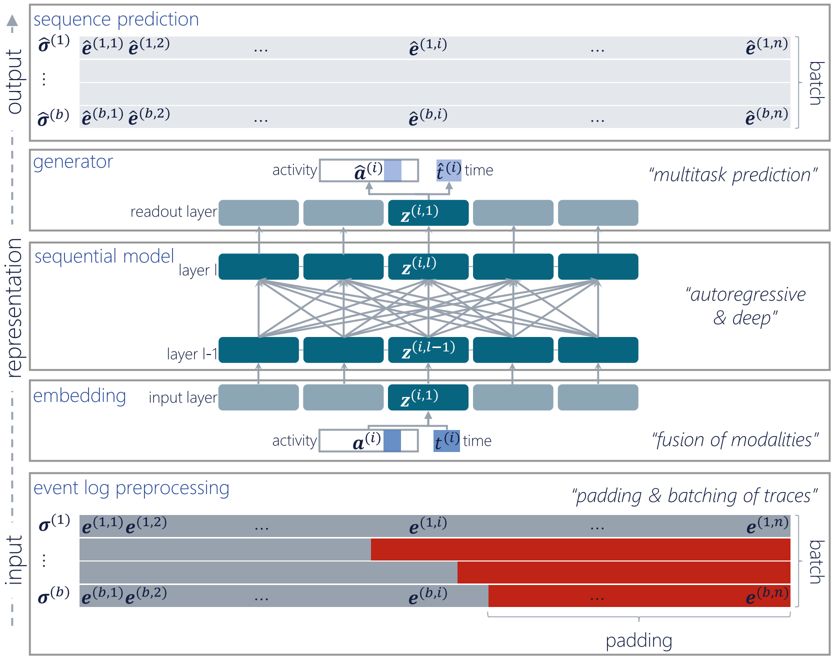

output
representation
input
sequence prediction
$\hat{\sigma}^{(1)}$ $\hat{e}^{(1,1)}$ $\hat{e}^{(1,2)}$ … $\hat{e}^{(1,i)}$ … $\hat{e}^{(1,n)}$
$\hat{\sigma}^{(b)}$ $\hat{e}^{(b,1)}$ $\hat{e}^{(b,2)}$ … $\hat{e}^{(b,i)}$ … $\hat{e}^{(b,n)}$
batch
generator
activity $\hat{a}^{(i)}$
$\hat{t}^{(i)}$ time
"multitask prediction"
readout layer
$z^{(i,1)}$
sequential model
layer l
$z^{(i,l)}$
"autoregressive & deep"
layer l-1
$z^{(i,l-1)}$
embedding
input layer
$z^{(i,1)}$
activity $a^{(i)}$
$t^{(i)}$ time
"fusion of modalities"
event log preprocessing
"padding & batching of traces"
$\sigma^{(1)}$ $e^{(1,1)}$ $e^{(1,2)}$ … $e^{(1,i)}$ … $e^{(1,n)}$
$\sigma^{(b)}$ $e^{(b,1)}$ $e^{(b,2)}$ … $e^{(b,i)}$ … $e^{(b,n)}$
batch
padding

Fig. 1. The DL framework used for suffix prediction proposed by [4].

2.2 DL Framework

To evaluate the effect of transfer learning, we made use of the framework proposed in [4]. The framework in [4] has the advantage that it is unified over many different DL architectures without any specific feature engineering or event log pre-processing. None of the traces of the event logs are trimmed or excluded providing a fair basis for comparison. We make the common assumption in DL that traces are independent from each other and, therefore, as defined in Sect. 2.1, only use the activity and timestamp attributes available in every event log to avoid possibly correlated features.

Figure 1 shows that it consists of three different parts: the embedding part, a sequential model and lastly the generator part. We briefly revisit their main function and refer to [4] for the in-depth description.

Embedding. The input to the framework is an event log that is divided into batches of size b. Padding is used to obtain sequences of equal size n and special tokens are added. The special tokens, which do not occur in the original event log, are: $[SOS]$ which is added to the start of each sequence, $[EOS]$ which is added to the end of each sequence, and $[PAD]$ which are used for padding. In the embedding layer an event is denoted by $e^{(i)} = (a^{(i)}, t_i)$ where $a^{(i)}$ denotes the one-hot encoding of activity a_i and the scalar timestamps $t^{(i)}$ are min-max scaled to the range of $[0, 1]$.

Sequential Models. We chose two different models to test transfer learning for suffix prediction: LSTM and GPT. These models were selected in order to have a comparison between a simple model and a model of higher complexity. We briefly review the architecture of the LSTM and GPT models. An **LSTM** is a type of recurrent neural network that improves the issue of vanishing gradients when back propagating. As other recurrent networks, it works in a autoregressive manner. It takes the previous output in combination with the current input, thus, making the network state dependent on previous input. LSTMs are comparably simple models and have been tested on the suffix prediction task in several publications [18–20]. The **GPT** model is a variant of a transformer model which takes only the decoder block of a transformer as its core. The model has been successfully used for sentence generation in natural language processing [21] and as well for next-event prediction in the context of predictive process monitoring [22]. Compared to the LSTM a transformer model can parallelise the sequence processing by avoiding the recurrence and use different position in the sequence to infer global dependencies between input and output rather than only keeping the "memory" through recurrence. We employ the same implementations of LSTM and GPT as in [4] and refer to [4] for the technical details.

Generator. The generator takes the output of the sequential model and produces multitask predictions. For each event in the sequence both the next activity and their time attribute are predicted, respectively. For the activity the model outputs a vector of logits $\hat{a}^{(i)}$ which are transformed by the softmax function. Then the most likely next activity by using the argmax or greedy search. Learning for the categorical feature is done via categorical cross-entropy and the loss function is then the average of these errors over all items in the sequence. As for the continuous time feature the learning is done via the squared error for which the loss function is the average of these errors over all items in the sequence. During inference the sequence generation is a loop where the next step is always conditioned on the previously generated sequence up until the $[EOS]$ is predicted or a predefined maximum length, which in our case is set to the length of the longest trace in the dataset.

3 Transfer Learning for Suffix Prediction

We describe how we used transfer learning in the framework for suffix prediction setting. First, a model needs to be trained on a base event log, which should be representative of the expected domain. We denote the trained model as base model and use it as basis for further training, i.e., transfer learning, on different event logs from different processes. Often the further training of the base model is constrained to certain parts of the model in a process denoted as fine tuning. The goal is to achieve better results for predictions on the other processes than when starting from a random initialization while spend less time on training. In general, transfer learning may also be used to address and entirely different

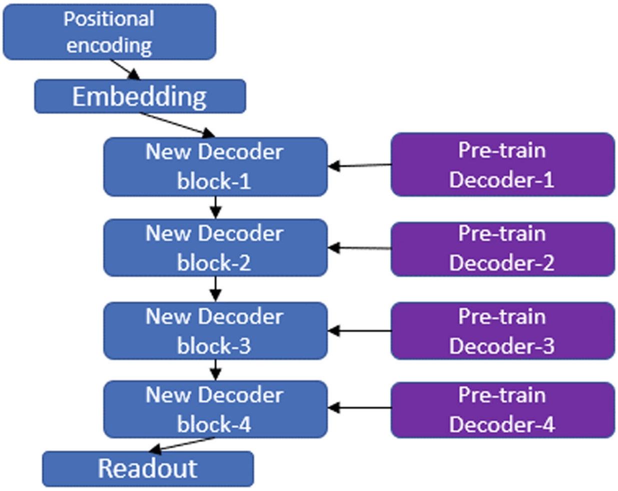


Fig. 2. Transferring the weights trained for the layers of the sequential model, here for the GPT architecture, to a new model with the embedding and readout layer trained from scratch.

task than for what the base model was trained. In the setting we investigated the task of suffix prediction remains fixed and only the target process changes.

However, even with the task of suffix prediction remaining the same, there are two issues that prevent direct usage of the trained base model:

1. Event logs from different process are defined over different sets of activities similar to the alphabet of different languages.
2. Events logs from different processes usually come from different domains with new process behavior that needs to be learned.

The first issue of processes being defined over *different activities* can be solved by re-training from scratch the entire embedding (input layer) and generator (readout layer) part of the network (cf. Fig. 1). Indeed, the dimensions of the embedding layer would not be compatible with the new alphabet of activities. This adapts the predictions to the new set of activities while the other layers, e.g., the decoder blocks of GPT in Fig. 2, can remain unchanged,

For the second issue of *different process behavior*, it seems useful to also retrain some of the layers of the actual sequential model (cf. Fig. 1) in the base model. This process is called *fine tuning* and enables the model to adapt to a new domain while keeping some of the learned distributions.

In our transfer learning experiment, a new model was instantiated for the event logs, each from a different process, with new input layer and readout layers matching the alphabets of the process. On this secondary model the original middle layers were laid over the randomly initialized parameters for input and readout layers. An example can be seen in Fig. 2 for the GPT architecture. In the framework of Ketyko et Al. [4], each sequential model consists of four layers or building blocks providing us the option to fine tune any combination of subsets of them.

Each of the four different layers can be trained or chosen to be not trained. When fine tuning, the layers that are not trained are denoted as being frozen. This makes a total of 16 different combinations for possible freezing. To limit the overall time of our experiments, in which we need to repeat the training for each of the event logs, we choose to explore seven out of these options. These combinations consisted of the combination of completely frozen layers, denoting the layers with $h^{(0,1,2,3)}$: $h^{(0,1,2,3)}$ meaning no retraining was done. All four of the different freeze three combinations: $h^{(0,1,2)}$, $h^{(0,1,3)}$, $h^{(0,2,3)}$ and $h^{(1,2,3)}$. Lastly for two and one layer freezing we tested combinations $h^{(0,1)}$ and $h^{(0)}$. The former layer combinations are interesting to investigate since with more layers frozen the retraining is cheaper. Whereas the latter may provide a higher chance of improving the model performance compared to training on random weights since earlier layers feeding into the model are retrained.

4 Evaluation

The setup of our experiment is described and the obtained results are discussed.

4.1 Experiment Setup

We evaluated the performance of the models on nine commonly used real-life event logs without any prepossessing each with different vocabulary size and event log properties as shown in Table 1. Our transfer learning method and our evaluation scripts are available online[1]. We focused on two event logs as dataset to create a pre-trained base model: the Helpdesk and Road Traffic Fine Management (RTFM) event logs. We choose those as both models performed well for them as reported in [4].

We used Damerau Levenstein Similarity (DLS) to evaluate the performance of suffix prediction. We used Mean Absolute Error (MAE) for the continuous time variables. We split the data into training and evaluation sets by 8:2 ratio after shuffling of full traces. This assumes independence of traces and the absence of concept drift in the process. Indeed, we are not aiming to predict, or claiming to be able to predict, such concept drifts. Please note that since we strictly restrict our model to be trained on the sequence of activities and relative time as features without using any log-level features, no data leakage from other traces can occur.

To obtain a base model for transfer learning, we trained the respective models on Helpdesk and RTFM for 400 epochs in the exact same manner with the same parameters as in [4]; however, with a changed random train/test split. Then, we transfer learnt models based on these pre-trained base models for Helpdesk

---

[1] https://zenodo.org/badge/latestdoi/527918382.

Table 1. Basic statistics of the used event logs for transfer learning.

Event log	Vocabulary size	Mean events per trace	Mean trace duration (hrs)
BPI12	36	20.1	19
BPI13_Closed	7	4.5	1968
BPI13_Open	5	2.9	41
BPI13_Incidents	13	8.7	180
BPIC_15	398	43.6	1473
BPI17	26	36.8	458
RTFM	11	4.3	4754
Sepsis	16	14.5	127
Helpdesk	9	3.6	93.6

Table 2. The best configurations for layer freezing when considering the DLS measure for the eight target event logs and two base models.

Event log	Base Model Helpdesk	Base Model RTFM
BPI12	$h^{0,1,2,3}$	$h^{0,1}$
BPI13_Closed	$h^{0,1,3}$	h^{0}
BPI13_Incidents	$h^{0,1,2}$	h^{0}
BPI13_Open	$h^{0,1,2}$	h^{0}
BPIC_15	$h^{0,2,3}$	$h^{0,1,2,3}$
BPI17	$h^{0,1}$	$h^{0,1}$
RTFM	$h^{0,2,3}$	–
Sepsis	$h^{1,2,3}$	$h^{0,1}$
Helpdesk	–	$h^{0,1}$

and RTFM. When transfer learning, we want to investigate if we can achieve a better performance in less time and, thus, we only use 200 epochs to fine tune the models on the other eight event logs. To make the training comparable, except for the training time, we used the exact same parameters as in the initial pre-training. This transfer learning is repeated with all the fine tuning combinations as indicated in Sect. 3. Finally, we compare the DLS and MAE scores achieved with less training effort (200 epochs) to those achieved when training for 400 epochs on randomly initialized models trained for each of the eight event logs.

4.2 Results

For the sake of space, we only report the best achieved DLS and MAE among the fine tuning configurations and list the best configuration of frozen layers for DLS in Table 2. Note that while in some cases there was considerable impact of the exact fine tuning configuration, the best configuration seems to be consistent for many of the logs. Transfer learning from Helpdesk required less layers to be fine tuned compared to RTFM.

Table 3. DLS (higher better) and MAE (lower better) obtained in regular training (400 epochs) compared to transfer learning (200 epochs) based on either the model pre-trained on RTFM or the model pre-trained on Helpdesk.

Log	Model	DLS			MAE		
		Regular	Helpdesk	RTFM	Regular	Helpdesk	RTFM
BPI12	GPT	**0.1606**	0.1586	0.1509	77.19	67.15	**54.83**
	LSTM	0.0622	**0.1377**	0.1188	101.78	**38.92**	68.44
BPI13_Closed	GPT	0.4712	0.6850	**0.7289**	585.01	111.47	**97.29**
	LSTM	0.5014	0.6959	**0.6976**	154.69	**95.70**	97.19
BPI13_Incidents	GPT	0.2445	**0.3082**	0.3021	76.94	26.83	**23.19**
	LSTM	0.2652	0.4893	**0.4902**	213.14	43.26	**19.39**
BPI13_Open	GPT	0.3450	**0.5157**	0.4171	105.17	52.55	**42.79**
	LSTM	0.4500	0.4918	**0.5341**	47.70	**47.31**	49.81
BPIC_15	GPT	**0.0255**	0.0213	0.0169	285.35	211.66	**164.65**
	LSTM	0.0239	0.0332	**0.0340**	2357.00	**331.32**	827.97
BPI17	GPT	**0.1428**	0.1424	0.1423	36.85	**32.50**	57.23
	LSTM	0.0549	**0.0598**	0.0460	36.44	**24.65**	35.72
Sepsis	GPT	0.0876	**0.1042**	0.1030	1078.09	**82.99**	142.14
	LSTM	0.0970	0.1536	**0.1992**	214.69	**33.18**	35.87
RTFM	GPT	0.8250	**0.8275**	–	158.02	**56.72**	–
	LSTM	0.5860	**0.7564**	–	535.44	**183.19**	–
Helpdesk	GPT	**0.8479**	–	0.8473	**4.13**	–	4.15
	LSTM	0.8556	–	**0.8599**	**4.01**	–	4.17

Table 3 compares the average DLS and average MAE score over all prefixes for the eight considered event logs between the regular models, starting from scratch, and the best transfer learned models, starting from the respective pre-trained base model (Helpdesk or RTFM). Regardless of being pre-trained on the Helpdesk or the RTFM event log, the average DLS score is better for most event logs when being trained for half of the epochs with transfer learning than without transfer learning. A notable exception to that is the *BPIC_15* event log for which the score is considerably worse in the GPT model. Also for BPI12, BPI17, and Helpdesk the transfer learning did not improve DLS with less training effort but achieved similar results as in 400 epochs training. Regarding the MAE score the obtained performance of the transfer learned models is even better with many of the MAE scores being improved considerably. Whereas the DLS scores are better or slightly better for the transfer learnt models both when using GPT or LSTM, the MAE metric scores are much better for the LSTM in the case of transfer learning.

4.3 Discussion

The results show that transfer learning improves the performance for most of the models with a smaller training time budget. We now discuss the results obtained.

For DLS, exceptions can be found in three event logs with most notably BPIC 15 that has a clear lower DLS than its base model. The models transfer learned from RTFM display results that are very similar to those pre-trained on Helpdesk. Again for most of the event logs the DLS increased with a decrease in the same event logs as well. The most reliable explanation for the difference in performance between different event logs is underlying process structure and data skewness. The three logs that saw a decrease in performance are all logs with a large vocabulary and often times complicated case structures in contrast to both the pre-train logs which feature more average vocabulary. The decreases in BPI 12 and BPI 17, which are less complex compared to BPIC 15, are also much more insignificant than those in BPIC 15. This in our opinion explains the difference between event logs. In conclusion, for less complex processes there is an increase in performance as these event logs align more with the logs used for pre-training whereas the more complex a process is the harder it becomes for a pre-trained model to capture its underlying process structure.

As for the MAE for all of the models the MAE decreases and in a few cases very substantially as well. For the RTFM pre-trained model, however, there is one exception where for the BPI17 set the MAE increased compared to the base model. The increase in performance for all of the event logs is rather surprising especially for the event logs with more complex underlying process structures. The most likely explanation for this is that as both Helpdesk and the RTFM have relatively low mean and median case durations it is better to predict shorter cases overall. This implies that the original model might not have been able to accurately capture that duration between events is likely shorter for events at the beginning of a case and instead structurally overestimated how much time was between cases. In conclusion the performance increases for each event log and no link can be made between increase and model complexity also implying that skewness of the event logs is of less importance to time prediction than to activity prediction.

When comparing the LSTM and the GPT scores, the scores for the base model are already worse compared to their GPT counterparts. The MAE scores show the same pattern as well, the scores for each of the event logs decreased and in some cases and in some cases to even less than 20% of their original base model as is the case for BPIC 15. In the transfer learning setting the better performance follows the same patterns as the GPT model: There is an increase in DLS performance for nearly all of the tested event logs with an exception of the BPI17 event log which does see a decrease in DLS albeit on the lower side. The MAE of the models achieved on most of the event logs decreases here as well with some event logs performing exceptionally well and making significant decreases in MAE as is the case for the BPI13_incidents set and the BPIC_15 event log. This seems to indicate that the LSTM model benefits exceptionally from the transfer learning setting.

We acknowledge limitations to our study that should be addressed in future work. We did not use hyper parameter tuning for the standard training and only use a single random training/evaluation split per model architecture. Due to the randomness of the training procedure and the dependence on particular training/evaluation splits further experiments are needed to show that our results also hold for different data splits. Still, our results are consistent for two different model architectures each using their own data split.

5 Conclusion

We investigate the effectiveness of transfer learning to reduce training time and improve performance for the suffix prediction task on event logs. Our experiments on eight real-life event logs and several transfer learning scenarios with the GPT and LSTM models show an increase in performance using less training time for most of the event logs for prediction of both the activity and time of suffixes. In the remaining case, the performance was on par with the performance of models trained for double the training time budget. An exception to this finding is the BPIC_15 data set that performs significantly worse on the GPT model for all of the transfer learning scenarios. This may be caused by very different prefix length distributions and much larger alphabet size of BPIC_15 compared to the datasets used to pre-train. Based on these initial experiments, we conclude that with a significant decrease of computing costs, at least halving, results stay equal or better for a majority of event logs. This seems to indicate that transfer learning is helpful for suffix prediction on event logs even across domain when, e.g., pre-training on RTFM and transfer learning to the Sepsis event log.

In future work, we need to address the clear limitations of our study such as investigating the effect of parameter tuning and testing if the results are consistent across multiple random training/evaluation splits. More research is required to investigate the impact of which layers to fine-tune and how that choice interacts with other hyper parameters. Finally, it would be interesting to systematically study the process information captured by the different layers to understand which event log properties impact the effectiveness of transfer learning.

References

1. Francescomarino, C.D., Ghidini, C.: Predictive process monitoring. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 320–346. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_10
2. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: review and benchmark. ACM Trans. Knowl. Discov. Data **13**(2), 17:1–17:57 (2019)

3. Pasquadibisceglie, V., Appice, A., Castellano, G., Malerba, D.: A multi-view deep learning approach for predictive business process monitoring. IEEE Trans. Serv. Comput. **15**(4), 2382–2395 (2022)
4. Ketykó, I., Mannhardt, F., Hassani, M., van Dongen, B.F.: What averages do not tell: predicting real life processes with sequential deep learning. In: SAC, pp. 1128–1131. ACM (2022)
5. Neu, D.A., Lahann, J., Fettke, P.: A systematic literature review on state-of-the-art deep learning methods for process prediction. Artif. Intell. Rev. **55**(2), 801–827 (2022)
6. Rama-Maneiro, E., Vidal, J., Lama, M.: Deep learning for predictive business process monitoring: review and benchmark. IEEE Trans. Serv. Comput. (2021)
7. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. J. Big data **3**(1), 1–40 (2016)
8. Zhuang, F., et al.: A comprehensive survey on transfer learning. Proc. IEEE **109**(1), 43–76 (2020)
9. Duan, L., Xu, D., Tsang, I.: Learning with augmented features for heterogeneous domain adaptation. arXiv preprint arXiv:1206.4660 (2012)
10. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: domain adaptation using asymmetric kernel transforms. In: CVPR 2011, pp. 1785–1792. IEEE (2011)
11. Wang, C., Mahadevan, S.: Heterogeneous domain adaptation using manifold alignment. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)
12. Zhou, J., Pan, S., Tsang, I., Yan, Y.: Hybrid heterogeneous transfer learning through deep learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
13. Tong, L., Weijian, N., Yujian, S., Qingtian, Z.: Predicting remaining business time with deep transfer learning. Data Anal. Knowl. Discov. **4**(2/3), 134 (2020)
14. Chen, H., Fang, X., Fang, H.: Multi-task prediction method of business process based on BERT and transfer learning. Knowl.-Based Syst. **254**, 109603 (2022)
15. Ni, W., Yan, M., Liu, T., Zeng, Q.: Predicting remaining execution time of business process instances via auto-encoded transition system. Intell. Data Anal. **26**(2), 543–562 (2022)
16. Peeperkorn, J., vanden Broucke, S., De Weerdt, J.: Can deep neural networks learn process model structure? An assessment framework and analysis. In: Munoz-Gama, J., Lu, X. (eds.) ICPM 2021. LNBIP, vol. 433, pp. 127–139. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98581-3_10
17. Augusto, A., Mendling, J., Vidgof, M., Wurm, B.: The connection between process complexity of event sequences and models discovered by process mining. Inf. Sci. **598**, 196–215 (2022)
18. Evermann, J., Rehse, J., Fettke, P.: Predicting process behaviour using deep learning. Decis. Support Syst. **100**, 129–140 (2017)
19. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 477–492. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_30
20. Camargo, M., Dumas, M., González-Rojas, O.: Learning accurate LSTM models of business processes. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 286–302. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_19

21. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training. Technical report, OpenAI (2018)
22. Moon, J., Park, G., Jeong, J.: Pop-on: prediction of process using one-way language model based on NLP approach. Appl. Sci. **11**(2), 864 (2021)

A Case for Business Process-Specific Foundation Models

Yara Rizk[1](✉), Praveen Venkateswaran[1], Vatche Isahagian[1],
Austin Narcomey[2], and Vinod Muthusamy[1]

[1] IBM Research, Cambridge, USA
{yara.rizk,praveen.venkateswaran,vatchei}@ibm.com, vmuthus@us.ibm.com
[2] Stanford University, Stanford, USA
anarc@cs.stanford.edu

Abstract. The inception of large language models has helped advance the state-of-the-art on numerous natural language tasks. This has also opened the door for the development of foundation models for other domains and data modalities (e.g., images and code). In this paper, we argue that business process data has unique characteristics that warrant the creation of a new class of foundation models to handle tasks like activity prediction, process optimization, and decision making. These models should also tackle the challenges of applying AI to business processes which include data scarcity, multi-modal representations, domain specific terminology, and privacy concerns. To support our claim, we show the effectiveness of few-shot learning and transfer learning in next activity prediction, crucial properties for the success of foundation models.

Keywords: Business Process · Foundation Models · Few-shot Learning · Transfer Learning

1 Introduction

Artificial intelligence, particularly with the emergence of deep learning, has disrupted many areas of our lives, from personal assistants like Alexa [39] to autonomous driving [5]. It has also been a disruptive force for businesses[1] deep learning is estimated to provide $3.5T+ of annual value [12] and can be the difference between companies' rise or demise.

Business processes provide a structured framework for work. They define tasks and identify their executors while capturing dependencies and providing logging and tracking capabilities. They also capture company policies and ensure compliance with regulations. With many enterprises relying on the business pro-

---

[1] https://www.gartner.com/smarterwithgartner/the-disruptive-power-of-artificial-intelligence.

His contributions were completed while he was an intern at IBM Research.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 44–56, 2024.
https://doi.org/10.1007/978-3-031-50974-2_4

cess management paradigm to standardize their work, process management tools grew to a \$11.84B industry and are projected to grow to \$26B in 2028[2]

However, the existing landscape of work has been rapidly changing, requiring companies to move from their static business process practices to more agile and automated methods due to increased supply chain disruptions and skill shortages from the recent pandemic. Thus, companies are making significant investments in AI-driven automation tools (e.g., process prediction, visualization, translation, etc. [27]), evidenced by companies' \$1M+ investments[3] and AI's projected \$3.2T+ business value [33]. Foundation models, the latest machine learning paradigm based on very deep neural networks, present an opportunity to improve business process automation and management.

Similar to natural language, images, or code snippets, business processes are yet another information representation paradigm. However, the unique nature of process features and modalities can render existing foundation models inadequate to accurately understand and reason over business processes. Hence, developing successful process-specific foundation models requires research efforts to treat process data holistically instead of separate, independent modalities.

In this paper, we propose creating foundation models that factor in the complexity of process data. We show that knowledge learned from one business process domain can transfer to others while requiring less data to train neural networks. This provides strong evidence of the effectiveness of deep networks, and specifically foundation models, on multiple business process tasks. We also discuss some of the challenges of creating foundation models for business processes and the risks and opportunities of foundation models' emergent behavior. First, however, we provide an overview of business processes, their unique properties and the tasks[4] that may be best suited for foundation models.

2 Background

Business Process Management. A *business process* is a collection of ordered *activities* that results in a product or a service [48]. Figure 1 shows a mortgage application process that takes every application through the same steps before making a decision. Processes introduce structure, improve consistency, and track the execution of activities for accountability, auditability and improvement over time. A graphical notation, e.g., business process model and notation (BPMN) [18], is generally used to represent such processes, capturing the relationship between activities (round-edged rectangles) performed by employee *roles* within

---

[2] https://www.marketwatch.com/press-release/business-process-management-market-size-growth-with-top-leading-players-growth-key-factors-global-trends-industry-share-and-forecast-2022-2031-2022-08-18.

[3] https://www.gartner.com/en/newsroom/press-releases/2021-09-29-gartner-finds-33-percent-of-technology-providers-plan-to-invest-1-million-or-more-in-ai-within-two-years.

[4] To avoid confusion with business process tasks or activities, we will use "downstream tasks" to refer to foundation model specific prediction tasks.

an organization, *events* (circles) that trigger processes or activities within them, and *decision points* (diamonds) that allow paths within the process flow to merge or diverge. *Swim lanes* place specific tasks within the scope of an employee role or department. A *trace* is an execution of a process; each process can produce many distinct traces when executed depending on input events and other factors.

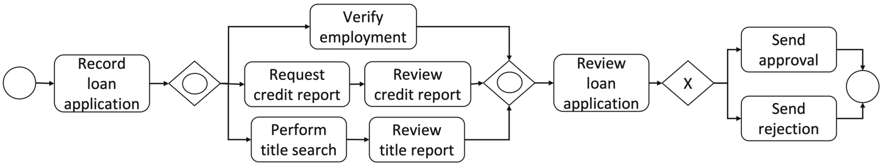


Fig. 1. Example of a mortgage loan application process (Source: [10])

Business process management consists of many problems related to the modeling or design, execution and governance of processes. Process mining leverages data analytics to improve process management. Process discovery analyzes event data to identify and derive processes from raw, unstructured data [42]. Ideally, process discovery should produce a BPMN or similar representation for the discovered process. Process optimization or re-engineering looks to improve existing processes [3]. This requires making changes to the process representation while maintaining the properties that characterize a valid process. Conformance checking verifies that the "as-is" process (i.e., how the process is being executed in reality) does not deviate from the "to-be" process (i.e., how the process was theoretically designed to be executed) [15]. Task automation through robotic process automation looks to create automation scripts that can programmatically execute tasks instead of humans [1], whereas automation mining programmatically identifies the best tasks to automate [17].

Foundation Models. Foundation models, coined in [7], refer to deep neural network models trained on massive data and can be reused (with minimal modifications) for multiple downstream tasks. They are characterized by "emergent" knowledge: the model is able to make predictions and perform downstream tasks that it has never seen before and wasn't trained on. Large language models were the first set of foundation models; trained on billions of English text from the internet, they learned the structure of language and became capable of performing natural language understanding and generation tasks [8,22]. This has been followed by a wave of new foundation models catering to problems across different domains such as vision [36], programming code, clinical and biomedical applications [2], among others. After training foundation models (generally) in an unsupervised or self-supervised paradigm, one of two approaches can be taken to use the model for a specific task. Either fine-tune it on a small set of labeled data or create a prompt from labeled data along with the input you want a prediction for. Both approaches have their pros and cons and have spurred many new open research questions and sub-fields of AI (e.g., prompt engineering [24]).

3 A Business Process Foundation Model

3.1 Overview

The business process management literature is already rich with machine learning solutions to improve business processes (e.g., [31]). Deep learning algorithms specifically have proven effective on multiple business process tasks including anomaly detection [19], process monitoring [23], process prediction [28], process querying [21], performance prediction [32], and decision making [41].

However, these works took a narrow view of the task and produced narrow solutions that would not realize the full potential of AI. Instead of training specialized networks on only one task, Chen et al. trained a multi-task prediction model for next-activity prediction on eleven real-world domains [11]. A bidirectional encoder representation with transformers architecture [22] was pretrained using a masked activity model and transfer learning on completed case logs. It outperformed other approaches from the literature and showed the benefits of training larger models on business process data. Therefore, generalizing this approach to many other tasks by training foundation models to understand the fundamental concepts of business process could be the key to unlocking the potential of neural networks in the field of business process.

In natural language processing, foundation models learned the building blocks of language. A finite number of letters can make up words; not all letter sequences produce valid words. Sentences are composed of word sequences that must abide by the syntactic structures imposed by language. Words play specific syntactic and semantic roles within sentences and can have varying semantic meanings based on context. Sentences also convey a semantic meaning that must be understood by the entity (person or otherwise) decoding the sentence.

Similarly, in the field of business process, foundation models need to learn about process artifacts, notation, and properties. Furthermore, intra- and inter-process features have been shown to affect the prediction [40]. Once a deep learning network internalizes all these concepts, then we can perform more complex downstream tasks that rely on this foundational understanding like optimizing processes or discovering them from unstructured data and events.

3.2 Data Types in Business Processes

The data describing business processes and generated from their execution consists of many different types of data. Whether we consider business process data to be a new modality [10] or treat it as a multi-modal problem, we must first understand what types exist before we can effectively train foundation models.

The first type of data embodied in a business process is a graph which represents the control flow of a process where activities and decision points are connected to form a directed graph with cycles, branches, root nodes and end nodes [49]. Once a process is executed, a sequence of events (or process trace) is generated. One process may have many different traces representing the various traversals of the graph and different decisions at decision points.

Processes and their events also have metadata associated with them which are generally represented by a multi-dimensional set of attributes that can be binary, categorical or continuous. For example, activities are typically associated with human workers (e.g., loan officer, or claims processor) from the enterprise organization, who are geographically distributed across timezones with varying vacation and holiday schedules. These human workers cannot work on two process cases at once, which in turn creates an implicit limit on the number of associated concurrent tasks across process instances. Events and tasks within a process can also have unstructured documents associated with them (e.g., images, text, video, audio). Interactions between participants (including social networks) in a process can be represented by graphs and times series data.

Considering only a subset of data would provide an incomplete view of the business process and may lead to sub-optimal predictions by machine learning models. Thus, it is important to identify effective approaches to handle these types of complex applications and interactions with diverse data types.

3.3 Downstream Tasks

We distinguish between two types of downstream tasks for foundation models: domain agnostic vs. domain specific. The former can be process mining, process optimization, trace prediction, etc. The latter can be process task prediction, decision recommendation at a decision point, automation of process tasks, etc. Furthermore, some of these downstream tasks can be time sensitive vs. not. For example, identifying a process from unstructured data can be performed offline. However, a decision making step during process execution is more time sensitive; the foundation model needs to make a decision within seconds or minutes as opposed to hours or days. The type of downstream tasks may impose computational or architectural constraints on the foundation models.

3.4 Model Architecture

Past work on pre-training multi-modal foundation models have focused on vision-language tasks to learn cross-modal representations of data. These efforts primarily use one of two architectures. The first is a *dual-encoder* architecture to encode different data modalities separately, and then use cosine-similarity of the feature vectors for modality interaction. This shallow interaction between modalities has been shown to perform poorly on several tasks [20]. The second is a *fusion-encoder* architecture with cross-modal attention, to jointly encode all possible data pairs to compute similarity scores for tasks. This results in a quadratic (for two modalities) time complexity and much slower inference speed than dual-encoder models whose time complexity is linear.

We envision leveraging a recent approach, called Mixture-of-Modality-Experts (MoME) [46,47], that uses a pool of modality models to replace the feed-forward network in a standard transformer architecture. It switches between different modality models to capture modality-specific information, and then uses shared self-attention across modalities to align information. Figure 2 describes

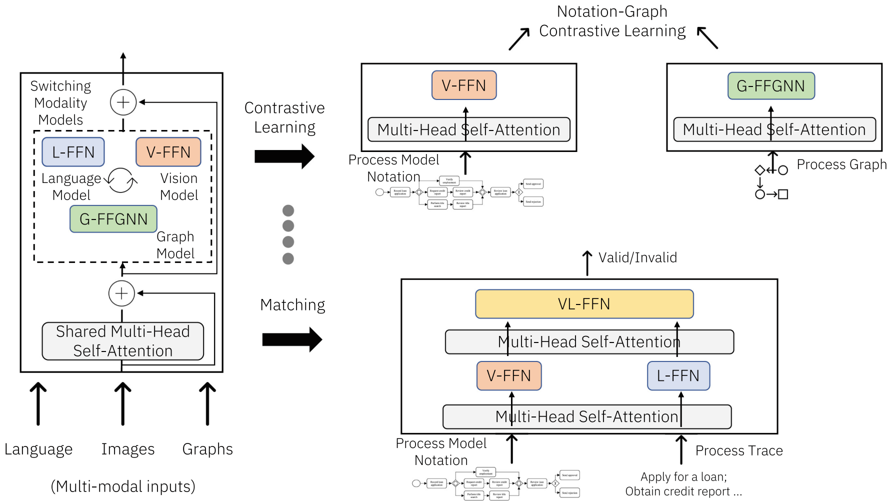


Fig. 2. Using Mixture of Modality Experts (MoME) Transformer to pre-train a Business Process Foundation Model on different tasks

our vision for implementing MoME for business process tasks, where we define expert feed-forward network and feed-forward graph neural network models for different modalities (language, vision, graph) and their combinations.

The transformer selects the appropriate mixture of expert models based on the modality of the input vectors. For example, a graph model processes a BPMN diagram. Traditional pre-training tasks like contrastive learning, masking, or matching, can be performed to capture cross-modal information in the business process context, and we show an example of two pre-training tasks in Fig. 2.

4 Evidence of Few-Shot and Transfer Learning

Foundation models are more effective when knowledge is transferable across tasks and domains, relying on transfer learning and few-shot learning to achieve these performance gains. Next, we show empirical evidence of the effectiveness of transfer learning and few-shot learning in the business process space. We focus on next activity prediction using long short-term memory networks (LSTMs), a popular neural network architecture in process prediction tasks [37], more suitable than other models for the small size of our datasets.

We rely on an architecture that mainly borrows from [9,44,45]. The LSTM has one hidden layer of size 100 followed by a linear layer of size 100 with ReLU activation. The input, a partial trace, is represented using one-hot encoding or embedding vectors. The model predicts the next activity, the time to the next activity and the final case outcome. We experiment on three datasets: 1) synthetic loan process [25], 2) BPIC 2015 [13], and 3) BPIC 2018 [14].

Tables 1 and 2 display the test accuracy of LSTM models trained on an increasing size of randomly sampled instances from the datasets. As little as 20 samples on the synthetic datasets (ROI - Remove, Optionalize, Insert and IOR - Insert, Optionalize, Remove variants) and 100 samples on the real-world datasets were able to achieve near optimal performance.

Table 1. Next Activity Prediction Accuracy on Synthetic Data

Dataset	Number of Samples										
	1	2	3	4	5	10	20	50	100	250	500
Synthetic-ROI	0.52	0.75	0.81	0.82	0.85	0.83	0.92	0.95	0.95	0.95	0.95
Synthetic-IOR	0.48	0.68	0.69	0.68	0.67	0.81	0.88	0.90	0.90	0.90	0.90

Table 2. Next Activity Prediction Accuracy on Real-world Datasets

Dataset	Number of Samples				
	10	100	250	500	1000
BPIC 2015	0.22	0.5	0.58	0.63	0.65
BPIC 2018	0.58	0.75	0.79	0.81	0.82

To demonstrate transfer learning on business process data, we trained the LSTM on one dataset (e.g., base or IOR) and then used that model's weights as the initial weights of the model trained on a second dataset (e.g., ROI) instead of randomly initializing the weights of the latter model. Transfer learning is successful when the latter model requires less training samples to outperform a model trained from scratch. Figure 3 displays the test set accuracy of LSTM models trained: (1) from scratch (i.e., randomly initialized weights), (2) from the base definition of the synthetic dataset, and (3) from the synthetic IOR dataset. We observe that transfer from the base definition accelerates learning on the smallest datasets, but for sufficient data, transfer learning conflicts weaken performance. Also, the additional transfer differences of IOR eliminate the benefits of transfer learning but transfer gains the most on rare activities.

In summary, we observed that surprisingly little data is needed to match peak performance. The diversity of the smaller dataset is crucial for learning since they are more comprehensive of the process and enable learning from fewer cases. Transfer from a related or similar processes also accelerates learning. Finally, domain knowledge such as process definitions, sequence constraints and biases can reduce few-shot learning error with and without transfer learning. Since foundation models leverage transfer learning (during fine-tuning) and few-shot learning (during prompting), these findings support our hypothesis of the promise of foundation models trained on business process-specific data and downstream tasks.

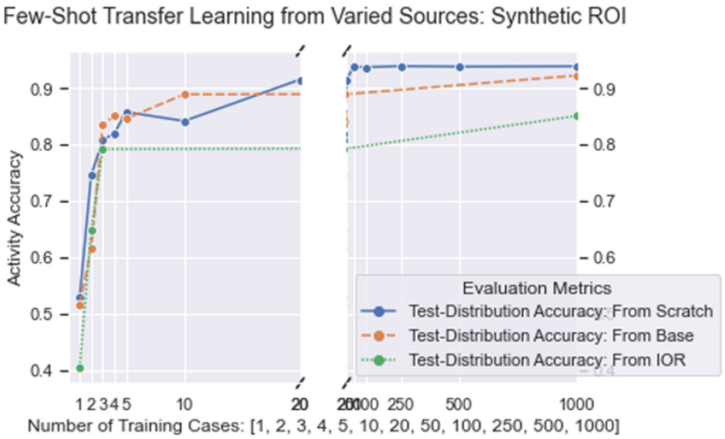


Fig. 3. Next Activity Prediction Accuracy - Evidence of Transfer Learning

5 Challenges

Data Scarcity and Privacy Concerns. A majority of foundation model training efforts consider tasks involving the generation of natural language (e.g., OpenAI GPT-3, Google T5), images (e.g., DALL.E 2) or code syntax (e.g., GitHub Copilot, Amazon CodeWhisperer) which have large amounts and variety of relevant and (un)labeled training data that has been collected and open-sourced by the larger research community. However, business processes suffer from a lack of sufficient labeled open-source real-world data to train foundation models due to the inherent proprietary nature of business processes, resulting in most corporations being unwilling to share their data and models [29,43].

Creating foundation models for business processes would require addressing the critical challenge of data availability. Privacy-preserving training such as federated learning approaches could help, wherein models can be trained on data across multiple business units and corporations without involving any data sharing. Another solution is data generation and augmentation techniques to leverage patterns from the literature (e.g., insertion of new tasks in the process, optionalization of tasks that were previously required in the process, and resequentialization of tasks [26]) or ones existing in the data to create realistic new process data. Generative models could create new data instances hallucinated from existing processes. However, such approaches would also require crowd-sourced data validation (by subject matter experts) and labeling efforts to ensure training data quality.

Breadth and Evolution of Processes. As with other domains, business process mining, monitoring, and automation can comprise of a multitude of possible tasks. These could be (1) process predictions – such as predicting a future process sequence given a partial trace, process completion time, process failures, etc.,

(2) process synthesis – including synthesising new process models from specifications or natural language input, process visualizations, etc., (3) explainability and summarization – wherein models explain various decisions and predictions, as well as provide accurate summaries of process traces, among other tasks.

These tasks involve different data modalities and input/output structures. Some of these tasks operate using text, some with images, and others with graphs. Hence, training a singular foundation model across these different tasks is a significant challenge. Determining the appropriate model parameters in this situation would require techniques like meta-learning [16] to ensure minimal additional training to perform well on different downstream tasks.

Furthermore, processes change over time, whether gradually or abruptly, as business evolve and adapt to changes in the environment. Also, errors could be identified in processes and iteratively improved as lessons are learned from running the processes. Thus, foundation models need to evolve and adapt efficiently to these process changes.

Domain Specific Language. Tasks based on natural language have well-defined language constructs and semantic meaning for models to reason on. However, business processes often have acronyms and technical phrases which are not common knowledge, but are critical for the model to understand. Further, process models often adhere to different standards and graphical notations (e.g., BPMN, Decision Model and Notation, etc. [49]).

Hence, it is critical to develop a domain specific language (DSL) for foundation models to reason over such business process specific terminology. Such a DSL would also enable users to enforce business policies and ensure the validity of the model outputs using techniques such as constrained semantic decoding [34]. However, the number of business domains and terminology is ever-increasing and nearly impossible to fully capture. This would result in situations where the model has limited knowledge or information, reflecting zero-shot or few-shot settings, that would require approaches like prompt-based fine-tuning of the model.

Prompt Engineering for Business Processes. Many real-world tasks may have very little to no data available to fine-tune foundation models. However, the use of prompts and in-context examples have enabled language models to perform better in zero-shot and few-shot settings [35]. The popularity of language tasks has even resulted in a public repository of natural language prompts [4].

While prompting has demonstrably improved performance, foundation models have also been extremely sensitive to prompt engineering, e.g., small changes to the prompt such as changing the prompt structure, reordering, and even the number of examples, can significantly reduce the model's performance [30].

This presents several challenges for business process models. Firstly, while the structure of prompts may often be straightforward for language tasks, this is not the case for many business process tasks. For instance, tasks involving the translation of natural language specifications to process models or summarizing process models using text, would require careful prompt engineering. Prompts in the business process domain can involve images or even graph structures, and identifying the most relevant examples or prompts also presents a challenge.

Human-in-the-Loop Feedback and Model Robustness. Many process automation tasks involve critical decision-making steps. The sensitive and regulated nature of business domains often results in the inclusion of human feedback in the decision making pipeline. This feedback could involve enforcing corporate policies, validating model outputs, changing intermediate decisions of the process pipeline, among others. Hence, process models would require an optimized approach to incorporate such human-in-the-loop feedback. Since fine-tuning large foundation models is an expensive process, it may not always be possible to continually update the model parameters with user feedback, thereby requiring approaches to incorporate the feedback within subsequent input prompts.

Additionally, the influence of malicious actors and data biases on model decisions can have a significant and costly impact on businesses. For instance, adversarial prompts and feedback could be used to bias the model to output incorrect or inappropriate decisions, or even obtain any confidential information used to train or fine-tune the model [7]. Hence, approaches to improve model robustness are critical for business process tasks. For instance, coupling constrained decoding with model outputs, where businesses can explicitly specify guardrails or policies [38], and careful consideration of data biases, distribution shifts, and information leakage during the pre-training process are important.

6 Risks, Opportunities, and Next Steps

The emergent behavior of foundation models has been a point of intrigue and concern in various fields (e.g., healthcare [50], education [6]); for business processes, things are no different. On the one hand, as foundation models become capable of generating, modifying and executing parts of a process, concerns around violating industry standards or company policies, auditability and interpretability must be addressed to ensure wide-spread adoption. On the other hand, generative models can unlock process optimizations and create new ways to work to achieve profitability without sacrificing sustainability goals. They can help make data-driven decision making a reality for business processes.

In summary, we believe that foundation models for business processes have tremendous potential to advance the field of process management and integrate AI into their practices. Both AI and BPM communities need to join forces to create the proper infrastructure to train and use such foundation models. Next steps for the community include identifying existing data sources and curating specialized datasets for training and fine-tuning. Safeguards should also be put in place to ensure that foundation models' emergent behavior does not have negative side-effects that may hinder its adoption in industry.

References

1. Van der Aalst, W.M., Bichler, M., Heinzl, A.: Robotic process automation (2018)
2. Alsentzer, E., et al.: Publicly available clinical BERT embeddings. arXiv preprint arXiv:1904.03323 (2019)
3. Arlbjørn, J.S., Haug, A.: Business Process Optimization. Academica (2010)
4. Bach, S.H., et al.: PromptSource: an integrated development environment and repository for natural language prompts. arXiv preprint arXiv:2202.01279 (2022)
5. Bernhart, W., Winterhoff, M.: Autonomous driving: disruptive innovation that promises to change the automotive industry as we know it. In: Langheim, J. (ed.) Energy Consumption and Autonomous Driving. LNM, pp. 3–10. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-19818-7_1
6. Blodgett, S.L., Madaio, M.: Risks of AI foundation models in education. arXiv preprint arXiv:2110.10024 (2021)
7. Bommasani, R., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
8. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
9. Camargo, M., Dumas, M., González-Rojas, O.: Learning accurate LSTM models of business processes. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 286–302. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_19
10. Chakraborti, T., et al.: From robotic process automation to intelligent process automation. In: Asatiani, A., et al. (eds.) BPM 2020. LNBIP, vol. 393, pp. 215–228. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58779-6_15
11. Chen, H., Fang, X., Fang, H.: Multi-task prediction method of business process based on BERT and transfer learning. Knowl.-Based Syst. **254**, 109603 (2022)
12. Chui, M., Henke, N., Miremadi, M.: Most of AI business uses will be in two areas. Harv. Bus. Rev. **20** (2018)
13. van Dongen, B.B.: BPI challenge 2015 (2015)
14. van Dongen, B., Borchert, F.F.: BPI challenge 2018 (2018)
15. Dunzer, S., Stierle, M., Matzner, M., Baier, S.: Conformance checking: a state-of-the-art literature review. In: 11th International Conference on Subject-Oriented Business Process Management (2019)
16. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning, pp. 1126–1135. PMLR (2017)
17. Geyer-Klingeberg, J., Nakladal, J., Baldauf, F., Veit, F.: Process mining and robotic process automation: a perfect match. In: BPM (2018)
18. Grosskopf, A., Decker, G., Weske, M.: The Process: Business Process Modeling Using BPMN. Meghan Kiffer Press (2009)
19. Huo, S., Völzer, H., Reddy, P., Agarwal, P., Isahagian, V., Muthusamy, V.: Graph autoencoders for business process anomaly detection. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 417–433. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_26
20. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML, pp. 4904–4916. PMLR (2021)
21. Kecht, C., Egger, A., Kratsch, W., Röglinger, M.: Quantifying chatbots ability to learn business processes. Inf. Syst. **113**, 102176 (2023)

22. Kenton, J.D., Chang, M.W., Toutanova, L.K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the NAACL-HLT (2019)
23. Kratsch, W., Manderscheid, J., Röglinger, M., Seyfried, J.: Machine learning in business process monitoring: a comparison of deep learning and classical approaches used for outcome prediction. Bus. Inf. Syst. Eng. **63**, 261–276 (2021)
24. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021)
25. Maaradji, A., Dumas, M., La Rosa, M., Ostovar, A.: Fast and accurate business process drift detection. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) BPM 2015. LNCS, vol. 9253, pp. 406–422. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_27
26. Maaradji, A., Dumas, M., La Rosa, M., Ostovar, A.: Detecting sudden and gradual drifts in business processes from execution traces. IEEE Trans. Knowl. Data Eng. **29**(10), 2140–2154 (2017)
27. McKendrick, J.: AI adoption skyrocketed over the last 18 months. Harv. Bus. Rev. (2021)
28. Mehdiyev, N., Evermann, J., Fettke, P.: A novel business process prediction model using a deep learning method. Bus. Inf. Syst. Eng. **62**, 143–157 (2020)
29. Mendling, J.: Advancing business process science via the co-evolution of substantive and methodological knowledge. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds.) BPM 2022. LNCS, vol. 13420, pp. 3–18. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16103-2_1
30. Min, S., et al.: Rethinking the role of demonstrations: what makes in-context learning work? arXiv preprint arXiv:2202.12837 (2022)
31. Nguyen, P., Isahagian, V., Muthusamy, V., Slominski, A.: Summarizing process traces for analysis tasks: an intuitive and user-controlled approach. In: International Joint Conference on Artificial Intelligence (2022)
32. Park, G., Song, M.: Predicting performances in business processes using deep neural networks. Decis. Support Syst. **129**, 113191 (2020)
33. Pettey, C., van der Meulen, R.: Gartner says global artificial intelligence business value to reach $1.2 trillion in 2018 (2018)
34. Poesia, G., et al.: Synchromesh: reliable code generation from pre-trained language models. arXiv preprint arXiv:2201.11227 (2022)
35. Radford, A., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
36. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: ICML, pp. 8748–8763. PMLR (2021)
37. Rama-Maneiro, E., Vidal, J., Lama, M.: Deep learning for predictive business process monitoring: review and benchmark. IEEE Trans. Serv. Comput. (2021)
38. Rizk, Y., Venkateswaran, P., Isahagian, V., Muthusamy, V., Talamadupula, K.: Can you teach robotic process automation bots new tricks? In: Marrella, A., et al. (eds.) BPM 2022. LNBIP, vol. 459, pp. 246–259. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16168-1_16
39. Schneider, F.: How users reciprocate to Alexa. In: Stephanidis, C., Antona, M., Ntoa, S. (eds.) HCII 2020. CCIS, vol. 1293, pp. 376–383. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60700-5_48

40. Senderovich, A., Di Francescomarino, C., Ghidini, C., Jorbina, K., Maggi, F.M.: Intra and inter-case features in predictive process monitoring: a tale of two dimensions. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNCS, vol. 10445, pp. 306–323. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_18
41. Tian, X., Pavur, R., Han, H., Zhang, L.: A machine learning-based human resources recruitment system for business process management: using LSA, BERT and SVM. Bus. Process. Manag. J. **29**(1), 202–222 (2022)
42. Van Der Aalst, W.: Process mining: overview and opportunities. ACM Trans. Manage. Inf. Syst. **3**(2), 1–17 (2012)
43. Venkateswaran, P., Isahagian, V., Muthusamy, V., Venkatasubramanian, N.: FedGen: generalizable federated learning for sequential data. In: IEEE International Conference on Cloud Computing (2023)
44. Venkateswaran, P., Muthusamy, V., Isahagian, V., Venkatasubramanian, N.: Environment agnostic invariant risk minimization for classification of sequential datasets. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1615–1624 (2021)
45. Venkateswaran, P., Muthusamy, V., Isahagian, V., Venkatasubramanian, N.: Robust and generalizable predictive models for business processes. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 105–122. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_9
46. Wang, W., Bao, H., Dong, L., Wei, F.: VLMo: unified vision-language pre-training with mixture-of-modality-experts. arXiv preprint arXiv:2111.02358 (2021)
47. Wang, W., et al.: Image as a foreign language: BEiT pretraining for all vision and vision-language tasks. arXiv preprint arXiv:2208.10442 (2022)
48. Weske, Mathias: Business process management methodology. In: Weske, M. (ed.) Business Process Management, pp. 373–388. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28616-2_8
49. White, S.A.: Introduction to BPMN. IBM Coop. **2** (2004)
50. Wiggins, W.F., Tejani, A.S.: On the opportunities and risks of foundation models for natural language processing in radiology. Radiol.: Artif. Intell. **4**(4), e220119 (2022)

Using Reinforcement Learning to Optimize Responses in Care Processes: A Case Study on Aggression Incidents

Bart J. Verhoef and Xixi Lu(✉)

Utrecht University, Utrecht, The Netherlands
b.j.verhoef@students.uu.nl, x.lu@uu.nl

Abstract. Previous studies have used prescriptive process monitoring to find actionable policies in business processes and conducted case studies in similar domains, such as the loan application process and the traffic fine process. However, care processes tend to be more dynamic and complex. For example, at any stage of a care process, a multitude of actions is possible. In this paper, we follow the reinforcement approach and train a Markov decision process using event data from a care process. The goal was to find optimal policies for staff members when clients are displaying any type of aggressive behavior. We used the reinforcement learning algorithms Q-learning and SARSA to find optimal policies. Results showed that the policies derived from these algorithms are similar to the most frequent actions currently used but provide the staff members with a few more options in certain situations.

Keywords: prescriptive process mining · reinforcement learning · Markov decision process · process optimization · process mining

1 Introduction

Prescriptive process monitoring focuses on analyzing process execution data to not only predict the future behavior of a process but also provide actionable recommendations or interventions to optimize the process [1–3]. It goes beyond *descriptive* or *predictive process monitoring* by actively suggesting specific actions or decisions for improving process performance, compliance, or efficiency. Considering the decision points in business processes, the ability to offer specific guidance to users regarding optimal actions is crucial, as it can lead to improved decision-making and efficiency.

One prominent approach is to use reinforcement learning, which learns online by interacting with an environment to adapt and improve its recommendations over time. The environments can be learned and built using the historical execution traces and the feedback they received. While reinforcement learning methods have been applied in business processes, healthcare processes exhibit distinct characteristics and present new challenges for these techniques [4], such as dynamic workflows, diverse stakeholders, and patient safety considerations. In particular, patients may exhibit very diverse statuses, and a wide range of actions is possible at any stage. Moreover, each patient

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 57–69, 2024.
https://doi.org/10.1007/978-3-031-50974-2_5

may react differently to these actions. These challenges may cause RL methods not to converge or not be able to improve the current policies. In such dynamic settings, it is worth investigating the validity and effectiveness of the RL approaches.

In this paper, we focus on the healthcare domain, and in particular, the process of actions and responses in the aggression incidents by clients with intellectual impairments in community care facilities [5]. Being in aggressive situations can have a severe impact on staff members since there is a mediation effect between experiencing aggressive behavior from clients and burnout through fear of assault [6]. This means that experiencing aggressive behavior leads to fear of assault, which in turn leads to burnout. It also has a negative impact on the clients themselves because aggressive behavior can lead to more aggressive behavior [7]. Therefore, learning the optimal way to act during aggression incidents helps de-escalate the incidents and reduce negative impact.

Previous studies have analyzed the aggression incidents of such clients within Dutch residential care facilities using a process mining approach [8] or proposing to mine potential causal patterns [9–11]. This meant that insights into the use of actions and their effects could be made visible to show which actions had a negative and which actions had a positive outcome in each situation. However, this approach can only provide recommendations for a single incident and does not take consecutive incidents and their consequences into account.

In this paper, we investigate the use of prescriptive process monitoring, inspired by [12], particularly reinforcement learning techniques, for this healthcare process, in which the optimal policies of the best possible action in a given situation (or state) can be determined. First, we train a *Markov Decision Process* (MDP) from the aggression incident log [10]. Second, we apply reinforcement learning techniques, aiming to find optimal policies for staff members to minimize aggressive incidents by clients with intellectual impairments. We use the model-free, value-based control algorithms: Q-learning and SARSA. The reason for choosing these methods, rather than the Monte Carlo methods used in [12], stems from their practical advantage of achieving earlier convergence on stochastic processes [13].

The structure of the paper is as follows. Section 2 discusses the related work. Then we explain the methods in Sect. 3, including the description of the data set and the design of the environment. Section 4 presents the results, and Sect. 5 discusses the results. Section 6 concludes the paper.

2 Related Work

Research in prescriptive process monitoring has been done in the recent couple of years, mainly with a focus on business processes. Fahrenkrog-Petersen et al. [1] used it to make a framework that parameterized a cost model to assess the cost-benefit tradeoff of generating alarms. Bozorgi et al. [14] researched it in the context of reducing the cycle time of general supply chain processes. Both use supervised learning methods instead of reinforcement learning methods and predict a threshold value that, when exceeded, recommends an action. The algorithms themselves do not make a recommendation; only predictions are made, and based on the predictions, a user-defined action is recommended.

Weinzierl et al. [15] also made this remark and proposed an alternative approach to prescriptive process monitoring in which there is a learning and a recommendation phase, in which the recommendation gives the next best action to take. Branchi et al. [12] used prescriptive process monitoring with Monte Carlo methods to determine the best actions to lend out loans and ensure most traffic fines are paid. The Monte Carlo methods are valid algorithms, although TD methods such as Q-learning and SARSA tend to converge earlier on stochastic processes in practice [13]. In this paper, we use Q-learning and SARSA to find optimal policies.

3 Methodology

This section describes the methods used in the research. First, we describe the data set. We then explain the preprocessing steps and the way the environment is built. Finally, we discuss the evaluation measures used.

3.1 Data Set

The data set is from a Dutch residential care organization with several facilities. The event data contains 21,384 reported aggression incidents from 1,115 clients with intellectual impairments. The data has been anonymized for privacy reasons. The reported incidents were reported by staff members between the 1st of January 2015 and the 31st of December 2017. The event data includes attributes such as the date of the incident, pseudonym client ID, the type of aggression, the countermeasure that the staff took, and the type of persons involved (such as family, staff members, and other clients). A simplified example of the event data is listed in Table 1.

In the event data, four types of aggression are reported, which are *verbal aggression* (va), *physical aggression against people* (pp), *physical aggression against objects* (po), and *self-injurious behavior* (sib). Eight distinct countermeasures are reported by the staff members: *talk to the client, held with force, no measure taken, seclusion, send to another room, distract client, terminate contact, and starting preventive measures*.

3.2 Data Cleaning and Preprocessing

To use reinforcement learning with this dataset, we preprocess the data. We follow the same steps as in [10]. First, we add the type of next aggression incident as an attribute

Table 1. A snippet of the incident data where the last column describes the countermeasures taken by staff members to stop the aggression.

Pseudonym client	Date of incident	Aggression type	Involved	Measures
ab45	05/01/2016	va	family	talk to client
ab45	06/01/2016	pp	client	none
lz12	06/01/2015	sib	unknown	seclusion
lz12	18/01/2015	po	client	none

of the current event, in order to create tuples of three which contain the type of current aggression, the countermeasures taken by a staff member, and the type of next aggression. The aim is to use the aggression types as the *states* a client is in and use the countermeasures as *actions*. Such a triplet describes a transition from one state to the next state after taking an action.

In the second step, we group incidents into *episodes*. According to a behavioral expert at the care organization [10], an *episode* is a sequence of incidents by the same client that occurred after each other, where the time between incidents is less than or equal to nine days. Following this domain knowledge, we segment the sequences of incidents into episodes. When two consecutive incidents e_i and e_{i+1} of a client are more than nine days apart, we insert a *Tau* after e_i as the final state of an episode. The incident e_{i+1} is the start state of the next episode. An overview of the approach is shown in Fig. 1.

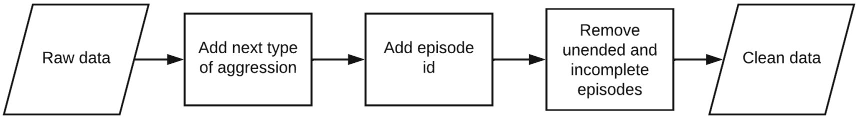

Raw data
Add next type of aggression
Add episode id
Remove unended and incomplete episodes
Clean data

Fig. 1. Preprocessing pipeline used to get enriched and clean data

We assign each episode a unique ID. The episodes that do not end in a *Tau* state are considered incomplete and, therefore, filtered. We obtained a total of 8,800 episodes after this filter, consisting of 19,848 incidents. In addition, the episodes where the incidents miss the values on the measures column are removed; these are incidents in which the staff member did not report the measures they had taken. Applying this filter reduced the number of episodes to 8,013, consisting of 15,464 incidents. Finally, we decided to remove the most infrequent action, ‘preventive measures started’ due to its ambiguity and to reduce the search space. Any episode that contains this action was removed, resulting in 14,676 incidents and 7,812 episodes for training the final MDP. In Table 2, a simplified example of the preprocessed log is listed.

Table 2. A simplified example of the preprocessed event data

Pseudonym client	Aggression type	Measures	Next aggression type	Episode Id
ab45	va	talk to client	pp	1
ab45	pp	none	Tau	1
lz12	sib	secluded	Tau	2
lz12	po	none	Tau	3

3.3 Building the Environment

Now that the data is cleaned and preprocessed, we use it to build a finite MDP. For this, we need the five-tuple consisting of the states, actions, transition probabilities, rewards, and discount factor [13]. The discount factor is a hyperparameter that can be tuned; therefore, we later perform hyperparameter tuning to determine the discount factor for the agent.

We describe the MDP using the standard formalization in [13] as follows:

- $\mathcal{S}$ = {va, po, sib, pp, Tau}, i.e., the set of states;
- $\mathcal{A}$ = {talk to the client, no measure taken, seclusion, holding with force, send to another room, distract client, terminate contact}, i.e., the set of actions;
- $\mathcal{P}$, which is the probability of going from one state to the next based on the action. This is determined using the following function

$$P(s, a, s') = \frac{\text{Number of times } a \text{ leads to } s'}{\text{Number of times } a \text{ is chosen in state } s} \tag{1}$$

- $\mathcal{R} : \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{Z}$, which is the reward function. We defined the reward function based on the literature in assessing the severity of the action and the state [16]. The reward (penalty) for each individual action or state is listed in Table 3.

Table 3. The rewards (penalty) assigned for each action or state, based on the severity of the action and the state [16]. When an agent takes an action and ends in the follow-up state, the combination of the action and state is used to compute the reward.

Action or state	Reward
Tau	1
Verbal Aggression (va)	0
Physical Aggression against objects (po)	−1
Self-injurious behavior (sib)	−3
Physical Aggression against people (pp)	−4
Client distracted, Contact terminated, Send to other room	−1
Hold with force, Seclusion	−2
Other actions	0

Another design choice has been made regarding the calculation of the transition probabilities. In the data set, multiple actions could be filled in at each incident. For this paper, a decision was made to consider only the most frequent action as the transition from one state to the next, in order to limit the number of possible actions and avoid having too many infrequent actions. Also, the reward function was designed based on the severity of the action and the state as indicated in the existing literature in aggression [16]. The simple reward function was designed on purpose such that the results can be more easily communicated to the experts. A subgraph of the environment can be seen in Fig. 2.

3.4 Training the Agents

We used the following parameters in the tuning: the learning rate, $\alpha \in [0, 1]$, the discount factor, $\gamma \in [0, 1]$, and the amount of exploration, $\epsilon \in [0, 1]$, which have an impact on the training of the agents and therefore the results. The best parameters are obtained experimentally by hyperparameter tuning using the best average reward of 100 runs as the goal, each consisting of 2000 episodes. The search spaces are $\alpha \in [0.1, 0.2, 0.3, 0.4, 0.5]$, $\gamma \in [0.2, 0.4, 0.6, 0.8, 1.0]$ and $\epsilon \in [0.1, 0.2, 0.3, 0.4, 0.5]$. First γ was obtained while keeping α and ϵ on 0.1. After this ϵ was obtained using the optimal γ value and $\alpha = 0.1$ and finally α was obtained using the optimal γ and ϵ value. Each parameter has been used for ten different runs to get a fair average. The hyperparameter values for both Q-learning and SARSA are $\alpha = 0.2$, $\gamma = 0.2$ and $\epsilon = 0.1$.

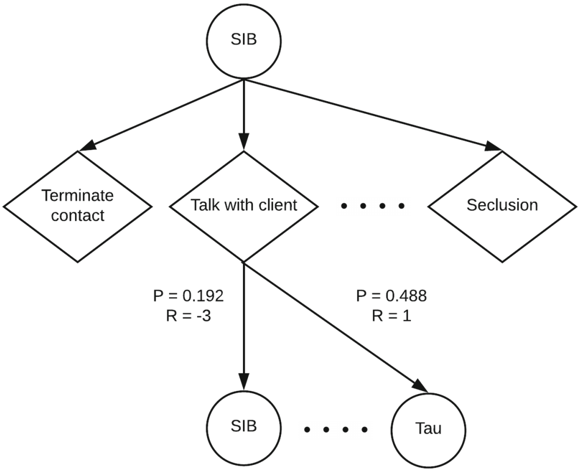


Fig. 2. A subgraph of the MDP, depicting the current state of self-injurious behavior (SIB), a sample of actions that can be chosen, and a sample of transitions. P is the probability of going to that state, and R is the reward associated with that action and next state.

3.5 Evaluation of Policies

We evaluate the agents both quantitatively, by comparing the average rewards, and qualitatively, by discussing the policies. For the quantitative evaluation, we compute the average reward for the best-trained agent using Q-learning and the best-trained agent using SARSA. These are then compared with the average reward of taking random actions and the average reward with the current policy. The current policy has been derived as the most frequent action taken in a state. The current policy is "talking to the client" when they display verbal aggression (va), physical aggression against people (pp), and physical aggression against objects (po). For the state of self-injurious behavior (sib) “no action” is the most frequently used action. For the qualitative evaluation, we discuss the results by looking at the most frequent variants for each agent and comparing these variants with the ones of the current policy.

4 Results

In this section, we first present the results regarding the rewards. Next, we discuss the results qualitatively, presenting the optimal policy and the variants. We used two baselines to compare the results: (1) using random actions and (2) taking the most frequent action at each state as the policy for the agent. The data set is shared under the NDA and thus unavailable. The code and the MDPs used in this paper are online available[1], which can be used to reproduce the results.

4.1 Quantitative Results

In this section, the average reward per policy is described and evaluated. It is listed in Table 4.

Table 4. Average reward per policy based on 10,000 runs, each with 100 episodes.

Policy	Average reward
Random	-3.783
Most frequent action	-1.105
Q-learning	-1.127
SARSA	-1.168

We run each policy for 10,000 runs, each consisting of 100 episodes, resulting in 1,000,000 episodes total. To test if the differences between the policies are significant, we performed a one-way ANOVA with the data from SARSA, Q-Learning, and the current policy. The one-way ANOVA was done using the Scipy library from Python 3, specifically the stats.f_oneway function. The p-value was $7.719e - 26$, which is smaller than 0.05, therefore, we can reject the null hypothesis that the groups have the same mean, meaning there is a significant difference between them, but we do not know between which. Therefore, we use a least significant difference test as a posthoc test using the pairwise_tukeyhsd from the Python 3 library statsmodels. This made three comparisons, it tested Q-learning against the current policy, tested SARSA against the current policy, and tested Q-learning against SARSA. It rejected all three null hypotheses, meaning the average reward per algorithm significantly differs from one another.

---

[1] https://git.science.uu.nl/6601723/ppm-aggressive-incidents.

4.2 Qualitative Results

This section describes the qualitative results where we show the derived policies and the most common variants of episodes per agent. The derived policies can be found in Table 5, where the action taken at each state for each policy can be found.

Table 5. Derived policies for Q-learning and SARSA together with the most frequent actions taken on the 10,000 runs, each with 100 episodes.

Policy	VA	SIB	PP	PO
Most frequent action	talk with client	no measure	talk with client	talk with client
Q-learning	talk with client	no measure	talk with client	no measure
SARSA	no measure	no measure	talk with client	talk with client

The five most common variants with their frequencies for each of the agents can be found in Tables 6, 7, and 8. In the tables, each variant is a distinct episode of tuples, where the first element of the tuple is the *current state*, the second element is the *action* taken, and the last element is the *next state* after the action. If the state is *Tau*, the episode is ended; otherwise, another action is taken.

Table 6. Five most common

Path	Frequency
(va, Talk with client, Tau)	14454
(sib, No measure, ‘Tau’)	13987
(po, Talk with client, Tau)	13100
(pp, Talk with client, Tau)	12769
(pp, Talk with client, pp) (pp, Talk with client, Tau)	4454

Table 7. Five most common variants when using the policy derived by Q-learning.

Path	Frequency
(va, Talk with client, Tau)	14417
(sib, No measure, Tau)	14294
(po, No measure, Tau)	12974
(pp, Talk with client, Tau)	12866
(pp, Talk with client, pp) (pp, Talk with client, Tau)	4526

Table 8. Five most common variants when using the policy derived by SARSA.

Path	Frequency
(va, No measure, Tau)	14926
(sib, No measure, Tau)	14025
(pp, Talk with client, Tau)	13079
(po, Talk with client, Tau)	12971
(sib, No measure, sib) (sib, No measure, Tau)	4313

In the tables, it can be seen that the four most frequent variants of episodes end in one action for all policies. For each state doing that action leads immediately to Tau regardless of the policy. Also, most of the episodes ended when only one action had been taken. When we take a closer look at the current policy, the Q-learning policy, and the SARSA policy, we see that most variants are the same with only two differences: (1) in the verbal aggression state (va), “no measure” action is suggested by the Q-learning; (2) in the self-injury-behavior state (sib), “no measure” action is suggested by SARSA.

4.3 Additional Analysis

Due to the results from taking all episodes, we decided to do an additional analysis on a subset of the data. We kept the episodes that had a length of more than or equal to three incidents and performed the same experiment as we did on the whole dataset. This subset contained 6687 incidents over 1360 episodes. Taking only the episodes longer than or equal to three incidents, we focus on the clients who display more severe behavior, which are the ones we want to help reduce in the first place. We again used Q-learning and SARSA as described above and compared them to taking random actions and the current policy, which in this case was “talk with client” in every state. The hyperparameter tuning was done the same as described in Sect. 3.4, resulting in the best performing Q-learning agent and best performing SARSA agent both with $\alpha = 0.1$, $\gamma = 0.2$ and $\epsilon = 0.1$. In the remaining parts of the additional analysis, we present the quantitative and qualitative results.

Additional Analysis Quantitative Results. In this section, the average reward per policy is shown and can be found in Table 9.

Table 9. Average reward per policy based on 10000 runs along 100 episodes

Policy	Average reward
Random	-11.925
Most frequent action	-7.342
Q-learning	-7.266
SARSA	-7.275

The same statistical tests as on the whole dataset were done. The p-value of the one-way ANOVA was $2.917e - 12$, which is smaller than 0.05, therefore, we can reject the null hypothesis. We use a least significant difference test as a posthoc test. This made three comparisons. It rejected two out of three null hypotheses. With a p-adj value of 0.6833, it did not reject the hypothesis that the rewards from Q-Learning and SARSA had the same mean. When taking a look at Table 10, we can see that they have learned exactly the same policy, so this result was expected.

Additional Analysis Qualitative Results. We list the derived policies with the current policy and compare the most common variants taken between a random baseline, the most frequent actions taken, and the two policies derived by the agents.

The derived policies can be found in Table 10, where the action taken at each state for each policy can be found. The Q-learning and SARSA agent learned that "talking to a client" was the best option when the state is verbal aggression (va), physical aggression against people (pp), or physical aggression against objects (po), and "no measure" when the state is self-injury behavior (sib).

Table 10. Derived policies for Q-learning and SARSA together with the most frequent actions taken on a subset of the data set only containing episodes $\geq$3

Policy	VA	SIB	PP	PO
Most frequent action	talk with client	talk with client	talk with client	talk with client
Q-learning	talk with client	no measure	talk with client	talk with client
SARSA	talk with client	no measure	talk with client	talk with client

The five most common variants with their frequencies for the current policy and the RL agents can be found in Tables 11 and 12. Both the Q-learning agent and SARSA agent learned the same policy. One noticeable difference between the frequent episodes of the policies of the RL agents and the most frequent policy is that the second most frequent episode of self-injurious behavior is added in Tables 11 and 12.

In the tables, it can be seen that the four most frequent variants end in one action for all policies. For each state, performing that action leads immediately to Tau regardless of the policy. This time only a fifth of all episodes lead immediately to Tau, for all policies except the random one. The random agent only has episodes ending with only one incident in its top ten, but this can be explained by their frequency, which is relatively much lower than the other policies (976 vs. 5052). When we take a closer look at the current policy and the RL agents' learned policy, the variants differ significantly in the frequencies of the self-injurious behavior (sib) state. The frequencies of the single-incident episodes for this state are similar between the RL agents and the current policy (4901 vs. 4772). When the episodes consist of two incidents that concern the self-injurious behavior (sib) state, the frequency of such episodes is much higher in the RL agents learned policy than the current policy (2720 vs. 1776), meaning that "no measure" results faster to Tau than "Talk with client" in this case.

Table 11. Five most common variants of episodes when using the most frequent actions for 1,000,000 episodes, plus the most common variant when considering SIB.

Path	Frequency
(va, Talk with client, Tau)	5052
(sib, Talk with client, Tau)	4772
(pp, Talk with client, Tau)	4520
(po, Talk with client, Tau)	4488
(pp, Talk with client, pp) (pp, Talk with client, Tau)	2617
(sib, Talk with client, sib) (sib, Talk with client, Tau)	1776

Table 12. Five most common variants by Q-learning for 1,000,000 episodes, plus the most common variant when considering SIB.

Path	Frequency
(va, Talk with client, Tau)	5207
(sib, No measure, Tau)	4901
(po, Talk with client, Tau)	4661
(pp, Talk with client, Tau)	4620
(pp, Talk with client, pp) (pp, Talk with client, Tau)	2749
(sib, No measure, sib) (sib, No measure, Tau)	2720

5 Discussion

The results indicate that the current policy and the RL-derived policies reach similar conclusions. The current policy performs slightly better than the RL agents when considering all episodes, but the RL agents provide staff members with additional options without having a significant negative impact on rewards. When considering the selected subset of the episodes, the RL agents slightly outperform the current policy, offering an alternative choice.

In both cases, the staff member can choose to talk to the client or take no action. Although the RL algorithms exhibit slight variations in performance compared to the current policy, the policies derived do not significantly differ. This alignment is reasonable considering the reward function used, which penalizes all actions except “no measure” and “talk with client”. These options align with the least disruptive impact on both the client and staff member, as indicated by previous studies.

However, it is important to note that the models may oversimplify the real situation, and further factors such as location, time, and individuals involved have not been included. Collecting relevant data and consulting behavioral experts could enhance future research in this field. For example, it is possible to learn the time distribution until the next incident and use this in the reward function.

Additionally, practical relevance should be acknowledged, as staff members face challenges in assessing situations and may need to use force in certain cases. Future research may aim to provide insights tailored to specific clients or client groups. Combining reinforcement learning and process mining in prescriptive process monitoring shows promise but requires careful consideration of data availability and exploration limitations.

6 Conclusion

This paper presents the application of reinforcement learning (RL) to optimize response policies in healthcare processes, specifically addressing aggressive incidents in care settings. The research aims to investigate the validity of RL in healthcare and the ability to find optimal response policies for staff members towards such incidents. The results have shown that RL algorithms can find such an optimal policy, which consists of taking no measures or talking with the client depending on the state. The policies are very similar to the current policy, i.e., the most frequent action taken by staff members.

Despite the simple MDP, the results do show that prescriptive process monitoring can be used in the healthcare domain. Interestingly, it may be more beneficial to use the techniques in more complex situations, rather than the simple situation. However, further research is necessary to validate this finding.

For future work, one may refine the environment by extending the MDP with more refined states and actions. Future research should be multidisciplinary, where such an environment can be more elaborately built based on experts in the field of aggressive behavior and staff members who work daily with clients. Results can then also be validated by the experts or staff to help them make better decisions and therefore their input is crucial.

Acknowledgement. This research was supported by the NWO TACTICS project (628.011.004) and Lunet in the Netherlands. We would like to thank the experts from the Lunet for their assistance. We also thank Dr. Shihan Wang and Dr. Ronald Poppe for the invaluable discussions.

References

1. Fahrenkrog-Petersen, S.A., et al.: Fire now, fire later: alarm-based systems for prescriptive process monitoring. Knowl. Inf. Syst. **64**(2), 559–587 (2021)
2. Metzger, A., Kley, T., Palm, A.: Triggering proactive business process adaptations via online reinforcement learning. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 273–290. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_16
3. Bozorgi, Z.D., Dumas, M., Rosa, M.L., Polyvyanyy, A., Shoush, M., Teinemaa, I.: Learning when to treat business processes: prescriptive process monitoring with causal inference and reinforcement learning. In: Indulska, M., Reinhartz-Berger, I., Cetina, C., Pastor, O. (eds.) CAiSE 2023. LNCS, vol. 13901, pp. 364–380. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-34560-9_22
4. Munoz-Gama, J., et al.: Process mining for healthcare: characteristics and challenges. J. Biomed. Inform. **127**, 103994 (2022)
5. Hensel, J.M., Lunsky, Y., Dewa, C.S.: Staff perception of aggressive behaviour in community services for adults with intellectual disabilities. Community Ment. Health J. **50**(6), 743–751 (2013)

6. Mills, S., Rose, J.: The relationship between challenging behaviour, burnout and cognitive variables in staff working with people who have intellectual disabilities. J. Intellect. Disabil. Res. **55**(9), 844–857 (2011)
7. Bushman, B.J., Bonacci, A.M., Pedersen, W.C., Vasquez, E.A., Miller, N.: Chewing on it can chew you up: effects of rumination on triggered displaced aggression. J. Pers. Soc. Psychol. **88**(6), 969–983 (2005)
8. Koorn, J., Lu, X., Leopold, H., Reijers, H.A.: Towards understanding aggressive behavior in residential care facilities using process mining. In: Guizzardi, G., Gailly, F., Suzana Pitangueira Maciel, R. (eds.) ER 2019. LNCS, vol. 11787, pp. 135–145. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34146-6_12
9. Koorn, J.J., Lu, X., Leopold, H., Reijers, H.A.: Looking for meaning: discovering action-response-effect patterns in business processes. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 167–183. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_10
10. Koorn, J.J., Lu, X., Leopold, H., Reijers, H.A.: From action to response to effect: mining statistical relations in work processes. Inf. Syst. **109**, 102035 (2022)
11. Koorn, J.J., Lu, X., Leopold, H., Martin, N., Verboven, S., Reijers, H.A.: Mining statistical relations for better decision making in healthcare processes. In: ICPM, pp. 32–39. IEEE (2022)
12. Branchi, S., Di Francescomarino, C., Ghidini, C., Massimo, D., Ricci, F., Ronzani, M.: Learning to act: a reinforcement learning approach to recommend the best next activities. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds.) BPM 2022. LNBIP, vol. 458, pp. 137–154. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16171-1_9
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. A Bradford Book, Cambridge (2018)
14. Bozorgi, Z.D., Teinemaa, I., Dumas, M., Rosa, M.L., Polyvyanyy, A.: Prescriptive process monitoring for cost-aware cycle time reduction. In: 2021 3rd International Conference on Process Mining (ICPM), pp. 96–103 (2021)
15. Weinzierl, S., Zilker, S., Stierle, M., Matzner, M., Park, G.: From predictive to prescriptive process monitoring: recommending the next best actions instead of calculating the next most likely events. In: Wirtschaftsinformatik (Zentrale Tracks), pp. 364–368 (2020)
16. Nijman, H.L., et al.: The staff observation aggression scale-revised (SOAS-R). Aggressive Behav. **25**(3), 197–209 (1999)

ProtoNER: Few Shot Incremental Learning for Named Entity Recognition Using Prototypical Networks

Ritesh Kumar(✉), Saurabh Goyal, Ashish Verma, and Vatche Isahagian

IBM Research, Cambridge, USA
{kumar.ritesh,saurabh.goyal1,vatchei}@ibm.com, draverma@amazon.com

Abstract. Key value pair (KVP) extraction or Named Entity Recognition (NER) from visually rich documents has been an active area of research in document understanding and data extraction domain. Several transformer based models such as LayoutLMv2 [1], LayoutLMv3 [2], and LiLT [3] have emerged achieving state of the art results. However, addition of even a single new class to the existing model requires (a) re-annotation of entire training dataset to include this new class and (b) retraining the model again. Both of these issues really slow down the deployment of updated model.

We present **ProtoNER**: Prototypical Network based end-to-end KVP extraction model that allows addition of new classes to an existing model while requiring minimal number of newly annotated training samples. The key contributions of our model are: (1) No dependency on dataset used for initial training of the model, which alleviates the need to retain original training dataset for longer duration as well as data re-annotation which is very time consuming task, (2) No intermediate synthetic data generation which tends to add noise and results in model's performance degradation, and (3) Hybrid loss function which allows model to retain knowledge about older classes as well as learn about newly added classes.

Experimental results show that ProtoNER finetuned with just 30 samples is able to achieve similar results for the newly added classes as that of regular model finetuned with 2600 samples.

Keywords: Business Document Information Extraction · Few Shot Class Incremental Learning (FSCIL) · Named Entity Recognition (NER) · Key Value Pair Extraction (KVP) · Token Classification · Prototypical Networks

1 Introduction

Business processes provide a structured framework for enterprises to do work. They define tasks, their executors, and capture dependencies as well as provide

---

Work done while author was working at IBM Research.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 70–82, 2024.
https://doi.org/10.1007/978-3-031-50974-2_6

logging and tracking capabilities [27]. They also align with company policies and compliance with governmental regulations. Business process tasks are typically associated with unstructured data in the form of documents, which contain information deemed critical to the successful execution of the business process. For example, a loan application will be associated with multiple documents containing name, salary, credit score etc. of an individual. In the age of digital transformation, where enterprises are focusing on augmenting business processes with Artificial Intelligence [25,26,28], automating the extraction of knowledge from these rich documents such as loan applications, invoices, purchase orders, and utility bills, understanding business documents is critical because Incomplete, and Inaccurate information can lead to process execution delay and loss of revenue. Most recently, key-value pairs extraction has received significant attention because of its ability to influence the automation of several downstream tasks and affect the completion time of the business processes.

Traditional approaches such as template matching and region-segmentation based models [5–8] have been commonly used in industry for KVP extraction as they provide flexibility to train and deploy the models at much faster pace. Unfortunately, these models only work for the documents that they have observed during model training time and even a slight change in the layout of the document results in poor performance [9].

Deep learning based models such as LayoutLMv3 [2] and FormNet [4] achieve state of the art results and work very well even for the unseen documents. Such properties of deep learning models have compelled their industry wide rapid adoption. Unfortunately, these models are not able to predict a new set of key classes for which the model is not explicitly trained. With the ever-evolving nature of form like documents, it becomes crucial for such models to support addition of new key classes on top of existing ones in a fairly simplified and straight forward manner. With the goal of addressing this issue, in this paper, we adopt Prototypical Network [12] based model architecture to support the addition of new key classes to an already trained model. Prototypical networks have been widely studied for several computer vision related tasks but their inclusion in language related tasks remains limited. This is mainly due to the fact that several existing machine learning based algorithms can be used to extract features from the images pertaining to a class and be treated as prototype whereas there is not a clear or equivalent approach readily available that can be exploited to create prototypes in the language domain. We present a novel approach to create prototypes corresponding to different classes in the language domain that uses only a few samples to facilitate few shot class-incremental learning (FSIL), while avoiding the model's Catastrophic Forgetting [16] problem.

2 Problem Formulation

As described earlier, with rapid digitization of business workflows, enterprises are expected to update their models frequently with the capability of extracting more and more key value pairs from visually rich documents. The problem can be illustrated using Fig. 1. Consider model M_0 trained to extract 4 key classes (PO Number, PO Amount, Currency, and Customer Name) is currently being used by an enterprise to automate several downstream tasks (e.g. 3-way matching [24]). However, the enterprise needs to further extract 3 more classes (PO Date, Country, and Bill To Address) along with the previous 4 keys to further extend the functionality. Therefore, the resulting new model M_1 needs to support 7 key classes in total instead of just 4. Formally, we define the problem as follows:

Definition 1. *Given a set of* N *classes, a Model* M_k *is trained on* K *classes where* $K \subset N$*. Train a new model* M_{k+j} *trained on* $K + J$ *classes, where* $J \subset N$*,* $K \cap J = \emptyset$*, and* $|J| + |K| \leq |N|$*.*

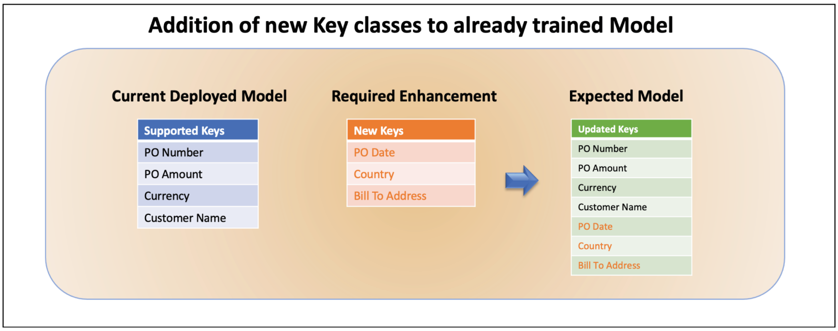


Fig. 1. Current deployed model supports extracting 4 classes. How to enhance the understanding of the model to support three additional classes.

In a conventional training setup, adding new classes requires re-annotation of the entire dataset to incorporate these new classes followed by retraining the model, which is a very slow and time consuming process because of its reliance on human annotators. This is further complicated by the fact that the original training dataset may not be available due to the data retention policies adopted by the organization. Our approach utilizes a handlful of newly annotated samples to facilitate few shot class-incremental learning (FSCIL) [20].

Note that adding new classes to an already trained model may introduce a Catastrophic Forgetting [16] problem where model tends to forget the knowledge acquired about older classes while learning about new classes. To alleviate this problem, we incorporate a hybrid loss function that combines cross entropy loss and cosine similarity loss. Cosine similarity loss is only applied to the older classes during addition of new class which forces the model to retain the knowledge about older classes whereas cross entropy loss allows the model to learn about new classes at the same time. It also helps the network not to overfit on the few-shot instances as well as not becoming biased to the base classes.

3 ProtoNER

3.1 Model Architecture

Transformer based NLP models such as LayoutLMv2 [1] and LayoutLMv3 [2] have shown to achieve state of the art results for KVP extraction on public datasets like CORD [13] and FUNSD [14] by leveraging text, layout, and image modalities. In this work, we leverage the LayoutLMv2 [1] model architecture as the basis for our modified prototypical network based architecture. Note that our architecture is generic enough to support other multi-modal architectures that are capable of performing KVP extraction such as DocFormer [10] or TiLT [11].

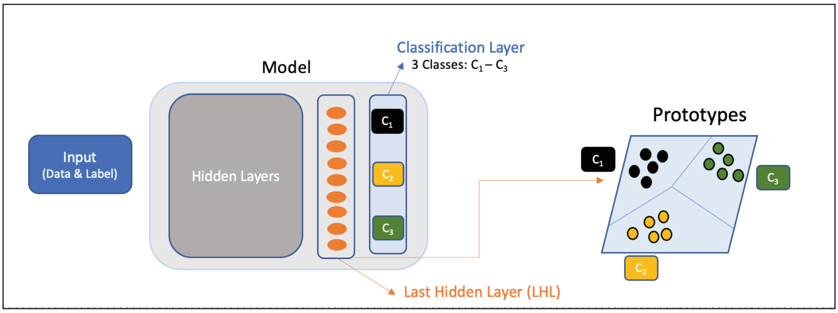


Fig. 2. Converting LayoutLMv2 to Prototypical Network Architecture

Figure 2 illustrates the overall architecture. For the sake of simplicity, all the transformer model blocks/layers are encapsulated as "Hidden Layers" block and only the last hidden layer (LHL) and classification layer are depicted. Though LayoutLMv2 model supports sequence length of up to 512 tokens and classifies each token, we have shown the classification head only for one token here and named it as "classification layer".

To convert the LayoutLMv2 model into prototypical network architecture, last hidden layer of the model is leveraged during training as well as inference. While training the model, if a token with ground truth label C_1 gets classified correctly by the classification layer, the last hidden layer representation for that token is saved as a prototype for key class C_1 under the prototype pool as shown in Fig. 2. In a similar fashion, prototypes corresponding to all of the classes are saved during training. Multiple prototypes per class are saved to capture better diversity within the class prototypes. The number of prototypes per class to be saved is a hyper parameter. The prototypes are saved only during the last epoch of the training to allow the model learn and achieve good accuracy across all key classes before saving the prototypes. The prototypes are only saved when the model classifies the token correctly i.e. the ground label for the token matches the predicted label. Each prototype is a vector of length equal to the length of the model's last hidden layer.

When the trained model is used for inferencing, the cosine similarity score is computed between the LHL representation of the token against all the saved prototypes from the pool and the label for the given token is derived by performing K-Nearest Neighbour search based on the computed cosine similarity scores. Doing the K-Nearest neighbour search from prototypes pool completely eliminates the need to have a classification layer.

3.2 Training Procedure

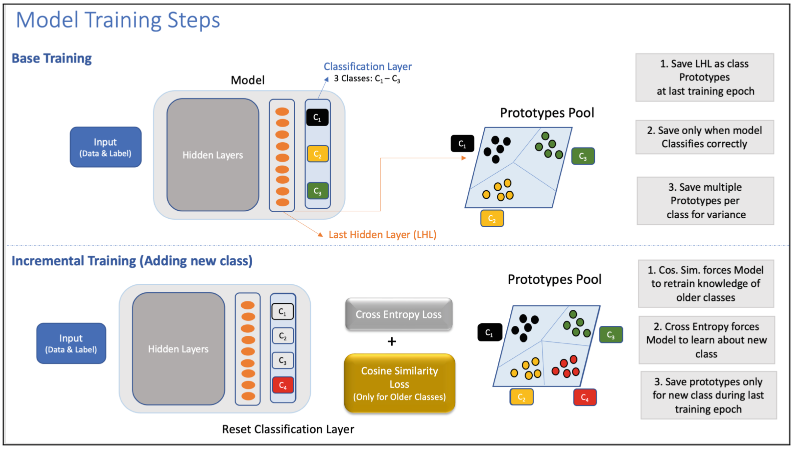

Model Training Steps
Base Training
Model
Classification Layer
3 Classes: $C_1 - C_3$
Input (Data & Label)
Hidden Layers
C_1
C_2
C_3
Prototypes Pool
C_1
C_3
C_2
Last Hidden Layer (LHL)
1. Save LHL as class Prototypes at last training epoch
2. Save only when model Classifies correctly
3. Save multiple Prototypes per class for variance
Incremental Training (Adding new class)
Input (Data & Label)
Hidden Layers
C_1
C_2
C_3
C_4
Cross Entropy Loss
+
Cosine Similarity Loss (Only for Older Classes)
Reset Classification Layer
Prototypes Pool
C_1
C_3
C_2
C_4
1. Cos. Sim. forces Model to retrain knowledge of older classes
2. Cross Entropy forces Model to learn about new class
3. Save prototypes only for new class during last training epoch

Fig. 3. Overall Model Training Process

Figure 3 highlights the overall training procedure for our model, which consists of two steps: Base Training, and Incremental Training. Base Training consists of training the model for the first N classes (N+1 if we include "Other" class) and incremental training consists of adding new class to the model achieved after base training.

During the base training, the model is trained for N classes as per the procedure described in the previous section. The trained model weights achieved after the base training is termed as base model. Multiple prototypes are saved for all of the N classes to create the prototypes pool.

During the incremental training phase to add new class to the base model, the prototypes saved for the older N classes are carried forward as is and not updated at any point of time during the incremental training. Only the prototypes pertaining to the "Other" class are discarded before initiating the incremental training. The classification layer of the base model is reset to reflect

N+1 classes. Using only the few newly annotated samples (annotated for n+1 key classes) and trained base model weights, the model is finetuned further to acquire knowledge about this newly added class. Our hybrid loss function is used only during this incremental training phase. For the tokens pertaining to the older key classes, the cosine similarity loss is computed between the LHL representation for the token and prototypes for the respective key class from the prototypes pool. Since multiple prototypes per key class are saved, we compute the cosine similarity loss between LHL representation of the token and each prototype for that key class to derive the average loss. This loss is simply added linearly to the cross entropy loss computed for the same token. Other possibility could have been to associate learnable parameters with both of the losses and let the model learn them during training. We did not explore this possibility in this work.

This way the model is forced to retain LHL representation for the older key classes as similar to the original state as possible. For the tokens pertaining to newly added key class, only the cross entropy loss is computed. Multiple prototypes are saved during the last epoch for the newly added class as well as the "Other" class. The same procedure can be repeated again in the future to add additional key classes. Note that more than one new class can be added concurrently during the same incremental training phase[1].

4 Experimental Results

4.1 Dataset

The dataset used for the analysis contains 2742 purchase orders obtained from various sources and consists of about 73 unique layout templates. 2600 samples are used for training and 142 for evaluation. Each document is annotated with a subset of the 10 pre-defined key classes. Table 1 lists these key classes along with their respective frequencies i.e. how many times these key classes appear in the dataset. The annotations contain 2-D coordinates and the key class label for the values corresponding to the pre-defined keys (not the words corresponding to keys themselves). Since any document such as purchase order or invoice generally contains additional text that does not pertain to any of the key classes, we also include an "Other" class along with the 10 pre-defined key classes to refer to those remaining words in the document. Note that all the annotations are at field level rather than at the word level as illustrated in Fig. 4.

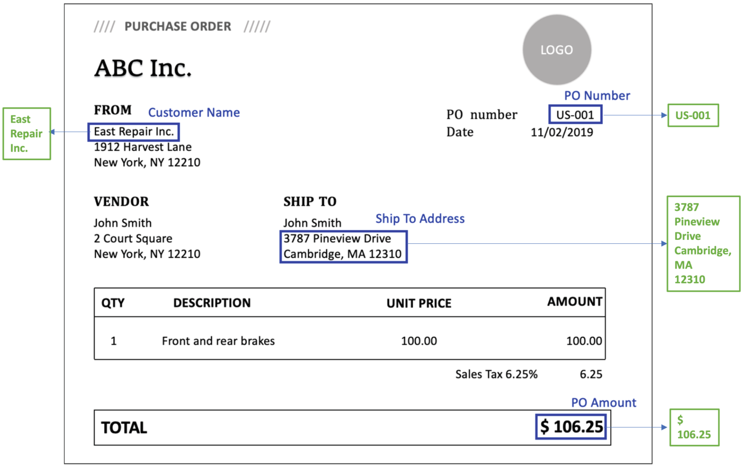
---

[1] The addition of multiple classes sequentially (one at a time) vs. all at same time results in similar accuracy.

Table 1. Frequency of keys present in Purchase Order dataset

#	Key Name	Frequency
1	PO NUMBER	2377
2	PO AMOUNT	1384
3	CUSTOMER NAME	1168
4	COUNTRY	1033
5	CURRENCY	1311
6	BILL-TO ADDRESS	1334
7	BILL-TO CUSTOMER NAME	1030
8	SHIP-TO ADDRESS	1390
9	SHIP-TO CUSTOMER NAME	1050
10	LOGO CUSTOMER NAME	1631

In order to process the data, each document is first passed through an Optical Character Recognition (OCR) engine to extract the words and their respective bounding box coordinates. Since annotations are done at field level and OCR extracts the text and corresponding bounding boxes at word level, we split the annotations at word level to align it with the OCR output. Both OCR output and pre-processed annotations are required for training the model for KVP extraction.



//// PURCHASE ORDER /////

LOGO

ABC Inc.

FROM
East Repair Inc.
1912 Harvest Lane
New York, NY 12210

PO number US-001
Date 11/02/2019

VENDOR
John Smith
2 Court Square
New York, NY 12210

SHIP TO
John Smith
3787 Pineview Drive
Cambridge, MA 12310

QTY	DESCRIPTION	UNIT PRICE	AMOUNT
1	Front and rear brakes	100.00	100.00

Sales Tax 6.25% 6.25

TOTAL $ 106.25

Customer Name
PO Number
Ship To Address
PO Amount
East Repair Inc.
US-001
3787 Pineview Drive Cambridge, MA 12310
$ 106.25

Fig. 4. Purchase order sample showing annotated field level key labels (i.e. name, and location) of Customer Name, PO Number, Ship To Address, and PO Amount along with their values. The list of text in the green boxes represents the word level OCR output obtained for each of the field level annotations. (Color figure online)

4.2 Evaluation

In order to evaluate our ProtoNER model, we follow the 2-step procedure described in the earlier section to train our model. In the first step, we train the pre-trained LayoutLMv2 [1] model on 2600 training samples annotated with only 4-classes as shown in Table 2. In order to do that, we modify the original training data annotations and replace all the key classes except the original 4 key classes with the "Other" class. The model is trained for 100 epochs with 2e-5 learning rate and 8 batch size. At the end of this first step of training, we save 50 prototypes for each of the 4 key classes along with the model weights. The decision to save 50 prototypes per class was taken based on the empirical analysis.

In second step, we further fine-tune the model trained in the previous step with just 30 samples annotated with all the 10 key classes. The training parameters used for this step are: train epochs = 100, learning rate = 5e-6, and batch size = 8.

This 2-step training regime mimics the practical industrial scenario where the initial model is usually trained on large dataset with small number of key classes and the model needs to be updated to cater for future requirements i.e. identify new key classes but with limited training data.

Table 2. ProtoNER training samples, keys, and model results are shown for base training (under subtable (a)) and incremental training (under subtable (b)). Common key classes across both models are highlighted in blue color.

Base Model Training	
Train Samples	2600
Keys	4
Prototypes per class	50
Test samples	142

Key Class	**Prec**	**Rec**	**F1**
PO Number	0.87	0.79	0.83
Logo Cust Name	0.87	0.79	0.83
Ship To Addr	0.83	0.85	0.84
Ship To Cust Name	0.76	0.87	0.81

(a) **Base model accuracy**

Incremental Model Training	
Train Samples	30
Keys	6
Prototypes per class	50
Test samples	142

Key Class	**Prec**	**Rec**	**F1**
PO Number	0.87	0.89	0.88
Logo Cust Name	0.90	0.76	0.81
Ship To Addr	0.87	0.78	0.82
Ship To Cust Name	0.76	0.88	0.81
Bill To Addr	0.63	0.85	0.72
Bill To Cust Name	0.74	0.79	0.76
Country	0.68	0.86	0.76
Currency	0.68	0.90	0.78
Customer Name	0.67	0.73	0.69
PO Amount	0.76	0.77	0.76

(b) **Incremental model accuracy**

Table 2 illustrates the Precision, Recall, and F1 scores achieved for key classes by **ProtoNER**. Table 3a shows the scores for the base model trained with 4 key classes and Table 3b shows the scores for the base model finetuned further to support 10 key classes. From the Table 2, it can be observed that the model is able to perform well for the newly added keys even after training with only 30 newly annotated samples. The hybrid loss function is able to force the model to retain the knowledge about older classes as well as gain knowledge about new classes. Also, the scores improve for the older keys after addition of new keys. The reason behind this improvement is that the false positives and false negatives for the 4 key classes get spread over 10 key classes now instead of 4.

The rationale behind how model is able to learn about new key classes from only few samples can be attributed to the sub-clustering being performed by the model inherently during the base training itself. Even though the words pertaining to left out 6 key classes are labeled as "Other" during the base training, the model inherently forms sub-clusters under the parent "Other" class umbrella for these 6 key classes. Exposing the model with few samples containing new key classes during incremental stage allows the mapping of such already formed sub-clusters to these new key classes. Meihan et al. [23] have reported similar observations under their few shot work.

Table 3. Training configuration for ProtoNER, LayoutLMv2-10C, and LayoutLMv2-4C-10C models. Both ProtoNER and LayoutLMv2-4C-10C are first trained for 4 key classes using 2600 samples followed by incremental training for additional 6 classes using 30 samples.

Training Attributes					
ProtoNER		LayoutLMv2-10C		LayoutLMv2-4C-10C	
Base train samples	2600	Train samples	2600	Train samples	2600
Base key classes	4	Base key classes	10	Base key classes	4
Test samples	142	Test samples	142	Test samples	142
Incremental samples	30			Incremental samples	30
Incremental key classes	6			Incremental key classes	6

Table 4. Results comparison between ProtoNER, LayoutLMv2-10C, and LayoutLMv2-4C-10C models obtained for 142 test samples. LayoutLMv2-10C model is trained for all 10 key classes using 2600 samples. LayoutLMv2-4C-10C and ProtoNER models are first trained for 4 key classes using 2600 samples followed by incremental addition of 6 key classes using only 30 samples. F1 scores for incremental key classes for both ProtoNER and LayoutLMv2-4C-10C models are highlighted in red color.

Results Comparison									
Key Classes	ProtoNER			LayoutLMv2-10C			LayoutLMv2-4C-10C		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
PO Number	0.87	0.89	0.88	0.88	0.82	0.84	0.77	0.55	0.64
Logo Cust Name	0.90	0.76	0.81	0.88	0.82	0.84	0.77	0.46	0.57
Ship To Addr	0.87	0.78	0.82	0.84	0.81	0.82	0.45	0.42	0.43
Ship To Cust Name	0.76	0.88	0.81	0.77	0.89	0.82	0.42	0.39	0.40
Bill To Addr	0.63	0.85	0.72	0.81	0.83	0.81	0.39	0.50	0.44
Bill To Cust Name	0.74	0.79	0.76	0.80	0.84	0.81	0.43	0.49	0.46
Country	0.68	0.86	0.76	0.82	0.85	0.83	0.62	0.64	0.62
Currency	0.68	0.90	0.78	0.80	0.93	0.86	0.72	0.78	0.75
Customer Name	0.67	0.73	0.69	0.73	0.79	0.75	0.50	0.69	0.58
PO Amount	0.76	0.77	0.76	0.86	0.88	0.86	0.64	0.62	0.62

4.3 Comparison Against LayoutLMv2 Model

We also trained 2 baseline LayoutLMv2 models to compare against our model. We used the original implementation source code provided by the authors of LayoutLMv2 here: https://github.com/microsoft/unilm/tree/master/layoutlmv2. The first baseline model **LayoutLMv2-10C** was trained by fine-tuning the pre-trained LayoutLMv2 model on 2600 training samples annotated with all the 10 key classes. The model is trained for 100 epochs with 2e-5 learning rate and 8 batch size.

The second baseline model **LayoutLMv2-4C-10C** was trained in 2 steps. It was first trained for 4 key classes using 2600 samples followed by finetuning further for all 10 key classes using only 30 samples. It was trained for 100 epochs with 2e-5 learning rate and 8 batch size followed by finetuning for 10 key classes for 100 epochs, 5e-6 learning rate and 8 batch size. All the models were trained on single V100 GPU.

Table 3 lists the overall training configuration for all of the 3 models. Table 4 compares the precision, recall and F1-score for all the 10 key classes for our model(ProtoNER) with LayoutLMv2-10C baseline model. It can be observed that for the original 4 key classes (PO Number, ShipToAddr, ShipToCustName, and LogoCustName) the F1-score of our model is comparable to the LayoutLMv2-10C model with 3% drop only for 1 specific key class(LogoCustName). For the remaining 6 key classes, our model is able to learn only with the help of 30 new samples. The results suggest that the hybrid loss function is able to force the model to retain the knowledge about the original 4 key classes during the incremental training phase as well as achieves about 90% of the LayoutLMv2-10C model's accuracy for the newly added key classes with just 30 samples.

Table 4 also shows the comparison of precision, recall and F1-score between our model(ProtoNER) and LayoutLMv2-4C-10C model. It can be observed that our model performs significantly better than the LayoutLMv2-4C-10C model on original 4-classes with gains in F1-score as high as 40% for some of the key classes and 32% on average. This is due to the fact that our model is able to retain the knowledge about the original 4-classes in the form of saved prototypes and hybrid loss function while LayoutLMv2-4C-10C model suffers from catastrophic forgetting problem. For the newly added 6 classes, layoutLMv2-4C-10C model undergoes severe over-fitting due to the small dataset size while our model is able to generalize better due to hybrid loss function.

5 Related Work

The adoption of language models (LM) really demands the flexibility of continual and incremental learning. In context of incremental learning for KVP/NER, Chen and Moschitti [19] present an approach for transferring knowledge from one model trained on specific dataset to a new model trained on another dataset containing new keys/classes. Their overall model architecture tries to learn the differences between the source and target label distribution with the help of neural adapter. Greenberg et al. [22] use marginal likelihood training to strengthen the knowledge acquired by their model from different available datasets while filling in missing labels for each dataset to align them. Both of these models require availability of more than one annotated datasets and also does not incorporate the few shot training aspect. Huang et al. [15] present a comprehensive study on the few shot training for NER task and mention about noisy supervised approach, knowledge distillation based teacher student model and prototypical networks based model.

On one hand, incremental learning aspect is being explored to allow addition of new classes to already trained model, on the other hand, different Few Shots based techniques are being developed to train model with minimal number of annotated examples using transfer learning. However, very limited work has been carried which exploits both Few Shot and Incremental Learning at the same time specifically for NER task.

Monaikul et al. [18] present model for incremental learning for NER task which follows teacher student architecture. Their approach passes the data through trained base model and considers the predicted labels as the ground truth label during the incremental training phase. It adds impurities to the training data since all the inaccurate predictions by the base model get passed to the incremental training. The complex inference head also requires sophisticated rules to eventually derive the final prediction which may induce/lead to inaccuracies. Zhou et al. [17] present meta learning based approach which relies on synthetically generated data. The model presented by Cheraghian et al. [21] few shot class incremental learning corresponds to vision domain where each sample contains data point pertaining to only one class and therefore, this model cannot be leveraged as is for the KVP task.

6 Conclusion

Our approach demonstrates how a prototypical network architecture inspired model setup with hybrid loss function can be used to incorporate real-life constraints and still achieve similar results as that of regular model. It provides a solution to add new keys on top of already trained model if and when required in the future with very limited data. It also eliminates the need to retain the original training dataset that could be a challenge in real-life scenario due to data retention policy adopted by different organizations. The overall setup provides flexibility to deploy such models in automated environment where end user can decides to add new keys with significantly less efforts.

References

1. Xu, Y., et al.: LayoutLMv2: multi-modal pre-training for visually-rich document understanding. In: ACL (2021)
2. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: pre-training for document AI with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia (2022)
3. Wang, J., Jin, L., Ding, K.: LiLT: a simple yet effective language-independent layout transformer for structured document understanding. arXiv preprint arXiv:2202.13669 (2022)
4. Lee, C.Y., et al.: FormNet: structural encoding beyond sequential modeling in form document information extraction. arXiv preprint arXiv:2203.08411 (2022)
5. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
6. Liu, X., Gao, F., Zhang, Q., Zhao, H.: Graph convolution for multimodal information extraction from visually rich documents. arXiv preprint arXiv:1903.11279 (2019)
7. Watanabe, T., Luo, Q., Sugie, N.: Layout recognition of multi-kinds of table-form documents. IEEE Trans. Pattern Anal. Mach. Intell. **17**(4), 432–445 (1995)
8. Seki, M., Fujio, M., Nagasaki, T., Shinjo, H., Marukawa, K.: Information management system using structure analysis of paper/electronic documents and its application. In: Proceedings of International Conference on Document Analysis and Recognition (ICDAR), pp. 689–693 (2007)
9. Hu, K., Wu, Z., Zhong, Z., Lin, W., Sun, L., Huo, Q.: A Question-Answering Approach to Key Value Pair Extraction from Form-like Document Images. arXiv preprint arXiv:2304.07957 (2023)
10. Appalaraju, S., Jasani, B., Kota, B.U., Xie, Y., Manmatha, R.: DocFormer: end-to-end transformer for document understanding. In: ICCV (2021)

11. Powalski, R., Borchmann, Ł, Jurkiewicz, D., Dwojak, T., Pietruszka, M., Pałka, G.: Going Full-TILT boogie on document understanding with text-image-layout transformer. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12822, pp. 732–747. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86331-9_47
12. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems (2017)
13. Park, S., et al.: CORD: a consolidated receipt dataset for post-OCR parsing. In: Workshop on Document Intelligence at NeurIPS 2019 (2019)
14. Jaume, G., Ekenel, H.K., Thiran, J.P.: FUNSD: a dataset for form understanding in noisy scanned documents. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW) (2019)
15. Huang, J., et al.: Few-shot named entity recognition: a comprehensive study. arXiv preprint arXiv:2012.14978 (2020)
16. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: the sequential learning problem. In: Psychology of Learning and Motivation (1989)
17. Zhou, D.W., Ye, H.J., Ma, L., Xie, D., Pu, S., Zhan, D.C.: Few-shot class-incremental learning by sampling multi-phase tasks. IEEE Trans. Pattern Anal. Mach. Intell. (2022)
18. Monaikul, N., Castellucci, G., Filice, S., Rokhlenko, O.: Continual learning for named entity recognition. In: AAAI (2021)
19. Chen, L., Moschitti, A.: Transfer learning for sequence labeling using source model and target data. In: AAAI (2019)
20. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12183–12192 (2020)
21. Cheraghian, A., Rahman, S., Fang, P., Roy, S.K., Petersson, L., Harandi, M.: Semantic-aware knowledge distillation for few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
22. Greenberg, N., Bansal, T., Verga, P., McCallum, A.: Marginal likelihood training of BiLSTM-CRF for biomedical named entity recognition from disjoint label sets. In: EMNLP 2018, pp. 2824–2829 (2018)
23. Tong, M., et al.: Learning from miscellaneous other-class words for few-shot named entity recognition. arXiv preprint arXiv:2106.15167 (2021)
24. BPI Challenge 2019 (2019). https://icpmconference.org/2019/icpm-2019/contests-challenges/bpi-challenge-2019/
25. Rizk, Y., et al.: A conversational digital assistant for intelligent process automation. In: Asatiani, A., et al. (eds.) BPM 2020. LNBIP, vol. 393, pp. 85–100. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58779-6_6
26. Rizk, Y., Venkateswaran, P., Isahagian, V., Muthusamy, V., Talamadupula, K.: Can you teach robotic process automation bots new tricks? In: Marrella, A., et al. (eds.) BPM 2022. LNBIP, vol. 459, pp. 246–259. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16168-1_16
27. Weske, M.: Business process management methodology. In: Weske, M. (ed.) Business Process Management: Concepts, Languages, Architectures, pp. 373–388. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28616-2_8
28. Huo, S., Völzer, H., Reddy, P., Agarwal, P., Isahagian, V., Muthusamy, V.: Graph autoencoders for business process anomaly detection. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 417–433. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_26

7th International Workshop on Business Processes Meet the Internet-of-Things (BP-Meet-IoT 2023)

7th International Workshop on Business Processes Meet Internet-of-Things (BP-Meet-IoT 2023)

The Business Process Management (BPM) discipline emerged in the mid-1990s, focusing on improving business processes. With the advent of the Internet of Things (IoT), which involves interconnected computing devices with sensing and actuating capabilities, there's potential to make business processes adaptive to their environment. Combining IoT and BPM can lead to innovative solutions beyond business, like in smart cities and healthcare. The objective of this workshop was therefore to attract novel research that tackles the challenges that emerge when these two fields are combined as well as to create a space for discussion and interactions between the research communities dealing with the integration between the IoT and BPM fields.

The 7th edition of this workshop was aligned with the 2nd International Workshop on Digital Twins for Business Processes (DT4BP) and started with a very insightful keynote by Wil van der Aalst, a very well-known scientist in the BPM community for the profound impact that his research has had and is having on the field of process mining and workflow management. His keynote, entitled "Data Models in the Middle", made clear the need to cover the gap between generated application-specific data and the different techniques from AI, ML, DS and tools and to address the differences found between these two "worlds" through the introduction of an intermediary layer based on models, specifically, the Object-Centric Event Data (OCED) and Measured and Event Data (MAED).

From the six international received submissions, each of them single-blind reviewed by three members of the program committee, the following five were finally accepted and selected for presentation:

An Object-centric Approach to Handling Concurrency in IoT-aware Processes, authored by Florian Gallik, Yusuf Kirikkayis, Ronny Seiger and Manfred Reichert. In this paper, the authors discuss the need for efficient coordination of concurrent processes involving physical resources in the context of Business Process Management (BPM) and IoT adoption.

An Event-Centric Metamodel for IoT-Driven Process Monitoring and Conformance Checking, authored by Marco Franceschetti, Ronny Seiger and Barbara Weber. In this paper, the authors discuss the challenge of analyzing process events for process monitoring and conformance checking when such events may not be readily available or suitable for analysis, especially in cases of manual or semi-automated processes not controlled by a Process-Aware Information System.

Process Mining on Sensor Location Event Data, authored by Dominik Janssen, Felix Mannhardt and Agnes Koschmider. In this paper, the authors discuss the need for efficient analysis of IoT sensor event data in domains like smart cities and logistics. To this end they propose a method for discovering processes from sensor location event data in single-occupancy settings using unsupervised learning through clustering to abstract events.

Viola: Detecting Violations of Behaviors from Streams of Sensor Data, authored by Gemma Di Federico, Andrea Burattin and Giovanni Meroni. In this paper, the authors discuss the use of sensor networks and the IoT to collect environmental data for various applications, such as monitoring the daily routines of individuals with dementia in healthcare. They mention the challenge of real-time monitoring using streaming conformance-checking techniques, which typically require a stream of activities rather than raw sensor data.

Analyzing Behavior in Cyber-Physical Systems in Connected Vehicles: A Case Study, authored by Harry Herbert Beyel, Omar Makke, Oleg Gusikhin and Willibrordus Martinus Pancratius van der Aalst. In this paper, the authors discuss the increasing data generated by internet-connected cyber-physical systems and how companies can use process-mining techniques to gain insights from this data. Specifically, they focus on the analysis of enriched and transformed data from connected vehicles with assistance systems.

The final sessions of the workshop consisted of a very lively discussion around the definition of Digital Twins and their relationship with IoT and Business Processes.

The workshop took place on the 11th of September, 2023, and attracted about 20 participants who actively interacted during the workshop presentations and the discussions. The organizers of this event would like to specially thank the authors of the submitted papers as well as the keynote and active participants in the fruitful discussions. We would also like to thank the PC members for their valuable input and the conference organizers who facilitated the workshop. We hope that the reader finds the final selection of papers interesting and useful to get a better insight into the integration of IoT and BPM from both theoretical and practical points of view.

September 2023

Agnes Koschmider
Francesco Leotta
Massimo Mecella
Estefanía Serral
Victoria Torres

Organization

BP-Meet-IoT 2023 Workshop Chairs

Agnes Koschmider	University of Bayreuth, Germany
Francesco Leotta	Sapienza Università di Roma, Italy
Massimo Mecella	Sapienza Università di Roma, Italy
Estefanía Serral	KU Leuven, Belgium
Victoria Torres	Universitat Politècnica de València, Spain

BP-Meet-IoT 2023 Program Committee

Adrian Mos	NAVER LABS, Grenoble, France
Andrea Delgado	Universidad de la República, Uruguay
Andreas Oberweis	Karlsruhe Institute of Technology, Germany
Felix Mannhardt	Eindhoven University of Technology, The Netherlands
Jochen De Weerdt	KU Leuven, Belgium
Jianwen Su	University of California at Santa Barbara, USA
Luise Pufahl	TU Munich, Germany
Mathias Weske	University of Potsdam, Germany
Pnina Soffer	University of Haifa, Israel
Ronny Seiger	University of St. Gallen, Switzerland
Udo Kannengiesser	Johannes Kepler University Linz, Austria
Vicente Pelechano	Universitat Politècnica de València, Spain
Yannis Bertrand	KU Leuven, Belgium
Zakaria Maamar	Zayed University, UAE

Experiences from the Internet-of-Production: Using "Data-Models-in-the-Middle" to Fight Complexity and Facilitate Reuse

Wil M. P. van der Aalst[1,2](✉)

[1] Process and Data Science, RWTH Aachen University, Aachen, Germany
wvdaalst@pads.rwth-aachen.de
[2] Celonis, Theresienstrasse 6, 80333 Munich, Germany
http://vdaalst.com

Abstract. Data-driven approaches play a key role in improving operational processes and production is no exception. The *Internet-of-Production* (IoP) is an ambitious initiative aiming at cross-domain collaboration in production while exploiting semantically adequate and context-aware data at different levels of granularity. The *Internet-of-Things* (IoT), in the context of production also referred to as *Industry 4.0* or the Industrial Internet of Things, provides a wide range of data assets. However, these are often handled in an ad-hoc manner with little support for reuse. Data pipelines convert machine- or system-specific data into a format suitable for data-science techniques such as machine learning. Based on an analysis of the data used in IoP, we developed so-called "Data-Models-in-the-Middle" (DMMs). Two such models are described in this paper: *Measurement and Event Data* (MAED) and *Object-Centric Event data* (OCED). OCED enables Object-Centric Process Mining (OCPM), allowing organizations to view their operational processes from any perspective using a single source of truth. However, OCED is not suitable for low-level machine data that contain a mixture of continuous measurements (e.g., time series data describing position, temperature, force, speed, etc.) and discrete events. Therefore, we also propose MAED as a data format. The combination of both "Data-Models-in-the-Middle" (MAED and OCED) provides a good coverage of many production-related use cases.

Keywords: Internet-of-Things · Internet-of-Production · Process Mining · Business Process Management · Reference Models

1 The Internet-of-Production

The *Internet-of-Production* (IoP) is a so-called *Cluster of Excellence* at RWTH Aachen University funded by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy. Over 200 researchers from 35 organizational units in production technology, production management, materials science, data science, computer science, social science, and management science

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 87–91, 2024.
https://doi.org/10.1007/978-3-031-50974-2_7

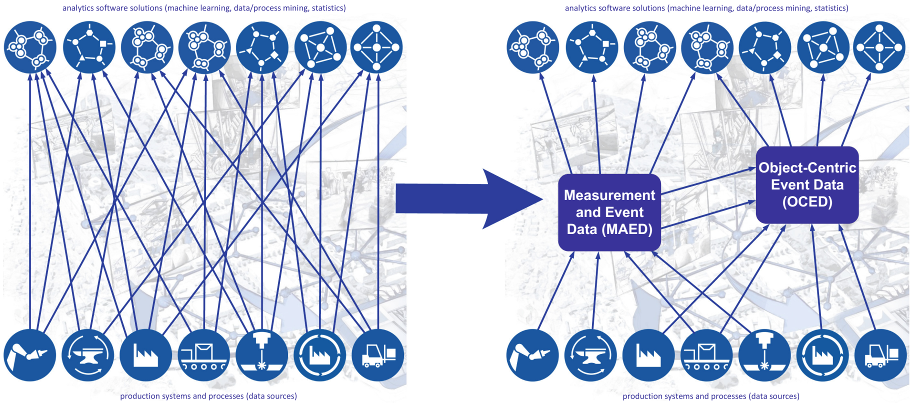

analytics software solutions (machine learning, data/process mining, statistics)
production systems and processes (data sources)
analytics software solutions (machine learning, data/process mining, statistics)
Measurement and Event Data (MAED)
Object-Centric Event Data (OCED)
production systems and processes (data sources)

Fig. 1. Production machines and systems produce specific data sets without explicit semantics and data pipelines are often ad-hoc with little reuse (left-hand side). We aim to tackle the problem using "Data-Models-in-the-Middle" (right-hand side) such as Measurement and Event Data (MAED) and Object-Centric Event Data (OCED).

have joined forces to create an infrastructure enabling a new level of cross-domain collaboration by providing semantically adequate and context-aware data related to production [3].

The two main concepts used in IoP are (1) the creation of a *World Wide Lab* (WWL) in which production engineering techniques, data, and software can be used across domains and (2) the use of so-called *Digital Shadows* (DS) capturing condensed knowledge by tightly connecting data and models [3]. To realize the ambitious IoP vision, there needs to be a unification of data and reuse of software. However, a systematic detailed analysis of over 80 data sets created and used in IoP, revealed that data are collected in ad-hoc one-of-a-kind formats and different organizational units tend to create "fragile data pipelines" to leverage data-science and machine learning tools. Therefore, we decided to focus on the creation of "Data-Models-in-the-Middle" as explained in the remainder.

2 Data-Models-in-the-Middle (DMMs)

Figure 1 illustrates both the problem and the envisioned solution. For each machine or system considered in the context of IoP, data are collected and analyzed using a range of techniques and software tools. However, these data sets tend to be one-of-a-kind and end-to-end pipelines tend to be created from scratch (left-hand side of Fig. 1). Note that the many connections on the left often correspond to complex manual data-transformation workflows. We advocate the use *Data-Models-in-the-Middle* (DMMs) as shown on the right-hand side of Fig. 1. These serve as a *decoupling point* between the application-specific data and focused analytics capabilities. The idea to unify data is not new. For example, the survey presented in [4] shows the usage of ontologies in such environments. However, the adoption of ontologies is limited, because it takes additional effort and the benefits are not immediately visible for a team analyzing a specific machine or system. In our view, a DMM should enable *immediate analytics capabilities*. Once the data is stored in the format of the DMM, it should be possible to visualize and analyze the data without further transformations and configurations. The idea is inspired by the recent uptake of *process mining* [1]. Given event data in the proper format, one can automatically generate process models, detect process-related problems (e.g., bottlenecks and anomalies), and predict performance (e.g., remaining processing time). By agreeing on basic concepts such as event, activity, case, timestamp, resource, etc. one gets access to a range of analytics capabilities. Note that event data have much more structure than tabular data fed into general-purpose tools. This allows for instant analysis results for a wide spectrum of operational processes in IoP and beyond.

We envision a set of 3–5 DMMs in the context of IoP. Here we briefly describe the two shown in Fig. 1: *Object-Centric Event data* (OCED) and *Measurement and Event Data* (MAED).

3 Example: Object-Centric Event Data (OCED)

Figure 2 shows a meta-model for *Object-Centric Event data* (OCED). Events are typed and may involve any number of objects. Also objects are typed and may be involved in any number of events. Events have a timestamp and any number of attributes. Objects do not have a timestamp, but can also have time-stamped attributes (e.g., price changes). Objects may be related (e.g., a part-of relation). Both the Event-to-Object (E2O) relations and Object-to-Object (O2O) may be qualified.

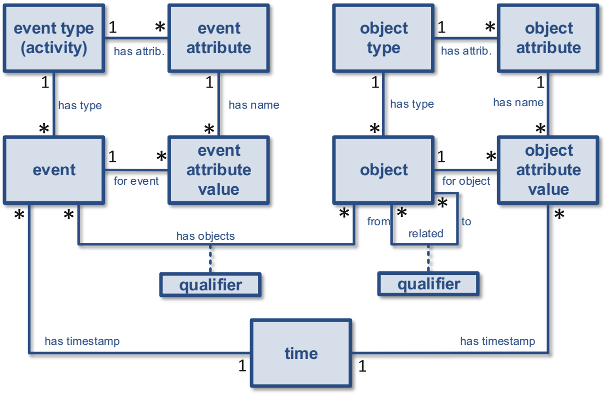


Fig. 2. Meta-model describing Object-Centric Event data (OCED).

Compared to traditional event data (stored for example in XES format [1]), OCED overcomes several limitations. For example, one event can involve any number of objects of different types (instead of a single case). However, it is still possible to automatically discover process models, expose performance and compliance problems, and predict dynamic behavior [2]. OCED enables *Object-Centric Process Mining* (OCPM) supported by tools such as OC-PM (www.ocpm.info), OCpi (ocpi.ai), and Celonis Process Sphere [2]. It is expected that the whole field will transition to OCPM.

4 Example: Measurement and Event Data (MAED)

OCED is limited to discrete events. However, in production, there are often continuous *measurements*, e.g., position, force, and temperature are monitored using a sampling rate. It does not make sense to see such measurements as events. Their occurrence carries no information; only the values measured do. Analyzing the data of many machines, we noted that they often contain a mix of discrete events and continuous measurements. This triggered the development of the *Measurement and Event Data* (MAED) meta-model shown in Fig. 3.

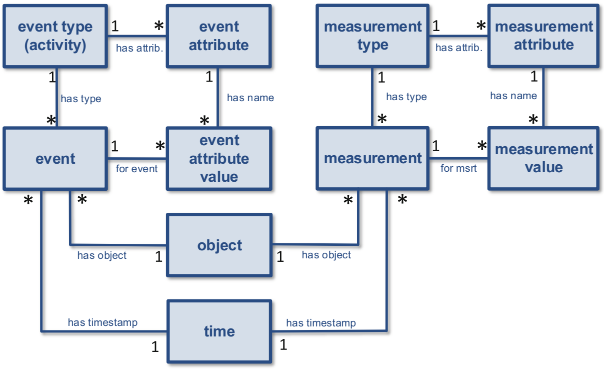


Fig. 3. Meta-model describing Measurement and Event Data (MAED).

Events and measurements are typed and have attributes. Both events and measurements are timed and refer to precisely an object (i.e., more restrictive than OCED). Although events and measurements look similar, they are handled completely different during analysis. Measurements are continuous and do not correspond to specific actions. For example, position, force, and temperature may be measured every 10 ms. Only the values matter. Events are discrete in nature and do not need to happen at specific times (e.g., the machine overheating or restarting). Therefore, we need both.

5 Conclusion

In this paper, we shared experiences from the Internet-of-Production (IoP) and suggested using “Data-Models-in-the-Middle” (DMMs) to fight complexity and facilitate reuse. We proposed two DMMs: Object-Centric Event data (OCED) and Measurement and Event Data (MAED). The former is mature and well-supported and the latter is still under development. Although the idea is not very original (see, for example, the many ontologies developed in the past [4]), we leverage experiences from process mining. Most ontologies do not provide dedicated types of analysis that can be used without further data transformations and configurations. This is in stark contrast with event data (stored in XES or OCEL format) which can be used to create process models immediately showing the actual process, including compliance and performance problems. Future work will also include supporting the transformation of MEAD into OCED. This aligns well with the challenges identified in [5], where the translation of low-level IoT data into events is seen as one of the key problems connecting IoT to Business Process Management (BPM). Moreover, in IoP we also want to standardize event types, object types, and measurement types. Compared to general ontologies, this our approach is more restrictive, but allows for immediate analytics capabilities without further data transformations.

Acknowledgement. Supported by the Alexander von Humboldt (AvH) Stiftung and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC Internet of Production, 390621612.

References

1. van der Aalst, W.: Process Mining: Data Science in Action. Springer, Berlin (2016). https://doi.org/10.1007/978-3-662-49851-4
2. van der Aalst, W.: Object-centric process mining: unraveling the fabric of real processes. Mathematics **11**(12), 2691 (2023)
3. Brauner, P., et al.: A computer science perspective on digital transformation in production. ACM Trans. Internet Things **3**(2), 15:1–15:32 (2022)
4. Ekaputra, F., Sabou, M., Serral, E., Kiesling, E., Biffl, S.: Ontology-based data integration in multi-disciplinary engineering environments: a review. Open J. Inf. Syst. **4**(1), 1–26 (2017)
5. Janiesch, C., et al.: The internet-of-things meets business process management: mutual benefits and challenges. CoRR abs/1709.03628 (2017)

Analyzing Behavior in Cyber-Physical Systems in Connected Vehicles: A Case Study

Harry H. Beyel[1](✉), Omar Makke[2], Oleg Gusikhin[2], and Wil M. P. van der Aalst[1]

[1] Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany
{beyel,wvdaalst}@pads.rwth-aachen.de
[2] Global Data Insight & Analytics, Ford, Dearborn, USA
{omakke,ogusikhi}@ford.com

Abstract. Cyber-physical systems embedded in various areas connected to the internet generate unprecedented volumes of data. Humans frequently interact with these systems, thus allowing companies to analyze the data and gain valuable insights into user-product interactions. To analyze the underlying behavior recorded in data, process-mining techniques can be used. However, to apply process mining, the low-level measurements have to be transformed into an event log. In this work, we analyze enriched and transformed connected-vehicle data dealing with an assistance system using process-mining techniques. We analyze the time spent in states of the system, compare behavior between different models using Kruskal-Wallis' and Dunn's test, and discover reasons for state switching. We demonstrate how companies can apply process mining on data collected by internet-of-things devices to understand the usage, ratify system requirements, and develop their products.

Keywords: Connected vehicle · Continuous Data · Sensor Data · Process Mining · Process Enhancement · Case Study

1 Introduction

Cyber-physical systems (CPSs) [17] and Internet of Things (IoT) [12] continue to emerge worldwide. CPSs focus on integrating physical components with computing and controlling systems for autonomous operation in real-time, while IoT emphasizes the interconnection of a wide range of devices and objects for data collection and communication. These systems are implemented in various domains, from household items to connected vehicles. Such devices generate a significant volume of data, and analyzing the overall systems' behavior and their corresponding human interactions is challenging. For example, the data from

---

This research is supported by Ford-RWTH Aachen University Alliance program. We thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 92–104, 2024.
https://doi.org/10.1007/978-3-031-50974-2_8

IoT devices are often low-level, but in many cases, companies are interested in understanding the higher-level interactions between humans and these devices to improve their products. Process-mining techniques can analyze recorded data to support understanding the underlying behavior of such IoT systems. Process mining has three areas: process discovery, conformance checking, and process enhancement [1]. Process-discovery techniques aim to discover a comprehensible process model that represents the underlying behavior in the data. Conformance-checking techniques quantify how well a process model represents the behavior in the recorded data. Process-enhancement techniques are used to generate more value based on a well-representative process model, such as observing bottlenecks. To apply process-mining techniques, data must be in the form of an *event log*—which is not often the case for data transmitted by IoT devices. Data from such devices may have timestamps that reveal when the recorded action occurred, yet no activity or case information is denoted in low-level measurements. Examples of such low-level data are a temperature sensor that might send data every 10 min or a vehicle that might record its speed every second. Such data have to be transformed into an event log where a measurement is associated with some identifier that can be tracked throughout the system. In contrast, consider a system for a given airport that records when a customer checks luggage in, scans the passport, boards the airplane, etc. Here, the identifier is a customer, and events are "checking the luggage", "scanning the passport", etc.

Using IoT data for process mining has challenges [13]. For example, identifying activities and processes associated with the recorded data may not be trivial. In this work, we focus on a subset of a hands-free driving system using connected test vehicles. The system allows the vehicle to be driven hands-free and without human intervention in defined zones as long as certain conditions are not violated. Vehicles equipped with this feature require connectivity and can receive over-the-air updates to improve the feature's behavior based on its usage. Therefore, it is important to develop methods that explain the feature's behavior by analyzing the connected vehicle data. On connected test vehicles equipped with a hands-free driving system, a data stream is recorded at a one-second rate and transmitted to the cloud using the vehicles' modem. Features of the collected data include an anonymized vehicle identifier, states and warnings of assistance systems, and a vehicle's speed. Analyzing the collected data is challenging for numerous reasons. One reason is that data are sent under different conditions, e.g., vehicles' software versions, sensors, ages, and connectivity conditions. Our goal is to analyze drivers' interactions with the system so that experts across various fields can interpret the results. In this work, analyzing interactions means investigating the amount of time spent in different states, discovering if there are differences between models concerning the usage of the system, and reasoning behind state-switching in the system. Therefore, we decided to analyze the data by using process-mining techniques. In Fig. 1, we illustrate how we use data to comprehend the underlying behavior of the system better. In [5], we describe how we preprocessed, enriched, and transformed the collected vehicle data. Moreover, we derived process models and evaluated and compared them. In this work, we concentrate on analyzing the behavior in greater detail. To analyze the behavior, we focus on a selection of process-enhancement techniques.

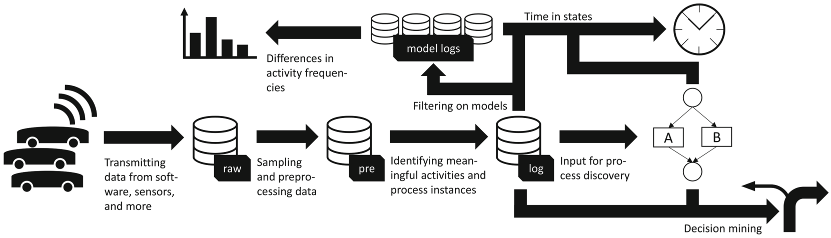

Time in states
model logs
Differences in activity frequencies
Filtering on models
Transmitting data from software, sensors, and more
raw
Sampling and preprocessing data
pre
Identifying meaningful activities and process instances
log
Input for process discovery
A
B
Decision mining

Fig. 1. Overview of our approach to gain insights into the behavior of a portion of the hands-free driving system.

This paper is structured as follows. In Sect. 2, we present preliminaries to this work. In Sect. 3, we discuss related work. We analyze the data in Sect. 4 by applying various process-mining techniques. In Sect. 5, we summarize and discuss our work and provide points for future work.

2 Preliminaries

In this section, we introduce the preliminaries of this work and refer to [5] for more details.

Given a set X, $\mathcal{B}(X)$ denotes the set of all multisets over set X. For example, if $X = \{x, y, z\}$, a possible bag is $[x, x, y] = [x^2, y]$. When combining multisets, we use $\uplus$, for instance, $[x, y] \uplus [x, z] = [x, y, x, z] = [x^2, y, z]$.

The received data recorded by the test vehicles are collected from US streets at a sample rate of one second. The data contain information related to vehicles' software and sensors. We sample, preprocess, and enrich the data as described in [5], resulting in the following features:

- *Vehicle*: Vehicle to which a datum belongs.
- *Model*: Model of vehicle. Four vehicle models exist: A, B, C, and D.
- *Journey*: A journey is created as soon as a recording of a vehicle takes place. Moreover, journeys are generated whenever the logger restarts recording. This means that traveling from one place to another can involve multiple journeys. Each journey is linked to a vehicle.
- *Run*: A run identifies the beginning of hands-free driving and its turn-off. Multiple runs can be contained in one journey.
- *Speed*: Vehicle speed in km/h.
- *Acceleration*: Acceleration of a vehicle in m/s^2.
- *Normalized acceleration*: Normalized acceleration of a vehicle over the corresponding journey. Min-max-normalization is performed.

- *State*: States of the traffic jam assist. We consider four states: hands-free driving and states one, two, and three.
- *Warning*: Different warnings are stored in this feature.
- *Timestamp*: Local timestamp of vehicle for each data recording.
- *Traffic jam*: By combining speed and acceleration entries, and timestamps, we approximate if a vehicle is in a traffic jam. The values are either 0 or 1.

We use these data to analyze the drivers' behavior and interaction with a portion of the hands-free driving system. To analyze drivers' interactions using process-mining techniques, the data log has to be transformed into an event log. An event log consists of at least three mandatory attributes: *case*, *activity*, and *timestamp*. We describe the transformation in [5]. In this earlier work, we defined and discovered maximal sequences of connected-vehicle data. Each maximal sequence is transformed into an event. For this conversion, we use runs as cases, rely on domain knowledge to identify activities, and treat a maximal sequence's first timestamp as the timestamp for an event. In this process, we ensure that we store additional information in events' features, e.g., the average speed during the underlying maximal sequence. The following provides a short overview of the additional features:

- *avg-acc*: average vehicle acceleration during the maximal sequence.
- *l-acc*: last acceleration recording in a maximal sequence.
- *avg-n-acc*: average normalized acceleration in the maximal sequence. Normalization is performed over the whole journey using min-max-normalization.
- *l-n-acc*: last normalized acceleration recording in the maximal sequence. Normalization is performed over the whole journey using min-max-normalization.
- *avg-speed*: average speed during the maximal sequence.
- *l-speed*: last speed recording in the maximal sequence.
- *avg-traffic*: average traffic jam situation during the maximal sequence
- *l-traffic*: last traffic jam situation in the maximal sequence.
- *model*: the vehicle model from which the maximal sequence was transmitted.

In the following, we define event logs formally. $\mathcal{U}_{case}$ is the universe of cases, $\mathcal{U}_{time}$ is the universe of timestamps, $\mathcal{U}_{act}$ is the universe of activities, $\mathbb{R}$ is the set of real numbers, $\mathbb{B}$ is the Boolean domain, and $M = \{A, B, C, D\}$ is the set of models.

Definition 1 (Event Log). *$\mathcal{U}_{ev}$ is the universe of events. $e \in \mathcal{U}_{ev}$ is an event such that $\pi_{act}(e) \in \mathcal{U}_{act}$ is the activity of e, $\pi_{case}(e) \in \mathcal{U}_{case}$ is the case of e, and $\pi_{time}(e) \in \mathcal{U}_{time}$ is the timestamp of e. Furthermore, $\pi_{model}(e) \in M$ is the model of e. Moreover, $\pi_{avg-acc}(e) \in \mathbb{R}$, $\pi_{l-acc}(e) \in \mathbb{R}$, $\pi_{avg-n-acc}(e) \in \mathbb{R}$, $\pi_{l-n-acc}(e) \in \mathbb{R}$, $\pi_{avg-speed}(e) \in \mathbb{R}$, $\pi_{l-speed}(e) \in \mathbb{R}$, $\pi_{avg-traffic}(e) \in [0, 1]$, and $\pi_{l-traffic}(e) \in \mathbb{B}$ represent the features mentioned before. An event log L is a set of events $L \subseteq \mathcal{U}_{ev}$.*

After applying all the mentioned steps to our sampled connected-vehicle data, we obtained an event log of more than 11,000 cases (i.e., runs) and roughly 26,000 events. An example event log is shown in Table 1. In the remainder of this work, we analyze the behavior of models and compare between models. Therefore, we introduce model event logs.

Definition 2 (Model Event Log). *Let* $m \in M$ *be a model and* $L \subseteq \mathcal{U}_{ev}$ *be an event log. A model event log* $L_m \subseteq L$ *is an event log that only contains events related to model* m*, i.e.,* $L_m = \{e \in L | \pi_{model}(e) = m\}$*.*

Table 1. Example fragment of the (anonymized) event log.

Case	Activity	Timestamp	Model
1	System started	13:37:10	A
1	Warning 1	13:45:30	A
1	Eyes on	13:45:32	A
1	Warning 1	13:58:10	A
1	State 1	13:58:13	A
2	System started	14:30:10	A
2	Warning 2	14:31:34	A
2	State 2	14:31:35	A
3	System started	16:45:32	B
3	State 1	16:58:10	B

By applying this filter to the event log shown in Table 1 concerning model A, we receive an event log containing cases one and two.

In the remainder of this paper, we use Petri nets as a process-model representation. An introduction is given in [1,20]. Furthermore, we use process-enhancement techniques, including decision mining and performance analysis. Decision mining helps to understand the reasoning behind choices presented in a Petri net based on provided data. We apply performance analysis to observe vehicles' time in states, e.g., hands-free driving. More information concerning process-enhancement techniques can be found in [1].

In [5], we picked the model discovered by region-based mining as the best de-facto model based on domain experts' opinions and quality metrics. Although we have different data for this work due to different filters, applying region-based mining with the same settings as in [5] on the received event log resulted in the model with the best scores according to quality metrics. The model is depicted in Fig. 2, and its score for fitness is 0.99, the score for precision is 0.77, and the generalization's score is 1.00. We use alignments to compute the fitness score [3]. Based on these alignments, we compute precision and generalization scores [2]. In the following, we briefly describe the model's behavior shown in Fig. 2. Each place in the Petri net marks a state in the hands-free driving system. After firing "System started", the system is in a hands-free driving state (*p1*). If any transition labeled with "Warning 1" fires, the state where this warning is active is reached (*p2*). The same holds for "Warning 2" (*p3*) and "Warning 3" (*p5*).

In the limited part of the system that we consider for this analysis, firing any transition labeled with "Eyes on" leads to a conflict-free hands-free-driving state (*p4*). If any transition labeled with "State 1", "State 2", or "State 3" is executed, the system turns off, and the final marking is reached (*p6*).

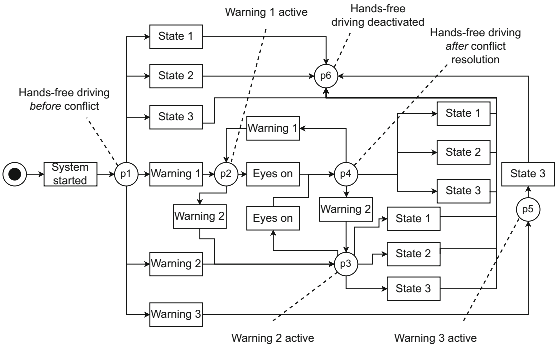


Fig. 2. Petri net obtained by using the region-based miner. The underlying transition system was built using the event name as the backward key, set abstraction with size one, and removal of self-loops. Also, label-splitting was used. Annotations describe the assistance system's state if a token is in a corresponding place.

3 Related Work

In this section, we present and discuss related work. A general framework for process discovery from sensor data is presented in [15]. Given location sensor event data, events are grouped into activity instances through correlation. Afterward, activities are discovered. Then, events are abstracted based on the groups and the discovered activities. Finally, processes can be discovered. Our study does not deal with location data, but we also have to define activities and process instances. In [19], visual process maps for habit mining are presented. Sensor data are collected from a smart environment so that one can track a person and create an event log using a visualization tool. To create an event log, sensor measurements are aggregated into sub-trajectories that represent movement actions. The created log can be used as input for process mining. Different views are possible, e.g., the complete daily routine of a person or smaller routines. In our study, we also aggregate data to show how the hands-free driving system works. Moreover, we can investigate different views based on how we define a process instance. In [10], recordings of multiple sensors are segmented by using windows. For each segment, features are extracted, and based on these features, clusters are created and labeled based on domain knowledge. Then, the segments are grouped together to create activities. To create process instances, domain knowledge is required. In our work, we do not rely on clusters to identify activities. However, we also extract additional information, such as acceleration and whether a vehicle is in a traffic jam. In [7], sensor recordings from a smart environment are used to discover process models for daily routines and process models for macro activities. The input data for their approach are sensor data that shows if an activation happened or not. Based on domain knowledge, macro activities are added. Traces are created based on the recordings' timestamps. Connected-vehicle data differs from the sensor data of this approach; in particular, we do not deal with locations. Yet, we also create activities based on activating and deactivating the feature's parts. In [14], a case study deals with integrating Business Process Management (BPM) and IoT in Australia's meat and livestock industry. The study does not reveal how challenges were overcome in detail, but it reveals how the industry benefits from combining BPM with IoT. For example, farmers can confidently plan daily activities through the embedded system's support. Based on our analysis, original equipment manufacturers such as Ford can also benefit since the behavior of users is better understood. When the behavior is better understood, the product can be adjusted to users' needs. In [4], data was collected from users interacting with a software. Using process mining, the behavior of users was analyzed. Based on the analysis, improvements for the software can be defined. Our work also aims to better understand users to implement improvements. Also, instead of relying on process discovery and filtering, we apply process enhancement. By applying process-enhancement techniques, which have not been performed in our previously mentioned work [5], we are sure to gain more insights into users' interactions.

4 Analysis of Connected-Vehicle Data

In this section, we use transformed connected-vehicle data to get insights into drivers' interaction with the hands-free driving system. First, we inspect how much time is spent in the system's states. Second, we investigate whether the model type influences the number of activities statistically significant. Therefore, we apply significance tests to measure differences to ensure that these are relevant. Third, we investigate the reasons for switching states by considering multiple features.

4.1 Time in States

Our goal is to analyze how much time is spent in the different states of the hands-free driving system. We have two reasons to carry out this analysis. First, we check if the time spent in different warning states meets the system's requirements. Second, we observe how long drivers are in certain states. These observations can be used to improve the product. By also considering each vehicle model separately, we compare them. We use ProM [8] to measure the time spent in states. For this analysis, we considered the model shown in Fig. 2 as the process model. As event logs, we used the whole transformed event log and four model event logs, each representing a different vehicle model (see Sect. 2). We computed the minimum, maximum, average, and standard deviation of spent

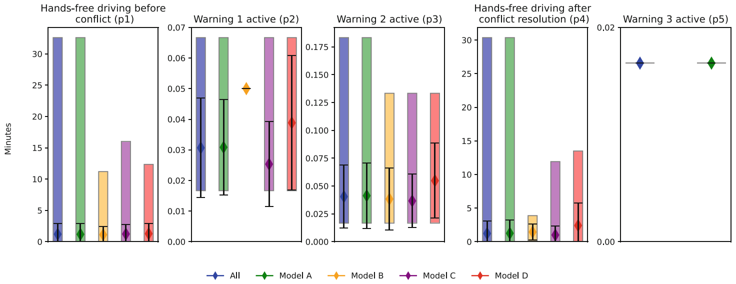


Fig. 3. Overview of time spent in different places across models. Diamonds denote the averages, and whiskers denote standard deviations.

time for each place. An overview of our measurements is depicted in Fig. 3. In general, the results must be treated with caution due to potentially erroneous data, e.g., based on sampling and connectivity issues. Nevertheless, the time spent in warning states, i.e., places *p2*, *p3*, and *p5* fit the defined requirements. Moreover, we denote that drivers spent different amounts of time with hands-free driving (places *p1* and *p4*). On average, drivers spend roughly one minute in such a state. Considering the small standard deviation, we conclude that the region is relatively dense. However, we cannot conclude why drivers exit the hands-free state after that time. Between the different vehicle models, we notice that model D is often above the other averages, and its standard deviation is greater than the others. Model C's standard deviation values are usually the smallest, implying a more coherent behavior. In addition, we observe that model A is responsible for the overall maximal values. In comparison, the maximal values for other models are small. Also, we deduce that "Warning 3" happens only once, which is the case for model A. The minimal values are (except once) the same.

4.2 Difference in Activity Frequencies

Different models may have different types of drivers, e.g., sporty and off-road drivers. Moreover, each model has different components and software. Therefore, we investigate if the model type influences behavior. Our approach is inspired by [6], yet, we only consider the number of occurrences of an activity.

First, we count the occurrences for each activity in a given case (i.e., run).

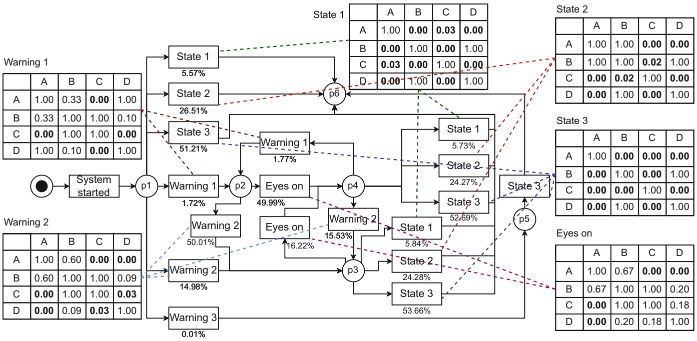
Definition 3 (Occurrence count: case). *Let* L *be an event log.* $count_c : \mathcal{U}_{act} \times \mathcal{U}_{case} \times \mathcal{P}(L) \rightarrow \mathbb{N}_0$ *is a function that counts how often an activity occurs in a given case for a given log, i.e.,* $count_c(act, case, L) = |\{e \in L | \pi_{case}(e) = case \wedge \pi_{act}(e) = act\}|$.

In the next step, we perform this action on all cases of a given model event log, resulting in a multiset, for which each element denotes the number of occurrences of the given activity in a case.

Definition 4 (Occurrence count: event log). *Let* L *be an event log, then* $C_L = \{\pi_{case}(e) | e \in L\}$ *provides all cases of an event log* L. $count_l : \mathcal{U}_{act} \times \mathcal{P}(L) \rightarrow B(N_0)$, *returns a multiset of case-wise occurrences of an activity in an event log, i.e.,* $count_l(act, L) = \biguplus_{case \in C_L} [count_c(act, case, L)]$.

To illustrate the definition, consider the event log shown in Table 1. Applying $count_l$ on this event log with activity "Warning 1" results in the multiset $[2, 0, 0] = [2, 0^2]$.

In the event log that we received from the provided vehicle data, "System started" appears exactly once in any case. Therefore, this activity is not interesting for our investigation. "Warning 3" happened exactly once in the overall event log, so this activity is also uninteresting. As a result, we focus on "Warning 1", "Warning 2", "State 1", "State 2", "State 3", and "Eyes on". Our analysis is conducted as follows. First, we create four model event logs, one for each model. Second, we apply $count_l$ on the four model event logs and the mentioned activities. This results in 24 multisets. Third, based on these multisets, we investigate differences for each activity across all models.



Warning 1

	A	B	C	D
A	1.00	0.33	**0.00**	1.00
B	0.33	1.00	1.00	0.10
C	**0.00**	1.00	1.00	**0.00**
D	1.00	0.10	**0.00**	1.00

Warning 2

	A	B	C	D
A	1.00	0.60	**0.00**	**0.00**
B	0.60	1.00	1.00	0.09
C	**0.00**	1.00	1.00	**0.03**
D	**0.00**	0.09	**0.03**	1.00

State 1

	A	B	C	D
A	1.00	**0.00**	**0.03**	**0.00**
B	**0.00**	1.00	**0.00**	1.00
C	**0.03**	**0.00**	1.00	**0.00**
D	**0.00**	1.00	**0.00**	1.00

State 2

	A	B	C	D
A	1.00	1.00	**0.00**	**0.00**
B	1.00	1.00	**0.02**	1.00
C	**0.00**	**0.02**	1.00	**0.00**
D	**0.00**	1.00	**0.00**	1.00

State 3

	A	B	C	D
A	1.00	**0.00**	**0.00**	**0.00**
B	**0.00**	1.00	**0.00**	1.00
C	**0.00**	**0.00**	1.00	**0.00**
D	**0.00**	1.00	**0.00**	1.00

Eyes on

	A	B	C	D
A	1.00	0.67	**0.00**	**0.00**
B	0.67	1.00	1.00	0.20
C	**0.00**	1.00	1.00	0.18
D	**0.00**	0.20	0.18	1.00

Fig. 4. Annotated process model based on Fig. 2. Under each outgoing transition from a decision point, we denote the relative frequency of the activity's execution. The tables show the adjusted p-values after applying Dunn's test. The values in bold signify statistically significant differences.

To check for differences, we applied the Kruskal-Wallis test [16]. The Kruskal-Wallis test is a non-parametric version of the ANOVA test [11]. If there is a statistically significant difference in the number of occurrences, we applied Dunn's test [9] with Bonferroni adjustment to check which model types differ. The adjusted p-values are shown in the tables depicted in Fig. 4. As we deduce from the values, by setting $\alpha = 0.05$, each activity reveals differences among the models. However, differences between the models differ between the various activities. For example, models A and B differ in "State 1" and "State 3" but do not differ in all other activities. We observe this since the p-values in the tables in Fig. 4 for "State 1" and "State 3" are below the threshold of 0.05; but the p-values are greater than 0.05 in all other tables. Models A and D differ in all activities' frequencies but not for "Warning 1". Models C and D differ in all activities' frequencies but not for "Eyes on". Models A and C differ across all activities, but models B and D never differ. Overall, this analysis reveals statistically significant differences between the models regarding their underlying behavior. Also, it shows where behavior does not differ. However, the reasons are unknown. Limiting the analysis only to the model type can lead to erroneous results since, for instance, environmental conditions are not considered. Furthermore, a small group of drivers can have a great impact since they can produce many cases (i.e., runs).

Table 2. Highest measured accuracy of prediction measured using multiple strategies for different places in the model shown in Fig. 2.

Place	Naive Bayes	Generalized Linear Model	Logistic Regression	Fast Large Margin	Deep Learning	Decision Tree	Random Forest	Gradient Boosted Trees	Support Vector Machine
p1	51.1	52.1	52.1	-	**52.5**	51.1	52.2	51.9	51.3
p2	75.3	70.5	70.6	71.7	**79.8**	72.5	69.8	78.7	49.7
p3	40.0	59.9	61.3	61.5	**63.7**	63.0	63.2	57.0	63.4
p4	53.4	51.3	53.0	53.1	**53.5**	53.2	52.9	53.7	52.9

4.3 Decision Mining

As shown previously, there are differences in activity frequencies. Therefore, we investigate the reasons for changes in behavior. To investigate reasons for changes, we analyze decision points. Decision points are places in a Petri net with multiple outgoing arcs. The model depicted in Fig. 2 has the following decision points: *p1*, *p2*, *p3*, and *p4*. We used PM4Py[1] to prepare the data that is used as input for RapidMiner[2]. The data is prepared as described in [18]. We use multiple features: the model type and all acceleration, speed, and traffic jam information we introduced earlier. We only consider information right before the decision. In RapidMiner, we use the auto-model functionality to predict the next activity for each decision point. Using auto-modeling in RapidMiner, different techniques with various parameter settings are applied to the data. The results are summarized in Table 2. A label is the outcome of a decision, for instance, "Warning 1". We refer to Fig. 4 for the label's distribution for each place. For *p2*, the labels are equally distributed, yet we observe the highest accuracy. By analyzing the models for this decision, we observed that speed and acceleration information significantly influence the outcome. In general, if the last speed value before a decision is low, "Warning 2" is the next activity; otherwise, the next activity is "Eyes on". A possible explanation is that drivers focus more on the road and care more about warnings if their vehicles are fast. However, accuracy values are similar to majority voting across the other places. This shows that the behavior cannot be well explained using our data. On the one hand, the

---

[1] https://pm4py.fit.fraunhofer.de/.

[2] https://rapidminer.com/.

uncertainty of explaining behavior is great because it reveals that there are, e.g., no failing components included in a specific model type. On the other hand, we cannot guarantee that there are no factors that influence the system. As an example, our analysis does not consider the environment, for instance, weather conditions. Also, we do not have access to information that shows, e.g., whether a curve is driven or the construction year of a vehicle. Therefore, we have limited features for further diagnostics.

5 Discussion

This paper presents how we enriched data recorded by connected vehicles and transformed these into an event log. We applied various process-mining techniques to gain insights into drivers' interactions with the hands-free driving system. First, we compared the time spent in states of the discovered process model between the vehicle models. We observed which model is responsible for maximal values, which has the greatest standard deviations, and the average time spent in states. Moreover, we used these insights to ratify system requirements and draw conclusions on drivers' behavior. Second, we investigate differences between the models by counting the occurrences of different activities per trace and applying statistical tests. Our analysis shows that there are multiple significant differences between the models but also similarities. Third, we applied decision mining to investigate reasons for drivers' behavior using numerous features. Our insights can be used to ratify the hands-free driving system and to develop it further. For example, the system can be calibrated based on a user's needs. In general, our work shows that applying process-mining techniques to analyze behavior in cyber-physical systems is valuable. Similar approaches can be helpful in other cases to investigate behavior in the cyber-physical domain. Focusing on detecting differences between different systems of the same kind can lead to interesting insights. However, our accuracy concerning the reasoning of decisions is rather low. One possible reason is that we need more features, such as weather conditions. Another possible reason is that there is no underlying logic behind the decision. Overall, we applied data-driven techniques. Thus, if we deal with incomplete data or contextual information is missing, the accuracy of our results is limited. As stated, the analyzed event log is rather small since we analyzed data transmitted by test vehicles. Having more events may validate our results. At the same time, there are ethical concerns when collecting and analyzing data from connected vehicles. Aspects like the privacy of drivers have to be considered when analyzing data. However, discussing these is beyond the scope of this work. Nevertheless, more and deeper analyses using process mining are possible. For example, using more features of the recorded data can lead to more detailed activities, which leads to deeper insights. In addition, concerning analyzing the time spent in states, more detailed statistics are beneficial for a more in-depth analysis. Also, extending the idea presented in [6] to compare more than two groups is an open point for future work.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, 2nd edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. Adriansyah, A.: Aligning observed and modeled behavior. Ph.D. thesis, Mathematics and Computer Science (2014). https://doi.org/10.6100/IR770080
3. Adriansyah, A., van Dongen, B.F., van der Aalst, W.M.P.: Conformance checking using cost-based fitness analysis. In: EDOC, pp. 55–64. IEEE Computer Society (2011). https://doi.org/10.1109/EDOC.2011.12
4. Astromskis, S., Janes, A., Mairegger, M.: A process mining approach to measure how users interact with software: an industrial case study. In: ICSSP, pp. 137–141. ACM (2015). https://doi.org/10.1145/2785592.2785612
5. Beyel, H.H., Makke, O., Yuan, F., Gusikhin, O., van der Aalst, W.M.P.: Analyzing cyber-physical systems in cars: a case study. In: DATA, pp. 195–204. SCITEPRESS (2023). https://doi.org/10.5220/0012136000003541
6. Bolt, A., de Leoni, M., van der Aalst, W.M.P.: A visual approach to spot statistically-significant differences in event logs based on process metrics. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) CAiSE 2016. LNCS, vol. 9694, pp. 151–166. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39696-5_10
7. Cameranesi, M., Diamantini, C., Potena, D.: Discovering process models of activities of daily living from sensors. In: Teniente, E., Weidlich, M. (eds.) BPM 2017. LNBIP, vol. 308, pp. 285–297. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74030-0_21
8. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H.M.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The ProM framework: a new era in process mining tool support. In: Ciardo, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536, pp. 444–454. Springer, Heidelberg (2005). https://doi.org/10.1007/11494744_25
9. Dunn, O.J.: Multiple comparisons using rank sums. Technometrics **6**(3), 241–252 (1964). https://doi.org/10.2307/1266041
10. van Eck, M.L., Sidorova, N., van der Aalst, W.M.P.: Enabling process mining on sensor data from smart products. In: RCIS, pp. 1–12. IEEE (2016). https://doi.org/10.1109/RCIS.2016.7549355
11. Fisher, R.A.: Statistical methods for research workers. In: Kotz, S., Johnson, N.L. (eds.) Breakthroughs in Statistics, pp. 66–70. Springer, New York (1992). https://doi.org/10.1007/978-1-4612-4380-9_6
12. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (IoT): a vision, architectural elements, and future directions. Future Gener. Comput. Syst. **29**(7), 1645–1660 (2013). https://doi.org/10.1016/j.future.2013.01.010
13. Janiesch, C., et al.: The internet of things meets business process management: a manifesto. IEEE Syst. Man Cybern. Mag. **6**(4), 34–44 (2020). https://doi.org/10.1109/MSMC.2020.3003135
14. Keates, O.: Integrating IoT with BPM to provide value to cattle farmers in Australia. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) BPM 2019. LNBIP, vol. 362, pp. 119–129. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37453-2_11
15. Koschmider, A., Janssen, D., Mannhardt, F.: Framework for process discovery from sensor data. In: EMISA, pp. 32–38. CEUR-WS.org (2020)
16. Kruskal, W.H., Wallis, W.A.: Use of ranks in one-criterion variance analysis. J. Am. Stat. Assoc. **47**(260), 583–621 (1952). https://doi.org/10.2307/2280779

17. Lee, E.A.: Cyber physical systems: Design challenges. In: IEEE (ISORC), pp. 363–369. IEEE Computer Society (2008). https://doi.org/10.1109/ISORC.2008.25
18. de Leoni, M., van der Aalst, W.M.P., Dees, M.: A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. Inf. Syst. **56**, 235–257 (2016). https://doi.org/10.1016/j.is.2015.07.003
19. Leotta, F., Marrella, A., Mecella, M.: IoT for BPMers. Challenges, case studies and successful applications. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 16–22. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_3
20. Reisig, W.: Petri Nets: An Introduction. EATCS Monographs on Theoretical Computer Science, vol. 4. Springer, Heidelberg (1985). https://doi.org/10.1007/978-3-642-69968-9

An Object-Centric Approach to Handling Concurrency in IoT-Aware Processes

Florian Gallik[1], Yusuf Kirikkayis[1], Ronny Seiger[2](✉), and Manfred Reichert[1]

[1] Institute of Databases and Information Systems, Ulm University, Ulm, Germany
{florian-1.gallik,yusuf.kirikkayis,manfred.reichert}@uni-ulm.de
[2] Institute of Computer Science, University of St. Gallen, St. Gallen, Switzerland
ronny.seiger@unisg.ch

Abstract. The increasing adoption of IoT in the context of Business Process Management (BPM) makes it necessary to efficiently coordinate concurrent processes and activities that involve physical resources. Traditional approaches to handling concurrency in BPM systems are not suitable for automating IoT-aware processes due to novel challenges raised by the IoT. We propose to handle concurrency in IoT based on object-centric processes implemented in the PHILharmonicFlows framework. The framework facilitates the data-driven modeling and coordination of object lifecycles and interactions, which are suitable to address concurrency in IoT-aware processes. The approach is demonstrated in a scenario from smart manufacturing. The results show that PHILharmonicFlows offers a flexible and comprehensible solution for coordinating concurrent activities in IoT settings with constrained physical resources.

Keywords: Internet of Things · Business Process Management · Concurrency · Object-centric Processes · Cyber-physical Systems

1 Introduction

The Internet of Things (IoT) is increasingly adopted in the context of Business Process Management (BPM) [10]. Data from IoT sensors can be used complementary to traditional event log data to improve process analysis and decision making [11,16]. In turn, BPM technologies experience a growing application in IoT use cases, e.g., for modeling and execution of *IoT-aware processes* [12]. BPM systems (BPMS) provide sophisticated means for modeling and automating processes involving IoT devices, digital resources, and humans. However, using BPMS for coordinating processes that involve the active control of IoT devices raises new challenges. The physical resources (actuators) are often constrained and unable to execute multiple activities concurrently [7,20]. This requires additional means for coordinating access to these resources [26]. Traditional techniques (e.g., worklists) are not efficient as they have not been designed for full automation, which may be required by the IoT use case (e.g., in smart manufacturing). In the context of IoT, related approaches either do not address the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 105–117, 2024.
https://doi.org/10.1007/978-3-031-50974-2_9

aspect of handling concurrency in IoT-aware processes [12], or they rely on complex hard-wired logic implemented by an additional coordination component as part of a control system for the IoT devices [21]. Without proper coordination of concurrent activities, the execution of IoT-aware processes in the physical world might be interrupted and lead to unforeseen behavior and exceptions.

In this work, we propose an approach to handling concurrency in IoT-aware processes based on the object-centric process management framework *PHILharmonicFlows* [13]. This framework provides components for the modeling, execution, and coordination of data-driven object lifecycles. In particular, *lifecycle processes* can be leveraged for handling concurrency in combination with business processes. We use PHILharmonicFlows to model both the lifecycles of IoT devices and coordination processes enabling the proper management of concurrent process/activity executions while ensuring certain coordination constraints. A scenario from the domain of smart manufacturing is used as running example.

The paper is structured as follows: Sect. 2 introduces the research problem and research question. Section 3 presents the approach for handling concurrency using PHILharmonicFlows. Section 4 discusses our work in comparison to related work. Section 5 summarizes the paper and gives an outlook.

2 Problem Statement and Research Question

In traditional business processes not involving IoT, **digital** resources such as databases and web services can be accessed repeatedly and concurrently as the underlying software infrastructure (e.g., web servers or database management systems) provides scalability and coordination [9]. Usually, the coordination of **human** tasks relies on a worklist-based approach, i.e., process participants can view and retrieve open tasks from a list. This type of coordination is not suitable in an IoT context as the efficient automation is one of the main goals of IoT-aware processes, without involving human actors. In IoT-aware processes for which certain **physical** resources (actuators) exist only once and which cannot always be scaled up (e.g., a robot or an oven), multiple process activities may not be able to use the respective resources concurrently [7,20]. Moreover, the execution of activities that involve physical resources might be long-running and prevent the resources from being used by other activities. In traditional manufacturing setups, the handling of concurrent requests to the same machine is part of the Manufacturing Execution Systems (MES) which, for example use queues for each machine [25].

In this work we assume the existence of BPMS used for orchestrating IoT-aware processes. Related approaches often rely on (1) the capabilities of BPMS to handle parallel activity executions [19]; (2) the services/control applications for the IoT devices to handle concurrent access [14], which is usually incomprehensible for the user; or (3) additional coordination components hard-coded with a programming language, which is also incomprehensible and inflexible [18,21].

Smart Manufacturing Example: Figure 1 illustrates a simple scenario of a smart factory [12]. Two processes include the same type of activity executed by the

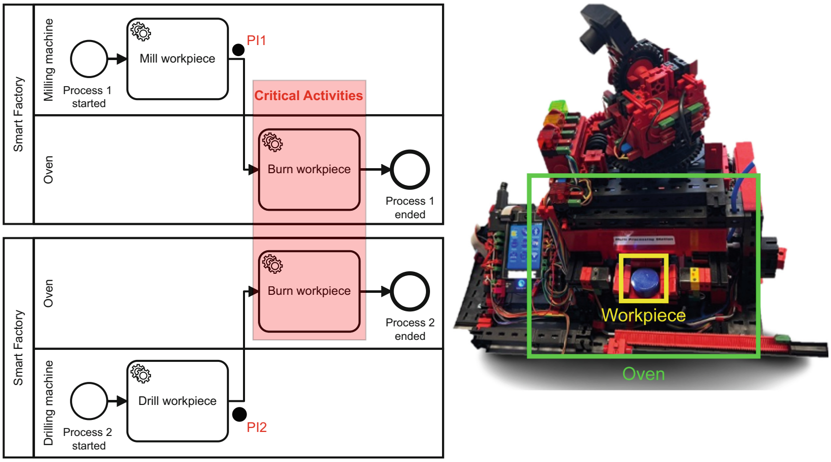


Fig. 1. Two IoT-aware process instances (in BPMN 2.0) accessing the same resource

same resource: in *Process 1*, the workpiece is milled first and then burned in the Oven; in *Process 2*, the workpiece is drilled first and then burned in the Oven. Critical here are the *Burn workpiece* activities in both processes as they require exclusive access the same physical resource *Oven* to execute the same activity. Similar problems can be found when executing the same activities concurrently in one process instance denoted as *batch activities* [18].

As shown, the use of BPMN 2.0 for modeling both processes does not make the problem of managing concurrency explicit. The depicted BPMN 2.0 models only represent a static view on the processes without any restrictions regarding their runtime behavior and involved resources [27]. As soon as two or more process instances of Process 1 and/or Process 2 concurrently reach the point of executing the *Burn workpiece* activity (*Multiple Instance Patterns* [2]), the runtime behavior cannot be reliably determined. Moreover, executing the activity *Burn workpiece* might take several hours, which blocks the resource for other process instances as only one workpiece can be burned at a time. *Petri nets* have been used for many years to make concurrency in processes explicit and analyzable [28]. However, Petri nets are mostly used to model and analyze the *control flow* of processes, which means that the executions of all process/activity instances have to be analyzed at runtime to identify potential conflicts [1]. Petri nets may be used complementary to processes to model the behavior of specific resources (e.g., production machines) when concurrency arises [28]. Nevertheless, the specification of constraints and dependencies with other resources, processes and materials as required by a state-of-the-art process management system fostering flexible IoT-aware processes is not well supported [3,10].

In summary, the coordination of concurrent access to physical resources that have limited parallel processing capabilities poses new challenges emerging with the adoption of BPM in IoT. This paper investigates the following research question: **How to coordinate concurrently executed activities in IoT-aware processes with constrained physical resources?** We aim at developing a generic solution that can be used to flexibly and comprehensibly coordinate concurrent process activities in IoT.

3 Handling Concurrency in IoT-Aware Processes

3.1 Object-Centric Processes

We investigate how concepts related to *Object-centric Processes* can be applied to handle concurrency in IoT-aware processes. Object-centric processes focus on the interactions and relations among objects [4]. These processes make the coordination of activities and interactions among objects explicit, which makes them suitable candidates for IoT settings where physical resources are constrained and need to be coordinated. *PHILharmonicFlows* is a process management framework focusing on object-centric processes [3]. It has proven to be well suited for modeling relations, interactions, and constraints among digital objects, data, and processes, with the main goal of achieving a high degree of flexibility [3,13]. PHILharmonicFlows leverages resource awareness and enables granular control and dynamic adaptation through a data-driven approach [3].

3.2 Relational Process Type Structure

In PhilharmonicFlows, the *Burn workpiece* process activity (cf. Fig. 1) consists of multiple *Objects* that need to be coordinated:

- Production: a representation of the burning process of workpieces.
- Workpiece: a representation of the physical item to be processed.
- Oven: a limited physical resource responsible for burning the workpieces.

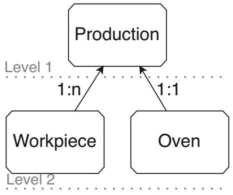


Fig. 2. Relational Process Type structure for the running example

A *Relational Process Type* structure is used to capture the involved objects and their interactions and relationships [22]. This structure serve as a foundation for coordinating the objects listed above (cf. Fig. 2). A Production object instance has a relation to exactly one Oven object instance and at least one, but potentially multiple Workpiece object instances (indicated by the min:max cardinalities). The directed relations between objects create a hierarchy within the relational process type structure. As shown in Fig. 2, Production is a *higher level* object (Level 1), while Oven and Workpieces are *lower level* objects (Level 2).

3.3 Object Lifecycles

PHILharmonicFlows incorporates the concept of objects into its data-driven and data-centric process management. For each business object involved in a real-world process, a corresponding object exists in the system. In our running example, the Production, Oven, and Workpiece objects have distinct lifecycles.

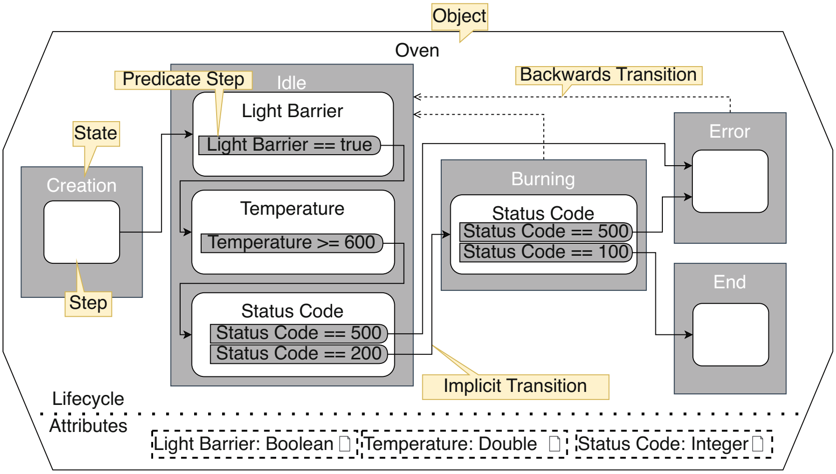


Fig. 3. Object Lifecycle of the Oven object (with annotations)

Conceptual Basis and Oven Object. Figure 3 shows the lifecycle of the Oven object which represents the machine used in the production process. An object encompasses data in the form of attributes and a state-based process model that details the object's lifecycle. This model shows the stages and transitions that the Oven object undergoes. The attributes capture relevant data points, which can be used for internal state comparisons. When an object is instantiated, it automatically enters its initial state (here: *Creation*). As this state only involves an empty step, a direct transition (*Implicit Transition*) to the *Idle* state occurs (represented by a solid arrow). An *Explicit Transition* (represented by dashed arrow) must be triggered explicitly by a user action or an external system [3].

Predicate Steps and Idle State. In the *Idle* state, attributes are set by an automated retrieval of data from the physical machine following the specified sequence of steps (cf. Fig. 3). Once all values are set and the associated conditions, represented by the *Predicate Steps*, are met, a transition to the next state occurs. In the example, a transition to *Burning* takes place if: 1) the Light Barrier is interrupted (indicating the presence of a workpiece in the correct position); 2) the oven has reached the required Temperature; and 3) the oven reports a Status Code of 200 (indicating it is ready for the burning activity). The latter plays a crucial role as it represents the process activity that requires coordination. Here, we see a limitation of PHILharmonicFlows, which was originally designed for working with digital processes [13]. It is capable of retrieving and evaluating data only once and in sequence; not supporting IoT settings that rely on data streams that need to be continuously evaluated to derive higher level states [11].

Backwards Transitions. In the *Error* state, there are two possible scenarios: 1) the error can be automatically resolved or 2) manual error handling is required. Once the error has been resolved, triggering the *Backwards Transition* (dashed backwards arrow) allows the machine to return to the *Idle* state, indicating that it is ready to proceed with normal operations. When the burning process is completed, the oven has two possible transitions, i.e., it may either move to the *End* state or return to the *Idle* state which is triggered by the machine itself. The transition to the end state indicates that the burning task was completed successfully and the lifecycle of the Oven terminated. Alternatively, the oven may transition back to the *Idle* state, signaling that it is ready for further processing.

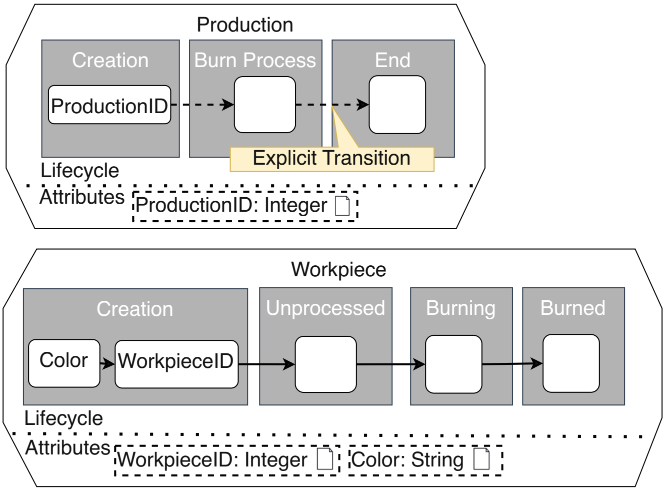


Fig. 4. Object Lifecycles for the Production and Workpiece objects

Production Object. Figure 4 shows the object lifecycles of the Production and Workpiece objects. The Production object represents the high-level production process that involves the Oven and Workpiece objects. For simplicity, in our example, the Production object only has one attribute, the ProductionID. When a Production object is instantiated, it enters the *Creation* state setting the *ProductionID* for that particular production instance. An transition to the *Burn Process* state takes place, which represents the crucial burn activity that has to be coordinated. This explicit transition is triggered either by an external system or by a user action, depending on the level of automation. Upon completion of the Burn process, the Production object transitions similarly to the *End* state indicating that the production process has reached its end.

Workpiece Object. The workpiece object represents the item being processed in the production process. As the workpiece progresses through the production process, it primarily undergoes state changes and transitions based on the actions or states of other objects, such as *Oven* or *Production*. The Workpiece object in this example possesses two attributes: color and ID. The object initiates in the *Creation* state, where the attributes are populated with data. Through an implicit transition, the Workpiece object transitions to the *Unprocessed* state, followed by the *Burning* state, and it ultimately reaches the end state *Burned*.

3.4 Coordination Process for Handling Concurrency

Coordination Processes offer a flexible way for managing interdependent lifecycle processes by expressing semantic constraints through relationships, which are enforced at runtime [13]. This is achieved through a high-level graphical notation. Fig. 5 shows the coordination process for handling concurrent access to the Oven.

Coordination Transitions. Coordination processes utilize the coordination transitions presented in Table 1 [23]. Coordination processes have the capability to either allow or disallow state activations. The actual activation of its states is determined by the object itself based on its lifecycle process. As a result of reaching a particular state in the object, it becomes feasible for a coordination process to allow the activation of this state well in advance, as the transition condition is already met. Conversely, a coordination process may disallow the execution until the specified coordination constraints are met, i.e., the coordination transition becomes enabled. To enable a coordination transition, it becomes necessary for all preceding transitions in the coordination process to have been enabled as well, e.g., the Production object must be in the *Creation* state to instantiate an Oven or a Workpiece object (cf. Fig. 5).

Ports. A coordination step (excluding the start coordination step) may have any number of ingoing *Ports* to which coordination transitions are connected (cf. Fig. 5). All transitions connected to a single port correspond to a logical (AND) conjunction. Multiple ports in a coordination step are considered as logical (OR) disjunctions, i.e., the coordination step is activated when all coordination transitions connected to one port are enabled.

Table 1. Types of supported coordination transitions in PHILharmonicFlows [23]

Name	Description of the coordination transitions
TopDown	The execution of one or more lower level processes is dependent on the status of a single higher level process
BottomUp	The execution of one higher level process depends on the execution status of one or more lower level processes of the same type
Transverse	The execution of one or more processes is dependent on the execution status of one or more processes of a different type. Both types of processes have a common higher level process
Self	The execution of a process depends on the completion of a previous step of the same process
SelfTransverse	The execution of a process depends on the execution of other processes of the same type. All processes have a common higher level process

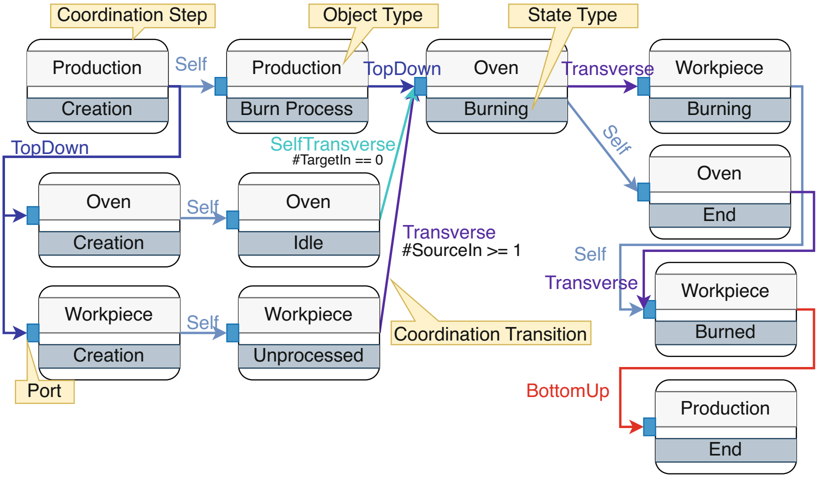


Fig. 5. Coordination Process for the Production, Oven and Workpiece objects

Coordination of Concurrent Activities. The coordination process depicted in Fig. 5 is responsible for managing the processes that access the Oven resource concurrently. Once a Production object becomes instantiated, the oven and workpieces can be instantiated. The 1:1 relationship between the Oven and Production objects (cf. Fig. 2) ensures that only one oven may be instantiated, whereas there is no limit on the number of workpieces. The transition of the Oven from state *Creation* to *Idle* is guided by its lifecycle (cf. Fig. 3). Similarly, the workpiece transitions from *Creation* to *Unprocessed* (cf. Fig. 4). The oven may enter the *Burning* state when the production process explicitly transitions to the *Burn*

Process state triggered by an external signal. At the same time, the oven must be in *Idle* state. This is defined by the *SelfTransverse* transition in combination with condition *#TargetIn == 0*, which ensures that there are no ovens in state *Burning.* Additionally, there must be at least one unprocessed workpiece, as defined by the transition *Transverse* with condition *#SourceIn >= 1.* Once the oven enters the *Burning* state, the corresponding workpiece transitions to the *Burning* state as well. After completing the burning process, the oven transitions to the *End* state, and the workpiece transitions to the *Burned* state. Consequently, the coordination process for the production terminates (cf. Fig. 5). In the manufacturing scenario, the coordination transitions are not fulfilled if the oven is occupied and another process seeks access. This request is then queued (e.g., in a FIFO queue). The queued requests are activated in new instances of the coordination process once the oven is idle again (cf. Backwards Transition in Fig. 3) and there is at least one unprocessed workpiece.

4 Related Work and Discussion

4.1 Related Work

First, we review related work that deals with concurrency handling in general: [6] presents a software architecture for IoT applications that use *accessors* as proxies for IoT devices and a coordination framework to handle concurrency. However, the focus of this work is on timestamps and sequential execution rather than on coordinating different object instances. [5] analyzes the challenges of coordinating concurrent IoT-aware processes, exploring complexities related to resource sharing, synchronization, and scalability. While it provides valuable insights into managing concurrent processes, it falls short in providing solutions to address these challenges. [28] presents a Petri net based method for scheduling resources and controlling IoT-aware processes. This approach allows analyzing and optimizing of complex shop floor processes. However, this approach lacks scalability and might be accompanied with a suboptimal resource utilization and reduced responsiveness to changes in respect to resource availability.

Second, we review related approaches from the BPM field: [17] introduces a framework for enabling adaptive, cyber-physical processes that involve physical entities and IoT devices. While the framework allows for monitoring, exception detection, and automated resolution strategies, it emphasizes process adaptation based on IoT data, rather than activity coordination. [14] introduces semantic web services to achieve easily configurable and flexible manufacturing systems. The proposed approach uses pre- and post-conditions for verifying service requests, which may be also used to handle and queue concurrent requests [21]. [24] proposes a method for synchronizing manufacturing processes based on BPMN 2.0, with a focus on optimizing production efficiency through buffering and unbundling operations. This approach dynamically adjusts process synchronization according to the available real-time data. As a drawback, however, only modeling patterns in BPMN 2.0 are used, and without these recipes, it is not exactly clear how process instances are coordinated [24]. *Batch activities* involving the same activities to be executed in different processes that lead to concurrency are discussed in [18]. The authors propose a modeling and execution approach for batch activities in BPMN-based business processes. As this problem resembles the concurrency issues addressed in our work, we adapt similar concepts but put a stronger focus on the data-driven lifecycle models and processes of objects involved in the executions. [15] introduces an approach for synchronizing process activities in workflow systems, enabling their dynamic coordination based on pre-specified synchronization rules [15]. While the approach enhances process control and sequencing, it lacks a comprehensive implementation of synchronization patterns. [8] categorizes various types of instance-spanning constraints (e.g., for coordination) and introduces a formal notation to represent and enforce these constraints within process models. A shortcoming is the lack of assurance that future instances will not breach inter-scenario constraints currently satisfied.

4.2 Discussion

With our work, we effectively address the lack of support for managing concurrency in IoT-aware processes [10]. IoT environments include numerous devices, objects, materials, and resources that are involved in executing IoT-aware processes. Existing approaches fail to express the complex relations and constraints among these entities to achieve the necessary coordination while maintaining flexibility. This paper discusses an answer to the research question on how to coordinate concurrent process activities in IoT environments with constrained physical resources (cf. Sect. 2). By proposing an approach based on object-centric processes in PHILharmonicFlows, we demonstrate how to model and manage concurrency in IoT-aware processes (cf. Sect. 3). The relational process type structure (cf. Sect. 3.2) aids in coordinating concurrent process activities by incorporating the relationships and interactions between objects. This hierarchy of objects and their relationships are crucial for managing access to shared physical resources in IoT. The framework allows modeling and managing object lifecycles (cf. Sect. 3.3) and thus enables the coordination of activities that access the same physical resource by enforcing semantic relationships and coordination constraints (cf. Sect. 3.4). One notable benefit of using PHILharmonicFlows is the inclusion of a runtime engine capable of executing the lifecycle models [3].

This work is focused on a specific use case from smart production. While this allowed us to illustrate basic concepts, it limits the generalizability of our findings. Note that the running example constitutes a simplified manufacturing process that handles only a single workpiece at the time. However, in real-world production environments, machines may exist multiple times and be capable of multi-tasking and batch processing to increase efficiency. While PHILharmonicFlows can be also used to model these capabilities, a potential disadvantage could be the quantity and complexity of the models required to represent these objects and coordination processes. This inherently necessitates that users acquire proficiency in the specific modeling language of PHILharmonicFlows.

5 Summary and Outlook

Process activities whose execution involve IoT devices need to be coordinated as the number of available devices and their multi-tasking capabilities might be limited. Traditional BPMS often rely on capabilities of external systems or additional components to handle concurrent activity executions, which are inflexible and incomprehensible for the user. This paper addresses the problem of coordinating concurrent process activities in IoT. We propose an object-centric approach based on the PHILharmonicFlows framework to handle concurrency in IoT-aware processes. PHILharmonicFlows allows coordinating objects which fosters concurrency control for IoT-aware processes as IoT devices can be represented as objects. The solution uses a relational process type structure to capture the relationships between objects in a hierarchical view. Object lifecycles and coordination processes are modeled and managed within the framework, allowing activities that access the same physical resource to be coordinated. We demonstrated the framework in a smart manufacturing scenario. The proposed solution addresses the challenges of coordinating activities that involve constrained physical resources, providing a flexible and comprehensible coordination mechanism.

In future research, we will extend PhilharmonicFlows beyond its current capacity for defining constraints and for handling concurrency towards facilitating dynamic and ad-hoc adaptations of IoT-aware processes. This will include additional capabilities for handling data streams and achieving real-time scheduling and work distribution with the goal of responding to immediate changes, demands, and errors within the IoT landscape.

References

1. van der Aalst, W.M.: Business process management as the "killer app" for petri nets. Softw. Syst. Model. **14**(2), 685–691 (2015)
2. van der Aalst, W.M., Ter Hofstede, A.H., Kiepuszewski, B., Barros, A.P.: Workflow patterns. Distrib. Parallel Databases **14**, 5–51 (2003)
3. Andrews, K., Steinau, S., Reichert, M.: Enabling runtime flexibility in data-centric and data-driven process execution engines. Inf. Syst. **101**, 101447 (2021)
4. Arnold, L., Breitmayer, M., Reichert, M.: Monitoring object-centric business processes: an empirical study. In: Nurcan, S., Opdahl, A.L., Mouratidis, H., Tsohou, A. (eds.) RCIS 2023, pp. 327–342. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-33080-3_20
5. Bicevskis, J., Karnitis, G., Bicevska, Z., Oditis, I.: Analysis of concurrent processes in internet of things solutions. In: Ziemba, E., Chmielarz, W. (eds.) FedCSIS-AIST/ISM -2021. LNBIP, vol. 442, pp. 26–41. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98997-2_2
6. Brooks, C., et al.: A component architecture for the internet of things. Proc. IEEE **106**(9), 1527–1542 (2018)
7. Derler, P., Lee, E.A., Sangiovanni Vincentelli, A.: Modeling cyber-physical systems. Proc. IEEE **100**(1), 13–28 (2012)
8. Fdhila, W., Gall, M., Rinderle-Ma, S., Mangler, J., Indiono, C.: Classification and formalization of instance-spanning constraints in process-driven applications. In:

La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 348–364. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_20

9. Guabtni, A., Charoy, F., Godart, C.: Concurrency management in transactional web services coordination. In: Bressan, S., Küng, J., Wagner, R. (eds.) DEXA 2006. LNCS, vol. 4080, pp. 592–601. Springer, Heidelberg (2006). https://doi.org/10.1007/11827405_58
10. Janiesch, C., et al.: The internet of things meets business process management: a manifesto. IEEE Syst. Man Cybern. Mag. **6**(4), 34–44 (2020)
11. Kirikkayis, Y., Gallik, F., Seiger, R., Reichert, M.: Integrating IoT-driven events into business processes. In: Cabanillas, C., Pérez, F. (eds.) CAiSE 2023, pp. 86–94. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-34674-3_11
12. Kirikkayis, Y., Gallik, F., Winter, M., Reichert, M.: BPMNE4IoT: a framework for modeling, executing and monitoring IoT-driven processes. Future Internet **15**(3), 90 (2023)
13. Künzle, V., Reichert, M.: Philharmonicflows: towards a framework for object-aware process management. J. Softw. Maint. Evol. Res. Pract. **23**(4), 205–244 (2011)
14. Malburg, L., Klein, P., Bergmann, R.: Using semantic web services for AI-based research in industry 4.0. CoRR abs/2007.03580 (2020)
15. Mangler, J., Rinderle-Ma, S.: Rule-based synchronization of process activities. In: IEEE 13th Conference on Commerce and Enterprise Computing, pp. 121–128 (2011)
16. Mangler, J., et al.: Datastream XES extension: embedding IoT sensor data into extensible event stream logs. Future Internet **15**(3), 109 (2023)
17. Marrella, A., Mecella, M., Sardina, S.: Supporting adaptiveness of cyber-physical processes through action-based formalisms. AI Commun. **31**, 47–74 (2018)
18. Pufahl, L., Weske, M.: Batch activity: enhancing business process modeling and enactment with batch processing. Computing **101**(12), 1909–1933 (2019)
19. Schulte, S., Janiesch, C., Venugopal, S., Weber, I., Hoenisch, P.: Elastic business process management: state of the art and open challenges for BPM in the cloud. Futur. Gener. Comput. Syst. **46**, 36–50 (2015)
20. Seiger, R., Herrmann, S., Aßmann, U.: Self-healing for distributed workflows in the internet of things. In: 2017 IEEE ICSAW 2017, pp. 72–79 (2017)
21. Seiger, R., Malburg, L., Weber, B., Bergmann, R.: Integrating process management and event processing in smart factories: a systems architecture and use cases. J. Manuf. Syst. **63**, 575–592 (2022)
22. Steinau, S., Andrews, K., Reichert, M.: The relational process structure. In: Krogstie, J., Reijers, H.A. (eds.) CAiSE 2018. LNCS, vol. 10816, pp. 53–67. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91563-0_4
23. Steinau, S., Andrews, K., Reichert, M.: Enacting coordination processes. arXiv preprint arXiv:2012.08409 (2020)
24. Traganos, K., Spijkers, D., Grefen, P., Vanderfeesten, I.: Dynamic process synchronization using BPMN 2.0 to support buffering and (un)bundling in manufacturing. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNBIP, vol. 392, pp. 18–34. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58638-6_2
25. Valckenaers, P., Van Brussel, H., Verstraete, P., Saint Germain, B.: Hadeli: schedule execution in autonomic manufacturing execution systems. J. Manuf. Syst. **26**(2), 75–84 (2007)
26. Vanderfeesten, I., et al.: Developing process execution support for high-tech manufacturing processes. In: Empirical Studies on the Development of Executable Business Processes, pp. 113–142 (2019)

27. Vasilecas, O., Smaizys, A., Rima, A.: Business process modelling and simulation: hybrid method for concurrency aspect modelling. Baltic J. Mod. Comput. **1**(3–4), 228–243 (2013)
28. Wu, X., Tian, S., Zhang, L.: The internet of things enabled shop floor scheduling and process control method based on petri nets. IEEE Access **7**, 432–442 (2019)

Viola: Detecting Violations of Behaviors from Streams of Sensor Data

Gemma Di Federico(✉), Giovanni Meroni, and Andrea Burattin

Technical University of Denmark, Kgs. Lyngby, Denmark
`gdfe@dtu.dk`

Abstract. Sensor networks and the Internet of Things enable the easy collection of environmental data. With this data it is possible to perceive the activities carried out in an environment. For example, in healthcare, sensor data could be used to identify and monitor the daily routine of people with dementia. In fact, changes in routines could be a symptom of the worsening of the disease. Streaming conformance checking techniques aim at identifying in real-time, from a stream of events, whether the observed behavior differs from the expected one. However, they require a stream of activities, not sensor data. The artifact-driven process monitoring approach combines the structure of the control-flow with the data in an E-GSM model. This paper presents VIOLA, the first technique capable of automatically mining an E-GSM model from a labeled sensor data log, which is then suitable for runtime monitoring from an unlabeled sensor stream to accomplish our goal (i.e., streaming conformance checking). This approach is implemented and has been validated with synthetic sensor data and a real-world example.

1 Introduction

The environment we live in is becoming smarter and interconnected thanks to the increasing use of sensor systems. These systems are able to collect large amount of data on the environment we are living in and could be used to perceive the activities performed in order to derive their dependencies. In particular, these data can be represented in the form of a process model, where the process is intended as the execution of a series of activities performed in the environment.

One field, among many, that would benefit from this technology is the healthcare domain. In the area of Ambient Assisted Living, sensor systems can be used to discover and monitor the course of a disease. For instance, people affected by dementia usually tend to follow a strict routine. A deviation in the routine could be a symptom of the worsening of the disease. Therefore, by automatically modeling a person's routines as a process model [5], it becomes possible to monitor whether deviations from daily habits are occurring, in order to monitor the evolution of their disease.

However, data captured by sensor systems and behavioral process models are at two different levels of abstraction. Sensor data represent low-level events at a specific point in time, whereas behavioral process models represent high-level

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 118–130, 2024.
https://doi.org/10.1007/978-3-031-50974-2_10

activities and their dependencies. In addition, the application scenario requires real-time analysis so that medical staff can be notified as soon as changes are detected and react accordingly. This makes post-mortem analysis trivial. Therefore, discovering a behavioral process model from sensor data, detecting in real-time when activities in the model are executed, and identifying deviations between the discovered model and the observed behavior as they occur becomes a challenging task.

To address this issue, streaming conformance checking [2,15] aims to analyze an event stream with respect to a process model. However, most approaches for conformance checking focus on the structure of the process (i.e. the control flow), ignoring the data generated when the process is run (e.g. sensor data). Hence, they expect the stream to contain high-level events, rather than low-level sensor data. Therefore, a pre-processing step in charge of abstracting activities from sensor data would be necessary. Artifact-driven process monitoring is one of the few approaches that explicitly combines the structure of a process with the data produced by the activities in that process [10]. An example is the Extended-GSM (E-GSM) modeling language, which includes both control flow dependencies and rules to determine when activities are executed. However, artifact-driven process monitoring requires the process and the rules to identify activities to be manually modeled, either from scratch or from an imperative process model [11].

The work presented in this paper is called VIOLA, and it aims at detecting violations of behaviors from a stream of sensor data. The approach relies on Process Mining and Machine Learning techniques to automatically generate an E-GSM model from a labeled sensor log, where the labels indicate high-level activities carried out and perceived (i.e., observed) by the sensors. The streaming conformance checking algorithm takes as input directly the stream of raw sensors measurements, which lies at a lower abstraction level. The paper aims to answer the following research questions. First, we would like to focus on the derivation of the E-GSM process model, verifying if the approach is able to construct a behavioral process model suitable for conformance checking, where the data are in the form of labeled activities (**RQ1**). The resulting model is then used by the engine to monitor the process at run-time. In particular, the engine receives a stream of unlabeled sensor data and uses the E-GSM model to detect which activities are running. The second research question focuses on recognition, so we want to assess the approach's ability to recognize activities from an unlabeled stream of sensor data (**RQ2**). Once the running activities have been identified, the engine verifies the conformance between the observed behavior and its representation in the model. Therefore, the approach should be capable of verifying if the observed behavior is consistent with the discovered behavioral model (**RQ3**). The approach has been validated with an artificial sensor log and a real-world one.

The rest of the paper is structured as follows. Section 2 presents the related work. The approach is explained in Sect. 3. It is then evaluated in Sect. 4, including a discussion of the results. Section 5 concludes the paper.

2 Related Work

The level of abstraction at which a process model is represented is reflected in the granularity of the event log used to derive the model. When the process is captured by sensor systems, the data is in the form of fine-grained sensor measurements, which result in a chaotic and non-expressive process model [14]. Instead, experts are interested in analyzing the behavior in terms of higher-level activities. Hence, sensor measurements need to be grouped into activity labels.

Several techniques to cope with this problem have been devised, ranging from event abstraction [16] to activity recognition [1]. One of the main challenges of this work is to recognize activities on a stream of data, i.e. online. In the field of event abstraction, Tax et al. [13] propose a supervised learning approach that applies a statistical modeling method to recognize activities. The approach takes as input a set of annotated traces used to train a conditional random field, which then operates on an unlabeled log to abstract activities. Moving to the activity recognition field, Sanchez et al. [12] propose a technique to recognize abnormal activities. They use a labeled event log to train a classifier, which is then tested on an unlabeled log. The approach is only capable to detect activities that were already included in the training set. Also Maswadi et al. [9] make use of classification algorithms, comparing Decision Tree (DT) and Naïve Bayes classifiers with the aim to perform a real-time classification of activities. The performance of the two algorithms is similar, but the DT classifier resulted in higher accuracy. The authors also highlight the importance of data pre-processing for feature selection, which can significantly improve classification accuracy. What is more, the DT is more clear in the explanation and representation of the decisions.

Furthermore, the process model must be able to represent activities but also consider the information behind them. That is, make the process data-aware. For the imperative languages, the focus is on data-aware Petri Nets [7,8]. E.g., the Colored Petri Net, where the data dimension is dependent on the control flow as data elements are attached to the tokens. During the verification of a Colored Petri Net [4], both the data and the control flow are checked to evaluate the next activity. When a violation is identified, the verification of the process instance is stopped. Moving towards declarative languages, Burattin et al. [3] recognize the need of verifying the conformance of multi-perspective declarative models, by proposing a conformance checking algorithm that exploits also the data dimension. It is worth mentioning that all of the aforementioned approaches revolve around the notion of *activity*, and the data is seen as an event attribute. In the context of this paper, the objective is to deal directly with a stream of data values, where the notion of activity naturally emerges from the data configuration observed. On the one hand, all the mentioned approaches do not allow for the streaming verification of the process. On the other hand, the existing online conformance checking algorithms [2,15] only focus on the control flow without considering the data. A solution is presented by Meroni et al. [11], E-GSM, in which both the data and the process perspectives are considered, by introducing the so-called process flow guards. The approach allows for the runtime verification of a process. However, no approach to discover E-GSM models

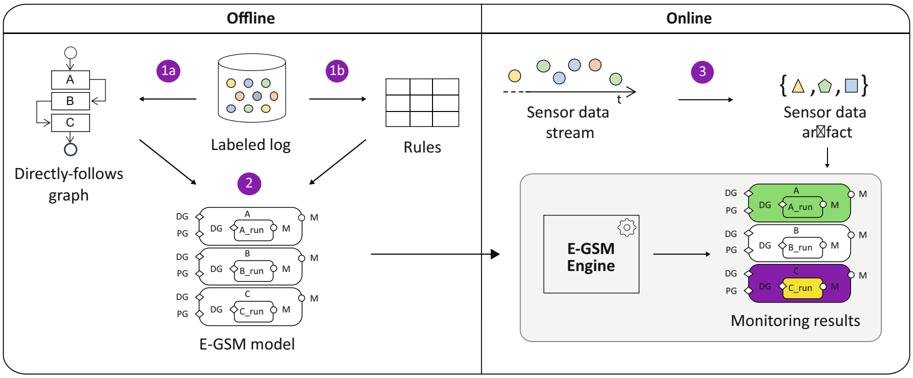


Fig. 1. Overview of the VIOLA approach. The different steps of the approach are numbered 1a, 1b, 2, and 3

from execution logs exists. Also, to determine when activities are executed, the E-GSM model predicates on labels representing the discrete state of the environment, rather than on raw sensor data.

3 Approach

The approach presented in this paper, named VIOLA, aims to identify violations of behaviors from a stream of sensor data. The streaming conformance checking, based on an E-GSM model, is able to process a stream of sensors, represent it in the form of an abstracted process model, and verify at run-time if new instances of the process are compliant.

An overview of the VIOLA approach is presented in Fig. 1. VIOLA is divided into two phases, one offline and one online. During the offline phase, the E-GSM model is constructed. The starting point is a labeled sensor log. *Step 1a* consists of the derivation of the process flow conditions, in the form of a Directly-Follows Graph (DFG). In *Step 1b*, the data flow constraints are derived and represented in the form of rules. The rules explain how each activity label can be recognized, starting from an unlabeled series of sensor measurements. In *step 2*, both the DFG and the rules are used to build the E-GSM model. In particular, information in the DFG is used to derive the structure of the process, whereas the rules are used to derive the conditions on the *data flow guards* and *milestones*. In the online (i.e., streaming) phase, a stream of unlabeled sensor data is processed (*step 3* of Fig. 1). The processing consists in dividing the stream in windows to be analyzed and organizing them in the sensor data artifact suitable for the E-GSM engine. Based on the content of the windows and the constraints on the stages, running activities are recognized. If the rules on the guards are not fulfilled, violations in the execution of the process are identified.

In the following sections, each step of the two phases is explained in detail. But before, background knowledge on E-GSM is provided.

3.1 Background

E-GSM is an extension of the Guard-Stage-Milestone (GSM) artifact-centric language explicitly designed for process monitoring. As shown in Fig. 3, E-GSM represents the units of work performed in a process (e.g., activities and process portions) with stages. Stages can be either atomic (e.g., A_run), or can nest other stages (e.g., A).

To determine when a stage is executed, it is annotated with data flow guards (e.g. $A.DG$ and $A\_run.DG$), and milestones (e.g., $A.M$ and $A\_run.M$). Data flow guards and milestones are Event-Condition-Action (ECA) rules, which require an event and a boolean expression to be specified. In particular, the boolean expression in an ECA rule is evaluated when the specified event happens, based on the state of the model at that time.

Stages can be in one of the following states: *unopened*, *opened*, *closed*. When the process starts, all stages are *unopened*, indicating that they were never executed. When a data flow guard holds and the associated stage is *unopened* or *closed*, that stage becomes *opened*, indicating that the process element represented by that stage is being executed. When a milestone holds and the associated stage is *opened*, that stage becomes *closed*, indicating that the process element completed its execution. Also, if that stage has any *opened* child stages, those become *closed* too.

A stage can be decorated with process flow guard (e.g., $A.PG$) to represent control flow dependencies that a stage should fulfill before being executed. A process flow guard is a boolean expression that is evaluated immediately after one of the data flow guards of the associated stage holds.

Stages can also be marked as *onTime*, *outOfOrder*, or *skipped*. When the process starts, all stages are *onTime*. When one of the data flow guards of a stage holds, the expression on the process flow guard of that stage is evaluated. If the expression evaluates to `true`, the associated stage remains *onTime*, *outOfOrder* otherwise. In addition, if the expression predicates on another stage being *opened* or *closed*, and that stage is *unopened*, then that stage becomes *skipped*.

It is worth noting that process flow guards do not enforce control flow dependencies. A stage can still be opened even if its process flow guard does not hold, as long as at least one of its data flow guards holds. This allows to continuously and autonomously monitor the execution of a process based on sensor data.

E-GSM supports different kinds of events. In this paper, we will cover only a subset that is relevant for VIOLA. When the E-GSM engine receives updated sensor data, it triggers two events e' and e'' to be emitted in succession. These events can then be used by data flow guards and milestones (e.g., $A.DG$ and $A\_run.M$) to check their boolean expression only when the sensor data changes. Also, when a milestone holds, it triggers an event (e.g., $A\_run.M$ when the milestone M of A_run is achieved) to be emitted. Such event can then be used by data flow guards and milestones (e.g., $A.M$) to check their value only when a portion of the process completed its execution.

Boolean expressions can predicate both on sensor data to check whether they assume a specific value (e.g., $A.DG$ and $A\_run.M$) and on the model itself to check whether a stage is *opened* (e.g., $A.PG$ and $A.M$).

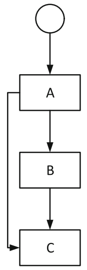
3.2 Steps 1a and 1b: Derivation of DFG and Decision Rules

The key aspect of the proposed approach is its capability of processing an unlabeled stream of data. To achieve the objective, an E-GSM model must be derived. For the automatic generation of the model, information related to the control flow and to the activities must be provided. As depicted in Fig. 1, the first phase is composed of two steps. In *step 1a* the control flow is discovered by means of a Process Mining algorithm, that derives a DFG. In the DFG, the nodes represent activities while the edges are the directly follows relationships between pairs of activities. We also assume that it is possible to augment the DFG with information regarding which nodes are starting nodes. This information can be extracted either from the DFG itself (e.g., by looking for nodes with no incoming edges) or by looking for activities that are predominantly observed as the first activity in the log (the actual procedure to perform this is outside the scope of this paper). The DFG is used to define, in the E-GSM model, the stages and their process flow guards. Since the E-GSM model is derived starting from the DFG, the expressiveness of the E-GSM model is fundamentally limited by the expressiveness of DFG. In particular, parallel executions are not allowed.

The stage names on the E-GSM model have the label of the high-level activities from the labeled sensor log. However, during the streaming conformance checking phase, an unlabeled stream is processed. For this reason, we have to derive the rules necessary to recognize an activity (*step 1b*), that will be used to define the data flow guards and milestones in the E-GSM model. We require that the same activity label cannot be repeated in a sequence, since ideally the same activity is not performed several times in a row. What is more, we assume that there is no interleaving in the execution of activities. An activity is characterized by a specific series of sensor data, and this information is used to train a classifier. For example, we can segment the sensor log into windows and, for each window, determine the set of sensors required to detect the corresponding activity label. Sensor names could then be used as features to train a DT classifier, where the leaves (i.e., the classes) refer to the activity labels, while the internal nodes predicate on the selected classification features. The classifier is translated into first-order logic expressions, each referring to the data conditions for a certain activity to occur.

3.3 Step 2: Generation of E-GSM Models from DFGs and Decision Rules

Once we have discovered the DFG and we have identified the rules to infer from sensor data when an activity is running, we can build an E-GSM model (*step 2* in Fig. 1) that will be used to monitor the process at runtime (i.e., streaming conformance checking).



(a) DFG

Label	Rule
A	x>1
B	y>2
C	z>1

(b) Rules

Fig. 2. DFG (a) and rules (b) extracted from labeled sensor log

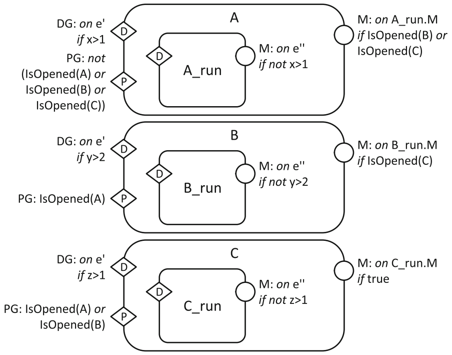


Fig. 3. Generated E-GSM model

Since an activity in a DFG can be connected with any other activity, existing techniques to generate E-GSM models from block-structured process models are not suited for this purpose. Also, approaches that enforce control flow dependencies cannot be used, since they would not allow monitoring executions that differ from the expected control flow. Therefore, we defined an entirely new transformation logic to derive the E-GSM model.

In particular, for each activity T in the DFG, the following elements are derived in the E-GSM model.

- **One atomic stage** S_run, which is used to detect when T is running.

  For example, stages A_run, B_run and C_run in Fig. 3 are derived from, respectively activities A, B and C in Fig. 2a.

- **One parent stage** S, which contains S_run and it is used to assess the conformance of T with respect to the expected execution flow.

  For example, stages A, B and C in Fig. 3 are derived from, respectively activities A, B and C in Fig. 2a.

- **One data flow guard** $S.DG$, which is attached to S. $S.DG$ is evaluated when event e' is emitted (i.e., every time updated sensor data are received, before e'' is emitted). $S.DG$ requires the rules from the decision tree associated with the class T to evaluate to `true`. In this way, whenever updated sensor data are received, $S.DG$ is evaluated. If the sensor data indicate that activity T is being executed (that is, $S.DG$ evaluates to `true`), stage S will become *opened*.

  For example, the data flow guard $A.DG$ in Fig. 3 requires the rule in Fig. 2b associated with activity A, that is, the value registered by sensor x to be greater than 1, to evaluate to `true`.

- **One data flow guard** $S\_run.DG$, which is attached to S_run, which is identical to $S.DG$.

> In this way, when activity A is being executed, stage A_run will be *opened*.

- **One milestone** $S\_run.M$, which is attached to S_run. $S\_run.M$ is evaluated when event e'' is emitted (i.e., every time updated sensor data are received, immediately after e' is emitted). $S\_run.M$ requires the rule associated to T to evaluate to `false`. In this way, whenever updated sensor data are received, $S.M$ is evaluated. If the sensor data indicate that activity T is no longer being executed (that is, $S.DG$ evaluates to `false`), stage S will become *closed*. Also, since e'' is always emitted after e', whenever updated sensor data are received, $S\_run.M$ will always be evaluated after all the data flow guards in the model.

> For example, the milestone $A\_run.M$ in Fig. 3 requires the fulfillment of the rule in Fig. 2b associated with activity A, that is, the value registered by sensor x to be greater than 1, to evaluate to `false`.

- **One milestone** $S.M$, which is attached to S. $S.M$ is evaluated when $S\_run.M$ is emitted. $S\_run.M$ requires at least one stage S_{succ} to be *opened*. S_{succ} is a parent stage derived from an activity T_{succ} which is a successor of T, that is, in the DFG T_{succ} is directly connected to T through an incoming arc. If no stage S_{succ} exists, then $S.M$ always evaluates to `true`.

> For example, the milestone $A.M$ in Fig. 3 is evaluated when $A\_run.M$ is emitted, and requires either stage B or C, which are derived from the activities in Fig. 2a that are successors of A, to be *opened*. In this way, when activity A stops being executed, stage A will become *closed* only if the next activity is either B or C.

- **One process flow guard** $S.PG$, which is attached to S. $S.PG$ requires at least one of the following boolean expressions to evaluate to `true`:
  1. At least one stage S_{pred} must be *opened*. S_{pred} is a parent stage derived from an activity T_{pred} which is a predecessor of T, that is, in the DFG T_{pred} is directly connected to T through an outgoing arc. If no stage S_{pred} exists, then this expression always evaluates to `false`.
  2. If T is an initial activity, that is, in the DFG the process start is directly connected to T with an outgoing arc, no parent stage in the model must be *opened*. If T is not an initial activity, then this expression always evaluates to `false`.

> For example, the process flow guard $A.PG$ in Fig. 3 requires that stages A, B and C must not be *opened*, since activity A in Fig. 2a is an initial activity and has no predecessor. Conversely, the process flow guard $B.PG$ in Fig. 3 requires that stage A must be *opened*, since activity A in Fig. 2a is a predecessor of activity B.

3.4 Step 3: Streaming Processing and Conformance Checking

The last step of the approach, *step 3* in Fig. 1, consists of the conformance checking of a stream of sensor data. During the online processing, the stream of sensor data passes through a pre-processing phase where the data are converted into the format suitable for the E-GSM engine. As introduced in Sect. 3.2 for the *step 1b*, the sensor data stream can be processed in windows. During the segmentation, we determine the set of sensors contained in each window, which is transformed into a data object (i.e. the sensor data artifact), and sent to the E-GSM engine. Each data object is evaluated by the engine according to the data flow guards and milestones, in order to check whether the target conditions are met, that is, if the activity is recognized and running.

To conclude, the approach presented in this paper allows the discovery of a E-GSM model starting from a labeled sensor log. The model is then suitable for the streaming conformance checking of an unlabeled sensor stream, in order to detect deviations in the behaviors.

4 Evaluation

To evaluate the approach presented in this paper, two experiments are conducted, one based on synthetic event logs, and the other based on a real dataset. The objective of the evaluation is to verify whether the approach is able to answer the research questions presented in Sect. 1. In particular, the use of synthetic event logs allowed us to conduct a controlled experiment where we could scrupulously verify the approach's ability to process a stream of events, correctly recognize activities, and verify the conformity of the behavioral process model. The experiment using a real dataset served to verify the actual applicability of the approach to a real use case. In the context of this paper, only the first experiment is explained, while the second one can be found in a technical report[1].

The first experiment is based on two synthetic datasets we constructed[2]. The datasets describe the movements of a person inside a smart environment, where movements are captured by sensors. To obtain the datasets, we used a smart environment simulator tool, called Linac [6], which produces as output a stream of triggered sensors, generated by the movements of a previously programmed agent. The stream has been sequentialized and stored in a single log file and it has been processed in order to be labeled. Two scenarios are constructed: a normal scenario used to discover the process model and construct the E-GSM, and a variation (i.e. a misalignment) used in the online phase for streaming conformance checking.

The map of the environment used in the simulation is composed of four rooms (named *A, B, C, D*). Each room is equipped with 5 pressure sensors on the floor that record pressure variations, and a presence sensor on the entry door that triggers at the passage in the detecting area. Sensors have a trigger frequency of 30 s. The person moves across the rooms, and the behavior changes according to the situation being simulated. In the base scenario (named l-base), the person moves between the four rooms, following the order from *A* to *D*.

---

[1] The appendix can be found at https://dx.doi.org/10.5281/zenodo.7982452.

[2] The source code can be found at https://github.com/gemmadifederico/VIOLA.

The log constitutes the base case, and it is used to verify the capability of the approach in detecting activities. In the variation of the scenario (named *l-error*), the path of the person varies, covering either all the rooms in a different order or just some of them. The variation is in the control flow, and the goal is to recognize the set of activities executed but also identify when the set differs from the modeled one, i.e. the violations.

The base behavioral model is derived from the normal scenario. The l-base log is split into training and test sets (70/30). Following the procedure indicated in Fig. 1, a DFG and a set of rules are derived. In our tests, we used the PM4Py and Scikit-learn libraries for Python[3] for deriving, respectively, the DFG and the set of rules. Firstly, the training set is pre-processed to group events into timed windows of 30 s. The size of the window was decided based on the average duration of each activity identifier, trying to approximate it. For each window, we extract the set of sensors and the corresponding activity label. The sensor names are used as features to train a classifier. The CART (Classification and Regression Tree) algorithm is used for the classification, while the Gini Index is adopted for the splitting. The obtained DT is then translated into boolean expressions to obtain the rules. Secondly, the corresponding function of the aforementioned library is used to derive a DFG representing the control flow of the process. Therefore, the DFG and the rules are translated in the E-GSM behavioral model, which is passed to the engine for the online phase. SMARTifact [10] is the process monitoring platform used by the engine. During the online phase, the logs are pre-processed by a Complex Event Processing (CEP) system, which simulates a stream, hides the label, and groups the data in timed windows (as in offline processing).

Table 1. Results of the synthetic simulation

	l-base	l-error
False positives	0	8
False negatives	0	1
True positives	400	235
Precision	1	0.97
Recall	1	0.99
F1 score	1	0.98

(a) Activity recognition evaluation

	l-base	l-error
onTime activities	400	50
outOfOrder act.	0	185
Tot. activities	400	236
Conformance	1	0.21

(b) Control flow evaluation

The results of the application of the approach are presented in Table 1. The evaluation covers the recognition of activities considering false positives, false negatives, and true positives (Table 1a); while the control flow is evaluated by considering onTime and outOfOrder activities, as well as the conformance measure (Table 1b). The first columns in Tables 1 refer to the verification with the

---

[3] See https://pm4py.fit.fraunhofer.de/, https://scikit-learn.org/stable/.

test set of the normal scenario. As can be noticed in Table 1a, all the activities are correctly recognized, returning a perfect value for precision, recall, and F1 score. When the control flow varies, that is `l-error`, it becomes more challenging to recognize the activities. However, the approach was able to recognize almost the entire set of activities (indeed precision and recall values are still high), starting from the sensor stream. Once the activities have been recognized, we need to verify if they conform to the control flow. An activity identified among the true positives is flagged as *onTime* when it fulfills the requirements of both the data and the process flow guards. If an activity is detected, and the data flow guard holds but the process flow guard doesn't hold, the activity is marked as *outOfOrder*. Therefore, the overall conformance is computed as the ratio between the total amount of *onTime* activities and the sum between the total *onTime* and the total *outOfOrder*. The results are shown in Table 1b. In the verification using the test set of the normal scenario, the control flow perfectly fits the reference model, as expected. For the variation instead, it is important to consider that not all the activities in `l-error` are non-compliant. In fact, there could be a partial match of the control flow. As a consequence, not all the activities are marked as *outOfOrder*.

The application of the approach on synthetic datasets produced promising results as the discovered data guards are capable of discriminating the different activities. Hence, we are able to construct a correct and meaningful behavioral model suitable for conformance checking. Therefore, we can positively answer to RQ1. In the base case, all the activities were correctly recognized. Also, no control flow violations were detected. Even with the introduction of variations in the behaviors, only a few activities were erroneously classified. Similarly, only a few non-existent control flow violations were detected. This aspect partially answers to RQ2 and RQ3, since it could generate a cascade effect. If an activity is mislabeled and therefore it is flagged as *running*, but violates the control flow, it will impact the conformance of the entire process instance. In other words: if an activity is wrongly classified, then the conformance will be evaluated against the classified activity, not the real one.

5 Conclusions and Future Work

In this paper, we presented VIOLA, a behavioral violations detection approach that works on streams of sensor data. VIOLA is able to automatically derive an E-GSM behavioral model starting from a labeled sensor log. What is more, VIOLA allows the online verification of the conformity between the behavioral process model and an unlabeled stream of sensor data. The behavioral model represents activity labels as stages, while the stages predicate on the sensor data. In other words, the approach is able to process a stream of raw sensor events (i.e. low-level of granularity), while the model represents higher-level activity labels. The approach is evaluated using synthetic and real datasets. In both cases positive results were obtained, supporting the quality of the proposed approach. The approach has some limitation mainly due to the expressiveness of the DFG graph used to generate the E-GSM model. The approach does not allow for parallel executions. What is more, we do not consider repetitive activities, i.e., when the same activity is repeated several times in a sequence. In fact, we assume that a person does not perform the same activity over and over again, but only that they are performing that specific activity. As a future work, we would like to improve the detection of activities by enhancing the hyperparameter tuning process, as it influences the performance of the task. In addition, we aim to formalize the translation from the DFG and the rules in the E-GSM model, in order to demonstrate their correctness.

References

1. Arshad, M.H., Bilal, M., Gani, A.: Human activity recognition: review, taxonomy and open challenges. Sensors **22**(17), 6463 (2022)
2. Burattin, A., Carmona, J.: A framework for online conformance checking. In: Teniente, E., Weidlich, M. (eds.) BPM 2017. LNBIP, vol. 308, pp. 165–177. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74030-0_12
3. Burattin, A., Maggi, F.M., Sperduti, A.: Conformance checking based on multi-perspective declarative process models. Expert Syst. Appl. **65**, 194–211 (2016)
4. Carrasquel, J.C., Mecheraoui, K., Lomazova, I.A.: Checking conformance between colored petri nets and event logs. In: van der Aalst, W.M.P., et al. (eds.) AIST 2020. LNCS, vol. 12602, pp. 435–452. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72610-2_33
5. Di Federico, G., Burattin, A., Montali, M.: Human behavior as a process model: which language to use? In: ITBPM@ BPM, pp. 18–25 (2021)
6. Di Federico, G., Nikolajsen, E.R., Azam, M., Burattin, A.: Linac: a smart environment simulator of human activities. In: Munoz-Gama, J., Lu, X. (eds.) ICPM 2021. LNBIP, vol. 433, pp. 60–72. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98581-3_5
7. Felli, P., Gianola, A., Montali, M., Rivkin, A., Winkler, S.: Cocomot: conformance checking of multi-perspective processes via SMT. In: BPM, pp. 217–234 (2021)
8. de Leoni, M., Felli, P., Montali, M.: A holistic approach for soundness verification of decision-aware process models. In: Trujillo, J.C., et al. (eds.) ER 2018. LNCS, vol. 11157, pp. 219–235. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_17
9. Maswadi, K., Ghani, N.A., Hamid, S., Rasheed, M.B.: Human activity classification using decision tree and Naive Bayes classifiers. Multimed. Tools Appl. **80**(14), 21709–21726 (2021)
10. Meroni, G.: Artifact-Driven Business Process Monitoring - A Novel Approach to Transparently Monitor Business Processes, Supported by Methods, Tools, and Real-World Applications, vol. 368. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32412-4
11. Meroni, G., Baresi, L., Montali, M., Plebani, P.: Multi-party business process compliance monitoring through IoT-enabled artifacts. Inf. Syst. **73**, 61–78 (2018)
12. Sánchez, V.G., Skeie, N.O.: Decision trees for human activity recognition modelling in smart house environments. Simul. Notes Eur. **28**, 177–184 (2018)
13. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.: Mining process model descriptions of daily life through event abstraction. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2016. SCI, vol. 751, pp. 83–104. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-69266-1_5

14. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.: Log-based evaluation of label splits for process models. Procedia Comput. Sci. **96**, 63–72 (2016)
15. van Zelst, S.J., Bolt, A., Hassani, M., van Dongen, B.F., van der Aalst, W.M.: Online conformance checking: relating event streams to process models using prefix-alignments. JDSA **8**(3), 269–284 (2019)
16. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. Granul. Comput. **6**(3), 719–736 (2021)

An Event-Centric Metamodel for IoT-Driven Process Monitoring and Conformance Checking

Marco Franceschetti(✉), Ronny Seiger, and Barbara Weber

Institute of Computer Science, University of St. Gallen, St. Gallen, Switzerland
`{marco.franceschetti,ronny.seiger,barbara.weber}@unisg.ch`

Abstract. Process monitoring and conformance checking analyze process events describing process executions. However, such events are not always available or in a form suitable for these analysis tasks, for example for manual processes and (semi-)automated processes whose executions are not controlled by a Process-Aware Information System. To bridge this gap, we propose to leverage Internet of Things (IoT) technologies for sensing low-level events and abstracting them into high-level process events to enable process monitoring and conformance checking. We propose an event-centric metamodel for monitoring and conformance checking systems that is agnostic with respect to process characteristics such as level of automation, system support, and modeling paradigm. We demonstrate the applicability of the metamodel by instantiating it for processes represented by different modeling paradigms.

Keywords: Process monitoring · Conformance checking · IoT

1 Introduction

Process monitoring and conformance checking are important tasks of Business Process Management that analyze the execution of a process in terms of *events*. Typically, process monitoring is concerned with the run-time analysis of events describing the execution of a process, emitted by a Process-Aware Information System (PAIS) that coordinates the process execution [6]. In turn, conformance checking analyzes the relation between a normative specification of a process behavior, e.g., a process model, and the actual process behavior recorded in the form of a collection of events, e.g., an event log, describing its execution [4].

The ability to monitor and check the conformance of processes is frequently challenged by the nature of the processes at hand and their domains. These present a spectrum of heterogeneity, differing in dimensions such as process structuredness, modeling paradigm, automation level, or process-awareness, with consequences for the availability and quality of the respective events describing the process executions. At one end of the spectrum are greenfield domains in which full automation is possible, activity execution is process-driven with the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 131–143, 2024.
https://doi.org/10.1007/978-3-031-50974-2_11

support of a PAIS, and all relevant events signifying a relevant state change in a process are generated automatically at the right level of abstraction for analysis: here, monitoring and conformance checking can be realized seamlessly [22]. At the other end of the spectrum are domains characterized by high flexibility, little to no automation, no PAIS support, and no process awareness: here, no process-relevant events are available out of the box and ad-hoc solutions must be deployed to detect events and enable monitoring and conformance checking [27]. Between these extremes of the spectrum are legacy domains, e.g., those of smart factories combining modern and legacy machines. Here, some degree of automation is possible, but the execution is not (fully) controlled by a (single) PAIS, and events are available at varying levels of abstraction [2], ranging from low-level events with no process-awareness to high-level events associated with relevant process state changes: thus, for monitoring and conformance checking, event data collection and processing are required [25].

As indicated in the BPM-meets-IoT manifesto [11], Internet of Things (IoT) technologies can provide a lightweight, transparent, and relatively inexpensive solution to automatically capture relevant execution aspects of processes with limited or no support from a PAIS. Integrating the IoT with processes allows sensing low-level events with no process-awareness; processing these events, e.g., by aggregation and abstraction, allows detecting high-level events that correspond to relevant process state changes, thus enabling monitoring and conformance checking. This may be realized, at design-time, by enriching process specifications with *monitoring points* [9], annotations to the control flow elements associating them with low-level events generated at run-time by the IoT stack. At run-time, the monitoring points capture these low-level events and abstract them into high-level process events describing the process execution [26].

The goal of this work is to define a general IoT-driven process monitoring and conformance checking framework that is agnostic with respect to characteristics such as process structuredness, modeling paradigm, automation level, and PAIS support. To this end, we define an event-centric conceptual metamodel, which only assumes observability of low-level events from IoT for event detection, and availability of a process specification for conformance checking. We validate the metamodel by instantiation in the healthcare and smart manufacturing domains.

This paper is structured as follows: in Sect. 2, we motivate our work with scenarios from the healthcare and manufacturing domains; in Sect. 3, we propose an IoT-based conceptual metamodel for process monitoring and conformance checking; in Sect. 4, we validate the metamodel; in Sect. 5, we discuss related work; in Sect. 6, we draw conclusions and indicate future research directions.

2 Motivation

Process monitoring and checking the conformance against a normative specification are important aspects of BPM, irrespective of the operational support from a PAIS orchestrating the execution. However, these are challenging tasks for manual processes with high human involvement, e.g., in healthcare, but also for processes with high level of automation, e.g., in smart manufacturing, in which

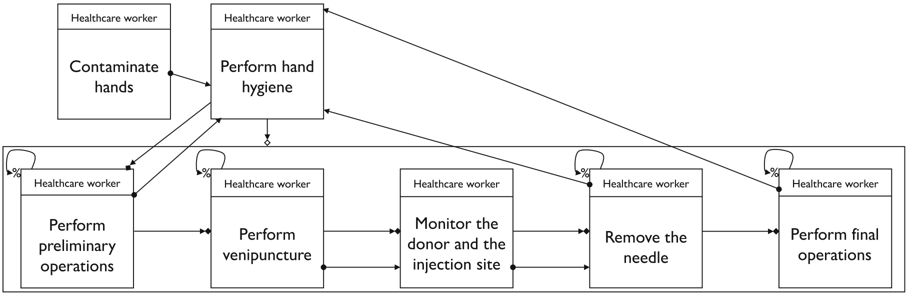


Fig. 1. Example blood donation process represented in DCR graphs notation

data with varying levels of abstraction and little or no process-awareness make it difficult to abstract low-level events to high-level process events [25].

An example from the healthcare domain is the phlebotomy process for blood donation, as defined in the guidelines in [20]. A simplified version of the process composed of five steps, with four additional steps for performing hand hygiene following the indications in [23] is shown in Fig. 1 in DCR graphs notation [10] (cf. Sect. 4.1). The process consists of the following sequence of steps: *Perform hand hygiene (before touching donor)*, *Perform preliminary operations*, *Perform hand hygiene (before aseptic procedure)*, *Perform venipuncture*, *Monitor donor and injection site*, *Remove needle*, *Perform hand hygiene (after risk of exposure to body fluids)*, *Perform final operations*, *Perform hand hygiene (after touching donor)*. The process is fully manual and cannot be automated with current technologies. Moreover, since typically healthcare settings are highly unpredictable, additional, unforeseen hand hygiene steps may be required, for example, due to nurses having to deal with emergencies that might contaminate their hands during the course of the process (cf. activity *Contaminate hands* in Fig. 1). Due to their high flexibility, such processes are often specified through guidelines in natural language or declarative models [17]. Frequently, traces of events of these processes are produced offline by the healthcare workers by manually documenting the operations in some Information System at the end of the work shift. However, this approach is prone to tampering, errors, or imprecision; it is also a cause for increased workload for the healthcare workers, with negative effects on their well-being and operations [8]. Thus, monitoring and checking the conformance of such processes is challenging, yet important, e.g., to ensure that whenever a hand hygiene indication is triggered, nurses disinfect their hands before proceeding with other process steps to reduce the risk of infections [23].

An example from the smart manufacturing domain is the storage process in [25]. It consists of the sequence of steps *Get workpiece from pickup*, *Read color*, *Move to High-bay Warehouse*, *Hold at High-bay Warehouse* + *Store workpiece* (in parallel), *Calibrate*. Unlike the phlebotomy, this process is fully automated and highly structured and predictable, which makes it suitable for modeling with imperative languages such as BPMN [25]. For such processes, sensor events generated by machines at run-time might be available; however, frequently these events are low-level or at different abstraction levels and lack process-awareness. Thus, at least some event aggregation, abstraction, and correlation steps are required to enable monitoring and conformance checking [25].

Prior work points to the IoT as a possible technology to integrate with processes like the above to enable automated process monitoring and conformance checking [11,25]. In the following, we propose a metamodel for such an IoT-driven monitoring and conformance checking framework.

3 Conceptual Metamodel

3.1 Methodology and Requirements

This work is grounded in the Design Science Research principles and inspired by the metamodel development method proposed in [16]. In line with [16], we held a workshop at the Cantonal Hospital of St. Gallen to identify concerns relevant to stakeholders from the healthcare domain. The outcomes of the workshop are the healthcare scenario presented in Sect. 2 and the requirements for an IoT-driven monitoring and conformance checking framework. The scenario and requirements were developed independently by the authors and validated by the healthcare stakeholders in a second workshop. The healthcare scenario was used by the authors to develop the metamodel presented here as the basis for a monitoring and conformance checking framework that is agnostic with respect to the characteristics discussed in Sect. 2. Both the healthcare scenario and the smart manufacturing scenario from [25] were used as representatives of the fully manual and legacy domain to validate the metamodel and its agnostic nature.

The workshops identified the suitability of the phlebotomy process described in Sect. 2. In the process, it is important to identify when healthcare workers are required to perform a hand hygiene as a pre- or post-condition for certain phlebotomy steps, as well as when hand hygiene is performed. The identification should be realized using IoT sensors capable of sensing low-level events in the physical world (e.g., the dispensing of hand sanitizer). These low-levels events should be processed (e.g., aggregated or translated using complex event processing) into high-level process events associated with the respective activities (e.g., phlebotomy and hand hygiene activities), to produce event logs suitable for process monitoring and conformance checking [26].

3.2 CAIRO Events

Events at the process level are at the core of the proposed metamodel. They are the manifestation of the progress of a process instance and refer to the execution of activities (e.g., the completion of a blood drawing activity) or the occurrence of some happening (e.g., the reception of a message), hence they must be at the basis of process monitoring and conformance checking. A widely accepted definition states that an event is an actual happening that is significant, atomic, and instantaneous [28]. Here, we put forward the idea that, in addition, to enable IoT-driven monitoring and conformance checking in processes like the phlebotomy process in Sect. 2, an event has to be detectable (*observable*) and it should carry context information. Indeed, events that cannot be detected do not contribute to monitoring or conformance checking, while neglecting context information might lead to ambiguities in interpreting a trace (cf. [15]): for instance, establishing the right event-activity-case associations might not be obvious without context information. Accordingly, we propose to extend the definition of [28]:

Definition 1 (CAIRO Event). *A CAIRO event is an actual (real world) occurrence that is:*

- Contextualized*: it happens within a context, which characterizes it;*
- Atomic*: it either occurs or not, i.e., it has all-or-nothing semantics;*
- Instantaneous*: it happens at a specific point in time;*
- Relevant*: it refers to a relevant process state change (*“significant” *in [28]);*
- Observable*: its occurrences can be detected.*

Each of the five CAIRO properties is necessary to support comprehensive and unambiguous monitoring and conformance checking of processes, since:

- “AIRO” non-contextualized events might admit uncertainty about their properties and the process instance they belong to, leading to ambiguous trace interpretations. For example, assume a hospital ward with concurrent processes instances involving several healthcare workers, in which a hand hygiene execution event is detected, but the healthcare worker performing it is unknown. In this case, it might be impossible to associate the hand hygiene event with the right resource and process instance with certainty, and to reliably detect hand hygiene indications relevant for conformance checking.
- “CIRO” events lacking the Atomicity property do not conform to the all-or-nothing semantics. These are incompatible with process monitoring and conformance checking, which require clearly identifiable events corresponding to state changes at the process level to match with a process specification.
- “CARO” events are not Instantaneous, i.e., they are not associated with a single timestamp, but to a time interval, leading to uncertainty about their specific time of occurrence (cf. [21]). Consider the phlebotomy process: if the completion of preliminary operations and performance of hand hygiene were not timestamped but had overlapping intervals associated, it would not be possible to identify with certainty in which order these steps were performed.
- “CAIO” events are not Relevant, i.e., they do not reflect process-level state changes. Irrelevant events in a trace might result in ambiguities in the interpretation of the process execution. This is the case, for example, of low-level events: assume, for instance, that a low-level event *“Nurse_proximity < 0.5”* were part of a trace for the phlebotomy process. The interpretation of this event is ambiguous, since it is unclear to which process state change it refers.
- “CAIR” events are not Observable, thus they are not detected, resulting in their absence from the process traces. Incomplete traces missing (unobservable) events might jeopardize monitoring and conformance checking results.

We now propose a conceptual metamodel with CAIRO events at its core.

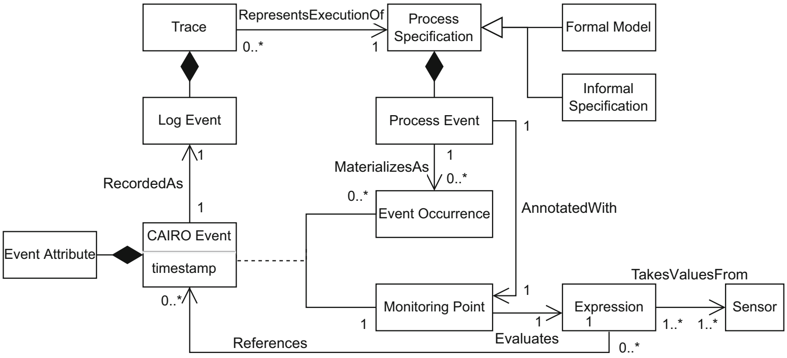


Fig. 2. UML class diagram for the proposed metamodel

3.3 Monitoring and Conformance Checking Metamodel

Figure 2 illustrates the proposed metamodel developed following the principles in [16] and centered around the concept of CAIRO event. A *CAIRO Event* (cf. Sect. 3.2) is a timestamped event at process level of abstraction corresponding to a relevant state change detected by a *Monitoring Point*.

A *Monitoring Point* is an executable object (e.g., an application or a service, cf. [24]) responsible for the abstraction of low-level sensor events with no process-awareness by emitting a *CAIRO Event* at the process level, and the contextualization and correlation of this *CAIRO Event* with a process instance. In line with [1], a *Monitoring Point* evaluates one *Expression* (e.g., a Complex Event Processing query [26]), whose variables are taken from the values of one or more *Sensors* sensing relevant features of the process environment. An *Expression* might *Reference* zero or more prior *CAIRO Events* (and indirectly their event attributes) to identify events unambiguously. This might be useful when knowing prior occurred events allows discriminating between new process-level events. For example, phlebotomy steps *Perform preliminary operations* and *Monitor donor and injection site* might be identified by the same low-level sensor events, but knowing that *Perform venipuncture* occurred before allows identifying the current step as *Monitor donor and injection site*. The *Expression* associated with a *Monitoring Point* is evaluated at run-time based on *Sensor* readings, and a *CAIRO Event* is emitted if it evaluates to `true`.

A *Process Event* represents any control flow element in a *Process Specification* reflecting a relevant state change at process level, e.g., the start of an activity. At design-time, a *Process Event* is *AnnotatedWith* a *Monitoring Point*; one *Monitoring Point* annotates exactly one *Process Event*. At run-time, a *Process Event* may *MaterializeAs* zero or more *Event Occurrences*. For example, *Process Event "Perform hand hygiene"* materializes multiple times during one execution of the process in Fig. 1, resulting in multiple instances of *Event Occurrence*. Each such occurrence is detected (by evaluating its *Expression*) by the *Monitoring Point* that annotates the respective *Process Event*; the *Monitoring*

Point emits a new *CAIRO Event* timestamped with the time of materialization of the *Process Event* as *Event Occurrence*. As *CAIRO Event* is an association class, each of its instances can be reconducted to the *Monitoring Point* that emitted it, and, via *MaterializesAs*, to the *Process Event* whose *Event Occurrence* was detected.

A *Process Specification* is the description of a process defining at least the control flow perspective, e.g., activities and events. To enable monitoring and conformance checking of a broad range of processes (cf. Sect. 2), the metamodel supports processes of different representation paradigms. Hence, the metamodel admits both process specifications in the form of informal specifications in natural language in which *Process Events* can be explicitly and unambiguously identified, such as the activities specified in the guidelines in [20], and formal specifications, e.g., in BPMN. Thus, a *Process Specification* may be either a *Formal Model*, e.g., in BPMN, or an *Informal Specification* in natural language.

A *CAIRO Event* might be associated with *Event Attributes*, whose values are assigned by the respective *Monitoring Point* based on *Sensor* readings. *Event Attributes* are useful for establishing event-instance correlations by storing the process instance context, e.g., as a *CaseId* attribute or trough several attributes forming a candidate key identifying the process instance. A *CAIRO Event* is *RecordedAs* a *Log Event*, part of a (potentially partial) *Trace* describing the (potentially partial) execution of an instance of the *Process Specification*. This trace, along with the process specification, makes conformance checking possible.

3.4 CAIRO Events in the Metamodel

In the metamodel, a *CAIRO Event* is contextualized through the composition association with one or more *Event Attributes*, each describing a contextual characteristic. For example, a CAIRO event for a hand hygiene activity may be associated with attributes: *Resource*, indicating the nurse, *Room* indicating the activity location, and *Dispenser* indicating the specific dispenser used.

Atomicity is enforced in the metamodel by the cardinality constraints associating each occurrence of a *CAIRO Event* with exactly one *Process Event*.

A *CAIRO Event* is instantaneous, since each instance of it has a timestamp denoting the point in time when the respective *Process Event* occurred. For example, a *CAIRO Event* instance with UNIX epoch time 1679309356 places the occurrence of the event on March 20, 2023 at 10:49:16.

Relevance is reflected by the fact that a *CAIRO Event* is associated with a *Monitoring Point* for the detection of the *Event Occurrence*s of the corresponding *Process Event*, and recorded as a *Log Event* in a *Trace*.

The requirement of being observable is expressed by the chain of associations relating each *CAIRO Event* with at least one *Sensor* deployed in the process environment (via *Monitoring Point* that *Evaluates* an *Expression*). For example, to detect a hand hygiene event, one could deploy a proximity sensor to detect the presence of a healthcare worker near a sanitizer dispenser, and a press sensor to detect the dispensing of the sanitizer. An expression evaluates the changes in the values of these sensor readings, resulting in the emission of a *CAIRO Event* if the values satisfy some condition (e.g., `proximity<0.5 && press_force>0.3`).

4 Metamodel Validation

The proposed metamodel is applicable to the spectrum of domains discussed in Sect. 2, irrespective of process structuredness, modeling paradigm, automation level, and PAIS support. To demonstrate such general applicability, we instantiated it for the cases of healthcare and smart manufacturing as two representatives of the fully manual domain and legacy domain, respectively [7].

4.1 Healthcare Domain

For the healthcare domain instantiation, we consider the phlebotomy process described in Sect. 2. The process, like several in this domain, is specified through guidelines in natural language (*Informal Specification* in the proposed metamodel). One could annotate the guidelines by marking all the relevant *Process Events* and annotating them with *Monitoring Points*, based on annotator tools such as the one in [19]. At run-time, these *Monitoring Points* would emit corresponding *CAIRO Events* matching with the annotations in the *Informal Specification*. Here, indeed starting from the guidelines and annotating them with the tool proposed in [19], we derived a formal model in DCR graphs notation [10]: we discuss a metamodel instantiation based on this notation. DCR graphs are a declarative event-based process modeling language, which is particularly suitable for representing healthcare processes requiring a high degree of operational flexibility, e.g., to support the repeated execution of activities. As DCR graph notation is a formal modeling language with defined semantics, the representation of a process in this language is a formal model. Thus, the *Formal Model* concept in the metamodel in Fig. 2 corresponds to the *Process concept* in the DCR graph metamodel (cf. [18]). The *Process Event* concept in our metamodel corresponds to the *Event* concept of the DCR graph metamodel.

Figure 3 shows an excerpt of an instantiation of the metamodel for the case of the phlebotomy process modeled as a DCR graph like in Fig. 1, limited to the *Perform hand hygiene* event. Figure 3a) shows that at design-time the event is annotated with a *Monitoring Point Hand hygiene detection service*, which has an *Expression* that evaluates the values for the healthcare worker's proximity to

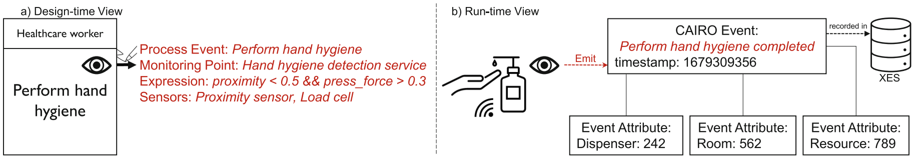

a) Design-time View
Healthcare worker
Perform hand hygiene
Process Event: Perform hand hygiene
Monitoring Point: Hand hygiene detection service
Expression: proximity < 0.5 && press_force > 0.3
Sensors: Proximity sensor, Load cell
b) Run-time View
Emit
CAIRO Event:
Perform hand hygiene completed
timestamp: 1679309356
recorded in
XES
Event Attribute: Dispenser: 242
Event Attribute: Room: 562
Event Attribute: Resource: 789

Fig. 3. Metamodel instantiation for an excerpt of a process in DCR graphs notation

the patient ($proximity < 0.5$) and the force exerted on the sanitizer dispenser by pressing on it ($press\_force > 0.3$). These values are provided by a proximity sensor and a load cell, respectively. At run-time (cf. Fig. 3b)), once the expression evaluates to `true`, a *CAIRO Event Perform hand hygiene completed*, with associated attributes indicating the dispenser (*242*), the room (*562*), and timestamp (*1679309356*) is emitted. The CAIRO event is recorded as a log event in a trace in a XES log to enable conformance checking against the model in Fig. 1.

4.2 Smart Manufacturing Domain

For the smart manufacturing scenario, we considered the production process from [26]. In a metamodel instantiation with processes encoded in BPMN 2.0 (here the modeling language is chosen according to the discussion in Sect. 2), the *Process Event* concept in our metamodel corresponds to the *Flow node* concept of the BPMN metamodel, which entails BPMN process events, gateways, and activities. The *Formal Model* concept in our metamodel corresponds to the *Process* concept in the BPMN metamodel. The *Monitoring Point* concept in our metamodel corresponds to the *Monitoring* concept in the BPMN metamodel.

While the BPMN 2.0 standard states that process events may be also activity events (e.g., the event associated with the start of an activity), it does not define in its metamodel any such event. Here, for simplicity and correspondence to the metamodel, we consider the execution of a BPMN activity to be associated with–and detected at–the completion of the activity: the activity completion detected by a *Monitoring Point* marks the occurrence of a *CAIRO Event* (timestamped with the activity completion time to ensure the *"I"* property). This is compatible with many event logs found in the literature, which associate activity executions with their completion time only (cf. [4] for examples). Nevertheless, if fine-grained activity event detection, e.g., differentiating between start and end, is a requirement, the BPMN metamodel could be extended in line with the extension proposed in [1].

Figure 4 shows the instantiation of the metamodel for the simplified production process in BPMN 2.0 [26]. Figure 4a) depicts the design-time view regarding the extension of a standard BPMN model with a *Monitoring Point* to detect the end of the *Burn Workpiece* activity. The *Expression* associated with the *Monitoring Point* refers to the oven temperature sensor and the status of its door. Figure 4b) presents the run-time view of executing one instance of the *Burn Workpiece* activity. Here, the *Monitoring Point* detects the end of the activity based on the modeled expression and emits a *CAIRO Event Burn Workpiece ended* with the corresponding timestamp.

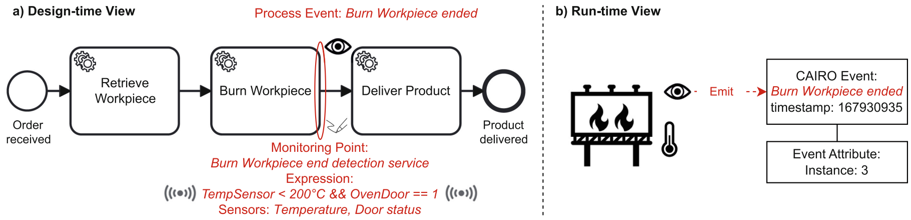


Fig. 4. Instantiation of the metamodel for a BPMN 2.0-based process

5 Related Work

A pivotal work calling for the integration of Internet of Things (IoT) technologies with Business Process Management (BPM) is the IoT-meets-BPM manifesto [11]. Our work relates to several of the challenges indicated in the manifesto, such as *Placing Sensors in a Process-Aware Way*, *Support for Managing Manually Executed Physical Processes*, and *Improving Online Conformance Checking*.

In [5] a systematic literature review can be found, which surveys existing modeling notations for IoT-aware business processes and potential application domains for the integration of IoT technologies with BPM. These could be used as a basis for choosing the most adequate language for modeling a process to be augmented with monitoring points based on IoT technologies. Another work towards the BPM-IoT integration is [13], which proposes an extension of BPMN 2.0 to model, enact, and monitor IoT-aware processes with an addition for IoT-driven events presented in [14]. The work builds on the assumption that a PAIS is available for the orchestration of processes. While the metamodel for monitoring and conformance checking proposed here was motivated by domains in which a PAIS is not available, the methods proposed in [13] could still be applied to a system built following our metamodel. The metamodel proposed in [3] is aimed at bridging process mining with IoT, and focuses on how low-level data is captured by IoT devices and how these data are used to generate an event log suitable for offline process mining tasks. Here, we do not assume monitoring points and the transformation of low-level IoT data into events relevant in the process execution to preclude online monitoring and online conformance checking.

Our proposed metamodel leverages the concept of monitoring point for the detection of occurrences of process events. Monitoring points have been first introduced by Herzberg [9], and subsequently applied to BPMN for process monitoring in [1]. Authors in [1] extend the BPMN metamodel to enable a low-level detection of activity lifecycle events, thus their monitoring metamodel in [1] could be integrated with ours to achieve a fine level of granularity for process event detection. Monitoring points have been adopted also in [12] for conformance checking of processes in a hospital environment, showing the importance of engaging with domain experts for the definition of the monitoring points in a process model. Nevertheless, the work does not consider IoT technologies for the implementation of monitoring points, but manual methods. In contrast, the work in [24] proposes a method to automatically generate services for an IoT-based activity detection; thus, the method is a possible candidate for the generation of instances of the *Monitoring Point* concept in our metamodel.

In Sect. 3.2, we argued the importance of the *contextualized* property of CAIRO events. In line with our position is the work in [15], which proposes a framework for the contextualization of event-activity mappings–a task that is of high relevance when process activities are observed through the occurrence of low-level events detected by IoT sensors, as the proposed metamodel assumes. The proposed framework could be used as a basis for the identification of event attributes that must be detected to contextualize CAIRO events in systems designed following the proposed metamodel.

6 Conclusion and Future Work

Process monitoring and conformance checking are often challenged by process domain characteristics such as level of system support and process structuredness. We presented a general event-centric metamodel to support process monitoring and conformance checking driven by the Internet of Things for the detection of process events, overcoming challenges posed by these characteristics. At the core of the metamodel, we introduced the notion of CAIRO event, emphasizing the significance of the CAIRO event properties. We validated the metamodel by instantiating it for two significantly different scenarios, demonstrating the agnostic nature of the metamodel with respect to domain characteristics.

In future work, we plan to implement a monitoring and conformance checking approach for the phlebotomy scenario in a laboratory simulation environment, and to check process conformance based on informal process specifications. Additionally, we will further investigate the role of the *Contextualized* property of CAIRO events, in particular for the disambiguation of detected events.

Acknowledgments. This work has received funding from the Swiss National Science Foundation under Grant No. IZSTZ0_208497 (*ProAmbitIon* project). The authors thank Estefanía Serral Asensio and the workshop participants for their feedback.

References

1. Baumgraß, A., Herzberg, N., Meyer, A., Weske, M.: BPMN extension for business process monitoring. In: EMISA 2014 (2014)
2. Beerepoot, I., Di Ciccio, C., Reijers, H.A., Rinderle-Ma, S., Bandara, W., et al.: The biggest business process management problems to solve before we die. Comput. Ind. **146**, 103837 (2023)
3. Bertrand, Y., De Weerdt, J., Serral, E.: A bridging model for process mining and IoT. In: Munoz-Gama, J., Lu, X. (eds.) ICPM 2021. LNBIP, vol. 433, pp. 98–110. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98581-3_8
4. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance Checking. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99414-7

5. Compagnucci, I., Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F.: Modelling notations for IoT-aware business processes: a systematic literature review. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) BPM 2020. LNBIP, vol. 397, pp. 108–121. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66498-5_9
6. de Medeiros, A.K.A., et al.: An outlook on semantic business process mining and monitoring. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2007. LNCS, vol. 4806, pp. 1244–1255. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76890-6_52
7. Franceschetti, M., Seiger, R., González González, M.J., Garcia-Ceja, E., et al.: Proambition: online process conformance checking with ambiguities driven by the internet of things. In: CEUR Workshop Proceedings, vol. 3413, pp. 52–59 (2023)
8. Golay, D., Karlsson, M.S., Cajander, Å.: Negative emotions induced by work-related information technology use in hospital nursing. Comput. Inform. Nurs. **40**(2), 113 (2022)
9. Herzberg, N.: Integrating events into non-automated business process environments. Doctoral thesis, Universität Potsdam (2018)
10. Hildebrandt, T.T., Mukkamala, R.R.: Declarative event-based workflow as distributed dynamic condition response graphs. arXiv preprint arXiv:1110.4161 (2011)
11. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., et al.: The internet of things meets business process management: a manifesto. IEEE Syst. Man Cybern. Mag. **6**(4), 34–44 (2020)
12. Kirchner, K., Herzberg, N., Rogge-Solti, A., Weske, M.: Embedding conformance checking in a process intelligence system in hospital environments. In: Lenz, R., Miksch, S., Peleg, M., Reichert, M., Riaño, D., ten Teije, A. (eds.) KR4HC/ProHealth -2012. LNCS (LNAI), vol. 7738, pp. 126–139. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36438-9_9
13. Kirikkayis, Y., Gallik, F., Reichert, M.: A holistic framework for IoT-aware business processes. In: Cabanillas, C., Garmann-Johnsen, N.F., Koschmider, A. (eds.) BPM 2022. LNBIP, vol. 460, pp. 89–100. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-25383-6_8
14. Kirikkayis, Y., Gallik, F., Seiger, R., Reichert, M.: Integrating IoT-driven events into business processes. In: Cabanillas, C., Pérez, F. (eds.) CAiSE 2023. LNBIP, vol. 477, pp. 86–94. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-34674-3_11
15. Koschmider, A., Mannhardt, F., Heuser, T.: On the contextualization of event-activity mappings. In: Daniel, F., Sheng, Q.Z., Motahari, H. (eds.) BPM 2018. LNBIP, vol. 342, pp. 445–457. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11641-5_35
16. Kurpjuweit, S., Winter, R.: Viewpoint-based meta model engineering. In: Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2007). LNI, vol. P-119, pp. 143–161. GI (2007)
17. Lenz, R., Peleg, M., Reichert, M.: Healthcare process support: achievements, challenges, current research. IJKBO **2**(4) (2012)
18. López, H.A., Simon, V.D.: How to (re) design declarative process notations? A view from the lens of cognitive effectiveness frameworks. In: PoEM-Forum 2022 (2022)
19. López-Acosta, H.A., Hildebrandt, T., Debois, S., Marquard, M.: The process highlighter: from texts to declarative processes and back. In: CEUR Workshop Proceedings, pp. 66–70. CEUR Workshop Proceedings (2018)

20. World Health Organization: Who guidelines on drawing blood: best practices in phlebotomy (2010). https://apps.who.int/iris/handle/10665/44294
21. Pegoraro, M.: Probabilistic and non-deterministic event data in process mining: embedding uncertainty in process analysis techniques. arXiv preprint arXiv:2205.04827 (2022)
22. Rinderle-Ma, S., Mangler, J.: Process automation and process mining in manufacturing. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 3–14. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_1
23. Safety, W.P., World Health Organization, et al.: Who guidelines on hand hygiene in health care. Technical report, World Health Organization (2009)
24. Seiger, R., Franceschetti, M., Weber, B.: Data-driven generation of services for IoT-based online activity detection. In: Monti, F., Rinderle-Ma, S., Ruiz Cortés, A., Zheng, Z., Mecella, M. (eds.) ICSOC 2023. LNCS, vol. 14420, pp. 186–194. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-48424-7_14
25. Seiger, R., Franceschetti, M., Weber, B.: An interactive method for detection of process activity executions from IoT data. Future Internet **15**(2), 77 (2023)
26. Seiger, R., Zerbato, F., Burattin, A., García-Bañuelos, L., Weber, B.: Towards IoT-driven process event log generation for conformance checking in smart factories. In: 24th International Enterprise Distributed Object Computing (EDOCW), pp. 20–26. IEEE (2020)
27. Tarallo, A., Mozzillo, R., Di Gironimo, G., De Amicis, R.: A cyber-physical system for production monitoring of manual manufacturing processes. Int. J. Interact. Des. Manuf. (IJIDeM) **12**, 1235–1241 (2018)
28. Wasserkrug, S., Gal, A., Etzion, O., Turchin, Y.: Complex event processing over uncertain data. In: DEBS 2008, vol. 332, pp. 253–264. ACM (2008)

19th International Workshop on Business Process Intelligence (BPI 2022)

19th International Workshop on Business Process Intelligence (BPI)

Business Process Intelligence (BPI) is a domain which covers the mostly quantitatively oriented efforts within Business Process Management (BPM) which serve various topics within both academia and practice. Most notably, with the development of data-driven process analysis BPI has become the foundation of the process analysts' toolbox with various solutions to describe, measure, improve, simulate, enhance, and verify processes.

The BPI workshop has a long-standing tradition of attracting papers which propose new developments within the domain and enticing dissemination and discussion from various angles, including the development of data-driven analysis, the managerial aspects and implications, as well as novel applications and case studies. Hence, the workshop has an audience stemming from various areas including computer science, information systems, formal methods, engineering, and business.

The 19th edition of this workshop attracted 9 international submissions. Each paper was single-blind reviewed by at least three members of the Program Committee. From these submissions, the top 4 were accepted as full papers, and 1 paper was accepted as a short paper for presentation at the workshop, which was held in Utrecht, The Netherlands. The papers were divided over two sessions which were both concluded by a panel with the papers' authors, the chairs, and the audience.

In the first session, two interesting review papers were presented. First Jungeun Lim and Minseok Song presented their paper on *Navigating Event Abstraction in Process Mining: A comprehensive analysis of sub-problems, data, and process characteristic considerations*, which gave an interesting overview of how recorded event instances in a log, but also on lower levels such as IoT data streams living next to the discrete recordings, can be grouped to perform further analytical tasks. Secondly, Selina Schüler and Sascha Alpers presented their paper *State of the Art: Automatic Generation of Business Process Models*. Process models can have various sources and the rise of big data has introduced new construction methods besides the classic manual elicitation. Event logs and process mining naturally have a big part in this context, however, new sources such as unstructured texts and event images present themselves as new sources of business process logic as well. The session was wrapped up by a panel on the impact on event log processing where it was discussed whether there are new innovative domains that are using event logs and hence apply process management and analysis in new contexts. Furthermore, various ideas were formulated that address these new contexts, especially in IoT.

In the second session, the short paper was presented by Zahra Sadeghibogar, Alessandro Berti, Marco Pegoraro, and Wil van der Aalst on *Exploring SLURM Logs through Process Mining: Insights into Scientific Workflows*, which offered an interesting new application area for process mining techniques in the context of high-volume data processing. Next, Jonas Cremerius, Hendrik Patzlaff, and Mathias Weske presented their paper on *Relationships between Change Patterns in Dynamic Event Attributes*, illustrating how tracking and storing the changes within event log attributes can uncover hidden correlations which can be of interest to process analysts. Finally, Neha Rina and Thomas Chatain presented their paper on *Timed Alignments with Mixed Moves*, which touched on a classical process mining problem from a perspective which makes the conformance checking problem harder but also more applicable in realistic situations. The session and workshop were closed by a panel on conformance checking and behavioral patterns. From the room it was clear that conformance checking issues are still very much relevant among the participants' day-to-day issues with current business process analysis. Mostly, the audience agreed that there is a need to make conformance checking results more approachable, in terms of both usability and understandability.

To conclude, the workshop once again brought together an interesting collection of works detailing various problems and suggesting new solutions in a variety of application domains. From the discussions and panels, it was clear that the main focus in the future should lie with usability and applicability of these solutions in practice. We hope that this will be an inspiration for future editions and an aspiration for potential submissions in the coming years. To wrap up this year successfully, selected papers from the workshop will be invited to submit an extended version to a special issue of the Data & Knowledge Engineering journal (DKE).

September 2023

Organization

Workshop Chairs

Andrea Burattin	Technical University of Denmark, Denmark
Johannes De Smedt	KU Leuven, Belgium
Marwan Hassani	Eindhoven University of Technology, The Netherlands

Program Committee

Ahmed Awad	University of Tartu, Estonia
Benoit Depaire	Universiteit Hasselt, Belgium
Claudio Di Ciccio	Sapienza University of Rome, Italy
Chiara Di Francescomarino	University of Trento, Italy
Luciano García-Bañuelos	Tecnológico de Monterrey, Mexico
Anna Kalenkova	University of Adelaide, Australia
Sander Leemans	RWTH Aachen University, Germany
Michael Leyer	Philipps-Universität Marburg, Germany
Fabrizio Maggi	Free University of Bozen-Bolzano, Italy
Jorge Munoz-Gama	Pontificia Universidad Católica de Chile, Chile
Marco Pegoraro	RWTH Aachen University, Germany
Pnina Soffer	University of Haifa, Israel
Eric Verbeek	Eindhoven University of Technology, The Netherlands
Matthias Weidlich	Humboldt-Universität zu Berlin, Germany
Hans Weigand	Tilburg University, The Netherlands
Han van der Aa	University of Mannheim, Germany
Wil van der Aalst	RWTH Aachen University, Germany

Relationships Between Change Patterns in Dynamic Event Attributes

Jonas Cremerius(✉), Hendrik Patzlaff, and Mathias Weske

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
{jonas.cremerius,mathias.weske}@hpi.de, hendrik.patzlaff@student.hpi.de

Abstract. Process mining utilizes process execution data to discover and analyse business processes. Event logs represent process execution data, providing information about activities executed in a process instance. In addition to generic event attributes like activity and timestamp, events might contain domain-specific attributes, such as a blood sugar measurement in a healthcare environment. Many of these values change during a typical process quite frequently. Hence, we refer to those as dynamic event attributes. Change patterns can be derived from dynamic event attributes, describing if the attribute values change from one activity to another. However, change patterns can only be identified in an isolated manner, neglecting the chance of finding co-occuring change patterns. This paper provides an approach to identify relationships between change patterns. We applied the proposed technique on the MIMIC-IV real-world dataset on hospitalizations in the US and evaluated the results with a medical expert. The approach is implemented in Python using the PM4Py framework.

Keywords: Process Mining · Change Pattern Detection · Relationships

1 Introduction

Business organizations analyse and improve their business processes in a data-driven manner. Data describing the actual execution of business processes is often complex and difficult to comprehend. Process mining provides a set of methods and techniques to make sense of this data and allows revealing unknown improvement possibilities [1].

With the increasing adoption of process mining, data specific to the domain under consideration is enjoying increasing attention. Frequent measurements, such as laboratory values (so-called *dynamic* event attributes) can be investigated by deriving change patterns that would characterize how the values of event attributes change [7]. This can help to understand, how patients develop throughout the treatment process. Surprisingly, change patterns can only be identified in an isolated manner, neglecting the chance of finding co-occurring change patterns.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 149–160, 2024.
https://doi.org/10.1007/978-3-031-50974-2_12

This paper investigates how change patterns co-occur within business processes. Identifying relationships between change patterns can be important when analysing, for example, treatment processes. Treatments can have side effects and might affect other laboratory values, which can be overseen due to the high amount of laboratory measurements available [15].

The remainder of this paper is organized as follows. Section 2 provides related work, and Sect. 3 introduces preliminaries. Section 4 presents the approach of identifying relationships between change patterns, and Sect. 5 applies the approach to the MIMIC-IV real-world data set on hospitalizations. We discuss the approach and its limitations in Sect. 6 before the paper is concluded in Sect. 7.

2 Related Work

In this section, we present research dealing with the analysis of dynamic event attributes and the identification of relationships between process characteristics.

In [7], the authors propose a method to detect change patterns in dynamic event attributes by applying statistical tests on event attribute values of activity pairs. The activity pairs can be in a directly or eventually follows relation and have one or more event attributes in common. An OLAP cube is proposed to store and explore the identified change patterns, describing whether the event attribute values tend to increase or decrease in the respective relation.

Whereas the field of dynamic event attribute analysis is rather unexplored, the identification of relationships between process characteristics, such as conducted activities or additional case/event attributes, is an established research area.

The authors in [11] allow correlating different process characteristics, e.g. activity duration, to understand contributing factors of different analysis use cases, such as predicting the next activity or the process outcome. Other works try to find causal relationships to identify reasons for a specific outcome [6]. In the field of concept drift detection, relationships between process characteristics are used to identify reasons for changes within processes over time [2].

It has already been shown, that there exist relationships between process characteristics and additional event attributes, which help to better understand different analysis use cases, such as process outcome prediction [2,6,11]. However, to this date, change patterns are analysed individually, neglecting the chance of finding co-occurring event attribute value changes. This can be problematic in complex processes, such as patient treatments, where side effects can occur, affecting different laboratory measurements [15]. This could lead to wrong conclusions regarding the effect of treatment activities if only positive and expected change patterns are considered. Therefore, this paper provides an approach to detect co-occurring change patterns in business processes.

3 Preliminaries

This paper builds on [7], where the detection of change patterns in dynamic event attributes is introduced.

Definition 1 (Event Log, Trace, and Event). Let V be the universe of all possible values and A be the universe of event attributes. An event e is a mapping of event attributes to values, such as $e \in A \rightarrow V$. The universe of events is defined by $E = A \rightarrow V$. If an event $e \in E$ has no value assigned to an event attribute $a \in A$, it is denoted as $e(a) = \perp$. A trace $t \in E^\star$ is a sequence of events. $T \subseteq E^\star$ represents the universe of traces, such that each event belongs to exactly one trace. An event log L is a set of traces, so $L \subseteq T$.

Normally, an event represents an activity conducted within a certain case at a given time, the latter of which is represented by a timestamp. These properties of events are represented by the event attributes activity (act), caseID, and timestamp. The events of a given trace have the same value for the caseID attribute, and they are ordered by their respective timestamps. For simplicity, we assume that two timestamps of events in a given trace are never equal.

Given events $e_i \neq e_j$ in a given trace t, let $e_i > e_j$ represent a directly follows relation, if e_j appears after e_i and there does not exist an event e_k in t which appears between e_i and e_j, where $e_k \neq e_i \wedge e_k \neq e_j$. Let $e_i \twoheadrightarrow e_j$ represent an eventually follows relation, if e_j appears at any position after e_i in t.

Event attribute value changes can be detected from event logs. These change patterns are stored in a matrix, which is defined next based on [7].

Definition 2 (Change Detection Matrix). A change detection matrix C_L for a given event log $L \subseteq T$ is a set of change analysis cells $c \in C_L$, such that $C_L := A_L \times R_L$, where $A_L \subseteq A$ is the set of event attributes being assigned a value $\neq \perp$ at any event $e \in t, t \in L$ and $R_L \subseteq V \times V$ is the eventually follows relation, including the directly follows relation implicitly. The elements of R_L consist of the respective activity names, so if $e_i \twoheadrightarrow e_j$, then $(e_i(act), e_j(act)) \in R_L$. A change analysis cell $c \in C_L$ represents one cell in the matrix, such that $c = (a, r)$, with $a \in A_L$ and $r \in R_L$. This allows describing, for example, that the dynamic event attribute $a =$ *blood pressure* in the eventually follows relation r *= (Give medication, Consult Physician)* significantly increases or decreases. The proposed approach in [7] considers trace variants as a third dimension, which we omit for the sake of simplicity in this contribution.

4 Approach

The goal of this paper is to identify relationships between change patterns in dynamic event attributes. This contribution takes as input an event log L with its respective change detection matrix C_L.

An example event log is illustrated in Table 1, showing hospital department visits of two patients diagnosed with acute kidney failure within a hospital treatment process. Besides caseID, activity, and timestamp, four additional event attributes are present representing laboratory measurements. During the treatment process, these measurements are taken multiple times.

The measurements are classified as dynamic event attributes, because these are associated with multiple activities and occur multiple times within traces [8].

This property makes them suitable for change pattern detection, which requires multiple values of an event attribute within traces [7].

Table 1. Example event log describing a high-level hospital process, having laboratory values as event attributes. BUN: Blood-Urea-Nitrogen, ICU: Intensive Care Unit

Case ID	Activity	Timestamp	Creatinine	BUN	Creatinine Level	BUN Level
1	Admit to hospital	8:00	1.6	25	High	High
1	Treat in ICU	11:00	1.0	15	Normal	Normal
1	Discharge	18:00	1.5	20	High	High
2	Admit to hospital	9:00	2.5	30	High	High
2	Treat in ICU	10:00	0.5	5	Low	Low
2	Discharge	20:00	0.8	8	Normal	Normal

A change pattern describes a change in event attribute values of one event attribute for a given relation, represented as a change analysis cell $c \in C_L$. Looking at the example in Table 1, the *Creatinine* value decreases in the directly follows relation (Admit to hospital, Treat in ICU) for both cases. Statistical tests are then conducted on the event attribute values of the relation in both cases, indicating a statistically significant increase or decrease [7].

However, *Creatinine* is not the only event attribute decreasing in this relation, as *BUN* decreases as well for both cases. Additionally, the respective level measurements change from high to normal. Thus, we can already see, that there is a significant change in the event attribute values, but we cannot relate them to each other, such that a decrease in *Creatinine* results in a decrease of *BUN*, too. In the following, we present an approach to automatically identify these relationships between change patterns.

4.1 Data Preparation

Given an event log L and a change detection matrix C_L, we propose an approach to identify relationships between change analysis cells $c_1, c_2 \in C_L$. As illustrated in Table 1, c_1 could describe a change pattern for the directly follows relation (Admit to hospital, Treat in ICU) in the event attribute *Creatinine* and c_2 could describe a change pattern for the same directly follows relation (Admit to hospital, Treat in ICU) in the event attribute *BUN*.

To determine a change pattern, the authors in [7] extract all relevant event attribute value pairs of a change analysis cell. Γ_c represents event attribute value pairs of a given change analysis cell $c \in C_L$. For example, Γ_{c_1} includes *Creatinine* value pairs of the directly follows relation (Admit to hospital, Treat in ICU): $\Gamma_{c_1} = \{(1.6, 1.0), (2.5, 0.5)\}$.

While the representation of event attribute value changes in Γ_{c_1} is sufficient to detect change patterns for a single change analysis cell, we require further preprocessing to determine relationships between change analysis cells.

To identify relationships, we need the actual differences in the values as input. Thus, we define Δ_c as the multiset representing the differences of the elements in Γ_c. We first define $\delta(v_1, v_2, a)$, which takes as input two event attribute values $v_1, v_2 \in V$ and the event attribute a itself. It calculates the difference based on the data type of the given event attribute:

Definition 3 (Change Pattern Value Difference). Let $v_1, v_2 \in V$ and $c \in C_L$ with $c = (a, r)$, $a \in A_L, r \in R_L$. We define $\delta(v_1, v_2, a)$ as:

$$\delta(v_1, v_2, a) = \begin{Bmatrix} v_2 - v_1, & \text{if } \mathit{type}(a) = \mathit{continuous} \\ v_1 \cdot v_2, & \text{if } \mathit{type}(a) = \mathit{categorical} \end{Bmatrix}$$

As illustrated in the example event log in Table 1, event attributes can have different data types. They can be continuous or categorical. This differentiation is already conducted when identifying change patterns in [7], by comparing the total number of values against the number of unique values of a variable based on [8]. Therefore, we assume the $\mathit{type}(a)$ function to return the data type of event attributes. For continuous event attributes, we subtract v_1 from v_2 to derive the event attribute value change. It is not as trivial for categorical variables, because it is not possible to calculate a difference there. Therefore, we concatenate v_1 and v_2, creating a new categorical variable as denoted by $v_1 \cdot v_2$.

Given the event attribute value pairs $\Gamma_c \subseteq V \times V$ of the cell $c \in C_L$, we define its change pattern value difference as a multiset Δ_c:

$$\Delta_c = [\delta(v_1, v_2, a) \mid (v_1, v_2) \in \Gamma_c] \tag{1}$$

It should be noted, that all values in Γ_c are $\neq \perp$. Thus, the values in Δ_c are also $\neq \perp$.

Table 2. ΔLog representing multiple change analysis cells Δ_c. Each table cell illustrates one Δ_c of the relation in the respective row. Only directly follows relations are considered here.

Follows Relation	ΔCreatinine	ΔBUN	ΔCreatinine Level	ΔBUN Level
Admit to hospital → Treat in ICU	−0.6 −2	−10 −25	High-Normal High-Low	High-Normal High-Low
Treat in ICU → Discharge	+0.5 +0.3	+5 +3	Normal-High Low-Normal	Normal-High Low-Normal

Table 2 illustrates some examples for Δ_c, where each cell in the table describes one change analysis cell for the directly follows relation in the respective row. We can see, that the changes in *Creatinine* and *BUN* seem to be in a linear relationship. Additionally, the differences in the continuous values are higher if the categorical level indication goes from High to Low than from High to Normal.

In the next step, we describe how we detect these relationships between change patterns automatically.

4.2 Relationships Determined by Correlation Methods

Relationships between mathematical or statistical variables are called correlations in statistics [3]. In this paper, we make use of established correlation methods from statistics to describe relationships between change patterns. Due to the data preparation, we have the value differences Δ_c available as variables, which can be used as input for the correlation methods. Therefore, we define a relationship between two change analysis cells $c_1, c_2 \in C_L$ as a correlation in the change pattern value differences $\Delta_{c_1}, \Delta_{c_2}$.

It is important to note, that correlation does not imply causation. This means, that if we find a correlation, two change patterns occur at the same period, but do not cause one or the other change pattern. Therefore, this contribution studies the co-occurrence of change patterns and not the reason for a change pattern to occur. Formally, we define a relationship identifier as follows:

Definition 4 (Relationship Identifier). A relationship identifier I_L is a function $I_L(c_1, c_2) = (r, p)$, taking as input two analysis cells $c_1, c_2 \in C_L$. It calculates Δ_{c_1} and Δ_{c_2}, and returns the respective correlation coefficient r along a p-value p, indicating the strength and statistical significance of the relationship.

$$I_L \in C_L \times C_L \rightarrow V \times V \tag{2}$$

This allows us to identify relationships for any combination of change analysis cells. The correlation methods, describing the strength of the relationships, are explained next.

Correlation methods have different assumptions regarding the expected data type and distribution of values. Event attributes can be continuous or categorical, which need different treatment when identifying relationships between them. Remember, that we correlate the values of change analysis cells $\Delta_{c_1}, \Delta_{c_2}$, which both relate to an event attribute. Thus, we need to consider all possible data type combinations. We make use of so-called non-parametric tests, because we cannot universally assume normality in the value differences. With that, we accept less power (the probability of correctly rejecting the null hypothesis) on normally distributed value differences, but avoid uncontrollable Type-I (false positives) and Type-II (false negatives) error rates when we would check for normality first [9].

If the value differences in Δ_{c_1} and Δ_{c_2} are continuous, the *Spearman* correlation method is proposed. The test ranks the value differences and determines a **monotonic** relationship in the ranks. For example, the values in Δ*Creatinine* and Δ*BUN* are in a monotonic relationship, as an increase in Δ*Creatinine* results in an increase of Δ*BUN* [3]. For categorical value differences, *Cramer's V* correlation method is proposed in this approach. The test makes use of so-called contingency tables, describing how often category changes co-occur [3]. Looking at Table 2 in the first row, we can see, that the Δ*Creatinine Level* category High-Normal co-occurs with the Δ*BUN Level* category High-Normal. The test looks at the frequency of these co-occurrences and determines a non-random relationship. Correlating continuous and categorical change pattern value differences requires another correlation method.

The *Kruskal-Wallis* test compares the continuous rank distribution across the categories of the categorical variable, similar to the *Spearman* correlation [12]. For example, if we compare the value differences in $\Delta Creatinine$ to the category change in $\Delta Creatinine\ Level$ in Table 2, there is a higher value change (-2) if the category goes from High to Low instead from High to Normal (-0.6).

The relationship identifier $I_L(c_1, c_2)$ provides a p-value p, the probability of obtaining the results at least as extreme as we observed, assuming that there is no relationship between two change patterns (null hypothesis). It also returns the correlation coefficient r, describing the strength of the relationship. A significance threshold α is used to determine statistical significance. In correlation, typical values for α are 0.05 or 0.01. Thus, if $p < \alpha$, we can reject the null-hypothesis and confirm a relationship between two change patters (alternative hypothesis) [3].

In the end, setting α depends on the level of certainty one requires for the analysis, which also depends on the application domain. One should consider adjusting α if multiple tests are conducted for explorative analysis, which is the case in this approach. There exist many methods for adjusting α, where controlling the *false discovery rate (FDR)*, as proposed by Benjamini/Hochberg, is especially appropriate for exploratory analysis. Conservative approaches, such as the Bonferroni correction, control the *family wise error rate*, leading to a strict control of Type I errors. This reduces the power of tests drastically when performing a high amount of them, leading to a high risk of Type II errors [14].

Whereas the significance threshold α can be used equally among different correlation methods, the correlation coefficient r must be interpreted differently. Due to the different data types, the correlation coefficients r cannot be compared across different data type combinations. For example, *Spearman* returns r between -1 and 1, whereas *Cramer's V* is only between 0 and 1 [3]. Nevertheless, there are indications for each r, when a correlation can be considered strong. Therefore, we suggest taking the thresholds for a strong correlation into account when determining a relationship between change patterns, which is $r > 0.6$ or $r < -0.6$ for *Spearman* and $r > 0.25$ for Cramer's V [3].

For *Kruskal-Wallis*, r starts from 0 and has no upper limit. A higher r results in a higher difference of continuous values among categories, which is the standardized variance between categories [12]. Thus, there is no threshold indicating a strong relationship between continuous and categorical values. We suggest taking p as a filter to look at potentially relevant relationships and then comparing them by using r. In the end, the interpretation of p and r is subject to the process analyst. We can only give estimations of the strength of relationships based on current consensus in the statistics community.

With the help of correlation methods, we are able to measure and rank the strength of relationships between change patterns. Next, we approach the correlation of change pattern value differences and process characteristics.

4.3 Correlating Change Patterns with Process Characteristics

So far, we have introduced the identification of relationships between change patterns relating to an event attribute. In process mining, time and resources play an important role as well [1]. In our event log definition, we have defined everything related to an event as an event attribute, including time and resources if available. Thus, the proposed correlations can be applied to time and resources, too. Whereas time is continuous, resources are of categorical nature.

Table 3. ΔLog representing multiple change analysis cells Δ_c including the time and resource perspective

Follows Relation	ΔTime	ΔResource	ΔCreatinine	ΔBUN	ΔCreatinine Level	ΔBUN Level
Admit to hospital → Treat in ICU	3 h 1 h	A → B A → C	−0.6 −2	−10 −25	High-Normal High-Low	High-Normal High-Low
Treat in ICU → Discharge	7 h 12 h	B → D B → D	+0.5 +0.3	+5 +3	Normal-High Low-Normal	Normal-High Low-Normal

With that, it is possible to derive relationships between the time difference of events and change patterns. Table 3 illustrates the change pattern value differences from Table 2 enriched with time and resource changes. Looking at the first follows relation, a shorter time in the directly follows relation (Admit to hospital, Treat in ICU) results in a **higher** change in ΔBUN, for example. Thus, it is possible to identify if a shorter or longer time between events results in a higher or lower change in event attribute values.

The same holds for resources, where in terms of change patterns, resource handovers are correlated to change pattern value differences. As illustrated in Table 3, a handover from resource A to C results in a higher change in ΔBUN in comparison to the handover from A to B.

The approach presented in this paper provides the opportunity to relate change patterns to each other by utilizing methods of correlation. Next, the approach is evaluated on a real-world healthcare dataset.

5 Evaluation

The proposed approach was implemented in Python with the help of the PM4Py framework[1] [4]. We extended the implementation of the original change pattern detection approach to detect and visualize relationships between change patterns [7]. The relevance of this approach is illustrated in a medical environment, where we generated event logs from the Medical Information Mart for Intensive Care IV (MIMIC-IV) database. The reason for choosing this database is its richness of data, allowing to generate event logs with multiple dynamic event attributes. First, we describe the dataset before we present the results.

---

[1] https://github.com/bptlab/relationships-between-change-patterns.

5.1 Dataset

MIMIC-IV is a relational database including hospital treatment processes. Its purpose is to support research in healthcare and is therefore publicly available [10]. The event log extracted from MIMIC-IV incorporates a high-level process, describing department visits of patients during their hospital stay, such as emergency department or intensive care unit (ICU), which is the same event log used for evaluating the original change pattern detection approach [7]. The event log contains 3447 hospital process instances with 13795 events of acute kidney failure (AKF) patients. AKF was chosen together with a medical expert, because of its high prevalence and its measurable disease progression by kidney-specific laboratory values. For each department visit, the event log provides up to 62 event attributes, including laboratory values and demographic information.

5.2 Results

In the following, we present the results of the applied approach on MIMIC-IV. Together with a medical expert, we discussed the identified relationships to evaluate the correctness of the findings.

We found 319 statistically significant change patterns, for which we identified 264 strong relationships ($abs(r) > 0.6$ for *Spearman* and $r > 0.25$ for *Cramer's V*, no threshold for *Kruskal-Wallis*) after applying the Benjamini/Hochberg correction on p for $\alpha = 0.05$ with 56 dynamic event attributes and 39 eventually follows relations [14]. The high amount of strong relationships is also due to many repetitions of the same attribute correlation in different relations.

The strongest and most frequent relationship was identified between *Hematocrit*, *Red Blood Cells*, and *Hemoglobin* in all significant change patterns with very strong correlation coefficients (*Spearman*) $r \geq 0.9$. This is an expected relationship, as these values strongly correlate [5]. Figure 1 visualizes identified relationships for all combinations of data types of event attributes along an enhanced process model. In the process model, the first line of each label represents a change pattern with its test statistic, e.g., **Hematocrit (+0.18)**, indicating that for the majority, but not for all cases, an increase of *Hematocrit* occurs in the relation (Post-ICU Medicine, Discharged). The labels below are the related change patterns with their correlation coefficient, e.g., **Red Blood Cells: 0.94**, saying that there is a strong monotonic relationship between value changes of *Hematocrit* and *Red Blood Cells*.

The plots on the left present the distribution of change pattern value differences in more detail. Plot 1 shows the strong monotonic relationship between *Hematocrit* and *Red Blood Cells*, which is almost aligned as a line with only a few outliers. The medical expert referred to them as potential measurement errors. Plot 2 illustrates the identified relationship between the continuous measurement *Hematocrit* and its respective categorical measurement *Hematocrit Level* indicating abnormality. The plot shows the distribution of change pattern value differences for each change in category. One can now see, that if the value remains low, *Hematocrit* can change in both ways.

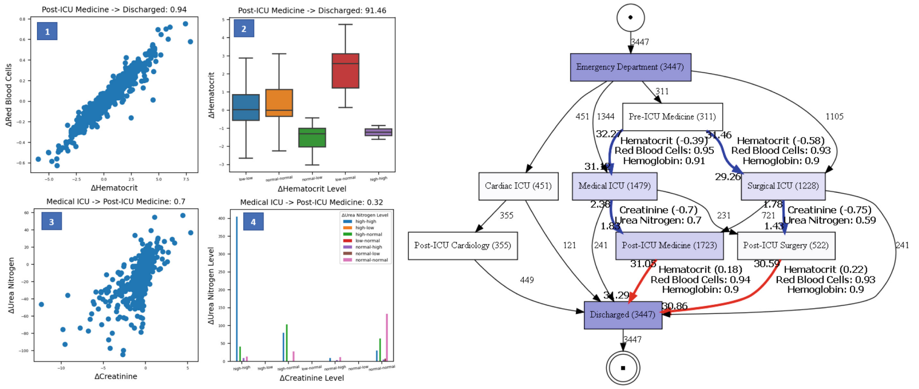


Fig. 1. Visualization of relationships between different change patterns. The plots on the left illustrate the distribution of value changes for all combinations of data types for two event attributes in a relationship. The directly follows graph on the right is enriched with identified relationships. Edges marked in blue indicate an event attribute decrease and red indicates an increase. Filter: Activity (0.8), Path (0.6) (Color figure online)

However, when it changes from normal to low, we logically observe only value decreases. The test statistic of $r = 91.46$ describes the variance between the means observed in the categories, as illustrated in the plot. As described before, it is difficult to estimate the strength of the relationship between continuous and categorical variables. We can only say, that the values in the categories are different by a given variance. Another interesting observation was the change pattern indicating a value decrease in *Creatinine* after ICU treatment, which is an important predictor of renal recovery [13]. As illustrated in the process model, it is strongly correlated with a change pattern related to *Urea Nitrogen (BUN)*. Even though both indicate renal recovery, they do not correlate when conducting one blood test [13]. Together with the medical expert, we found out, that dialysis is conducted during ICU treatment, which has an effect on both measurements. Hence, we observe a correlation in the value decreases of both measurements, even though these measurements do not correlate when analysing them at one point in time. This is interesting because we observe this correlation only at a certain part of the process, where conducted treatment activities lead to the correlation. Plot 3 plots the change pattern value differences, which are more scattered due to the lower correlation coefficient of $r = 0.7$.

Plot 4 shows the correlation between two categorical measurements of *BUN* and *Creatinine*. It shows, how often the categorical changes co-occur for both measurements, e.g., when *BUN Level* changes from high to high, *Creatinine Level* also changes from high to high. The test statistic of $r = 0.32$ (*Cramer's V*) indicates a strong correlation for that. All in all, this evaluation has shown that relationships in dynamic event attributes can be detected, providing novel insights about the process. We could not present all relationships and focussed on the ones which we together with the medical expert found most interesting. The other relationships can be explored using our tool provided in the GitHub repository.

6 Discussion

In this contribution, we propose an approach to identify relationships between change patterns by utilizing correlation methods from statistics. This allows retrieving novel insights about change patterns, providing a better understanding of how event attributes change throughout the process.

We could reveal relationships in continuous and categorical event attributes, which could be confirmed by the medical expert together with medical literature. Interestingly, relationships between change patterns can also provide hints about what is happening in rather abstract processes, such as hospital department visits, where the correlation of value decreases in *BUN* and *Creatinine* indicate dialysis treatment in the ICU.

Furthermore, we found correlations in change pattern value differences, even though the values do not correlate when looking at them at one point in time, such as *BUN* and *Creatinine*. Thus, our approach reveals relationships dependent on activities conducted at certain parts of the process, illustrating that considering the process perspective can provide important context information when analysing dynamic event attributes.

This approach utilizes correlation methods to identify relationships between change patterns. Whereas the correlation in-between continuous and categorical event attributes provides a measure for the strength of a relationship, the correlation of continuous and categorical event attributes can only be described by a p-value and a measure of variance between categories. This requires more expertise in interpretation, which is a limitation of this approach. Additionally, the correlation in-between continuous event attributes is limited to monotonic correlations. Further, we propose to identify relationships between all change patterns, resulting in a high amount of statistical tests. One could also think about identifying relationships only in change patterns of interest, reducing the amount of statistical tests conducted drastically.

Even though this paper focusses on the healthcare domain, this approach is applicable to any event log including dynamic event attributes. For example, the manufacturing domain could be interesting with lots of sensor data available.

7 Conclusion and Future Work

This contribution researches methods to identify relationships between change patterns. This enables process analysts and domain experts to better understand how event attributes change together during the process. We have shown, that process-specific relationships occurring only between certain activities can be identified, which were evaluated together with a medical expert.

Future work could deal with the identification of relationships beyond monotonic ones, such as quadratic relationships. Additionally, methods to better describe correlations between continuous and categorical event attributes could be researched. Another interesting field of research could be the explanation of detected change patterns and their relationships, such that the relationship of *BUN* and *Creatinine* occurred due to dialysis conducted during ICU treatment.

References

1. van der Aalst, W.: Process Mining. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. Adams, J.N., van Zelst, S.J., Quack, L., Hausmann, K., van der Aalst, W.M.P., Rose, T.: A framework for explainable concept drift detection in process mining. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 400–416. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_25
3. Akoglu, H.: User's guide to correlation coefficients. Turk. J. Emerg. Med. **18**(3), 91–93 (2018)
4. Berti, A., et al.: Process mining for python (PM4PY): bridging the gap between process- and data science. CoRR abs/1905.06169 (2019)
5. Billett, H.H.: Clinical Methods: The History, Physical, and Laboratory Examinations. Butterworths, Boston (1990)
6. Bozorgi, Z.D., et al.: Process mining meets causal machine learning: discovering causal rules from event logs. In: 2020 2nd International Conference on Process Mining (ICPM), pp. 129–136 (2020)
7. Cremerius, J., Weske, M.: Change detection in dynamic event attributes. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds.) BPM 2022. LNBIP, vol. 458, pp. 157–172. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16171-1_10
8. Cremerius, J., Weske, M.: Supporting domain data selection in data-enhanced process models. In: Wirtschaftsinformatik 2022 Proceedings, vol. 3 (2022)
9. García-Pérez, M.A.: Statistical conclusion validity: some common threats and simple remedies. Front. Psychol. **3**, 325 (2012)
10. Johnson, A., et al.: MIMIC-IV. In: PhysioNet (2022)
11. de Leoni, M., van der Aalst, W., Dees, M.: A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. Inf. Syst. **56**, 235–257 (2016)
12. McKight, P.E., Najab, J.: Kruskal-wallis test. The Corsini Encyclopedia of Psychology, p. 1 (2010)
13. Schiffl, H.: Discontinuation of renal replacement therapy in critically ill patients with severe acute kidney injury: predictive factors of renal function recovery. Int. Urol. Nephrol. **50**(10), 1845–1851 (2018)
14. Storey, J.D.: The positive false discovery rate: a Bayesian interpretation and the q-value. Ann. Stat. **31**(6), 2013–2035 (2003)
15. Whitehead, N.S., et al.: The effect of laboratory test-based clinical decision support tools on medication errors and adverse drug events: a laboratory medicine best practices systematic review. J. Appl. Lab. Med. **3**(6), 1035–1048 (2019)

State of the Art: Automatic Generation of Business Process Models

Selina Schüler(✉) and Sascha Alpers

Karlsruhe Institute of Technology (KIT), Institute of Applied Informatics and Formal Description Methods (AIFB), Kaiserstr. 89, 76133 Karlsruhe, Germany
`{selina.schueler,sascha.alpers}@kit.edu`

Abstract. Manual Business Process Modeling can be a time-consuming and error-prone process, resulting in low-quality models. Therefore, various approaches have been proposed in research and practice to partially or fully automate the creation of business process models. We conducted a systematic literature review (LR) on automatic generation of business process models. The LR demonstrates that there exist various approaches for automated generation of business process models, which differ in terms of the input data, generation methods, and modeling languages utilized. Despite active research into automatic model generation, there is still considerable scope for improvement. New technologies are continually being developed to support automatic model generation. Additionally, the increasing digitalization of processes and organizations generates vast amounts of data that can be harnessed for modeling.

Keywords: Business Process Model · automatically generated models · Systematic Literature Review

1 Introduction

Business Process Modeling plays a critical role in business process management (BPM). Alongside choosing the appropriate modeling language based on the modeling purpose, the quality of the model and the time required for its creation are also pivotal for effective BPM. Two important roles for Business Process Modeling are identified: the *knowledge carriers* (i.e. business domain experts and other stakeholders), who possesses the knowledge about the natural or artificial original and provides it for modeling, and the *modeler*, who is responsible for modeling the business process. These roles can be fulfilled by one person or several individuals, provided they possess the necessary knowledge and skills [33]. Additionally, there is a user or machine who utilizes and interprets the resulting model. If the modeler and the knowledge carrier are different people, communication issues and transformation effects may lead to errors and time consumption, resulting in low-quality business process models. Transformations occur during the mapping of an original to a model through simplification, mapping, and interpretation [26]. Moreover, a lack of modeling or process expertise

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 161–173, 2024.
https://doi.org/10.1007/978-3-031-50974-2_13

can also lead to low quality in business process models. As a result, various approaches have been proposed in research and practice to partially or fully automate the creation of business process models. The provision of information by the knowledge carrier can be supported by the use of selected data within the business process. Additionally, the modeling task performed by a modeler can be replaced by an algorithm that takes the data as input and automatically generates a business process model as output. The validation can be tackled by an automatic quality assessment of the business process model [30]. In the case of semi-automatic or automatic generation of business process models, the quality of the generated models may also be inadequate if the generation algorithm produces low-quality results. This can occur, for example, due to insufficient checking, failure to meet the selected requirements for the business process model, or inadequate data quality. However, using low-quality business process models as a basis for communication, decision-making, process analysis, or execution can lead to problems [21].

To explore existing approaches for automatic generation of business process models and analyze them systematically, a LR will be conducted based on the guidelines of [15]. The LR comprises three steps: planning, execution, and consideration of the results. To identify further relevant literature beyond the initial LR results, the forward and backward search technique of [32] will be applied. In this LR, we have chosen a generic and inclusive search strategy to maximize the identification of various approaches and find their associated challenges. Despite there are already LRs focusing on a specific area (e.g. generation of models in healthcare or from source code), our investigation is not limited to these restrictions. Given that the challenges are likely to relate to numerous generation approaches, our intention is to synthesize these challenges. In Sect. 2, we present background on Business Process Modeling and Data in Business Processes. This is followed by the presentation of our LR. We briefly describe our method for conducting the study in Sect. 3 and then present our findings on the state of the art in Sect. 4 and research needs in Business Process Modeling in Sect. 5. Based on our findings, we then discuss recommendations for automatic generation of business process models in Sect. 6.

2 Business Process Modeling

Business process modeling languages, such as Petri nets, event-driven process chains (EPC), Business Process Model and Notation (BPMN), UML activity diagrams (UML-AD), or their extensions, typically emphasize the control-flow aspects of business processes by representing the order of activities [33]. The partial order of activities arises from physical facts between activities (e.g. an invoice cannot be sent until it has been created) or organizational constraints (e.g. defined by the company or law). While there are other perspectives to consider, this paper focuses on approaches that automatically generate models with a control-flow-oriented perspective. Other modeling languages are available for different modeling purposes, such as Entity-Relationship model and UML class diagrams for data models, and UML statecharts for statecharts.

The modeling of business processes is often done manually by modelers who initially collect detailed information about the business process to be modeled, for example, through interviews with knowledge carriers, analyses of documents or through their own observations. To automatically generate a comprehensive business process model by algorithms, it is necessary to consider relevant data associated with the process[1]. If data is considered in terms of its relevance for the creation of a business process model, it can be understood as information, which can be divided into *carriers* and *carried information*. The carriers refers to the medium or material that carries or transmits information. The carriers of information can be storage media, such as a database used to store and organize data. Similarly, files can be considered carriers because they represent a logical unit of data. Carriers can differ based on the data format, which defines the structure of the data related to a specific data model. Therefore, data can be classified as structured, semi-structured, or unstructured. When generating a business process model, it is crucial to consider the relevant data associated with the business process. The relevance of the data depends on the carried information, which can include data that is explicitly assigned to a business process and company data that cannot be specifically assigned. It is also important to determine whether the data refers to the process type, a process instance, or to objects. Communication data is another category, as it can relate to several of these categories. Thus, based on the carried information, the following categories can be defined: process instance related data; process type related data; process object related data; enterprise related data; communication data.

3 Development of Search Protocol

In the following sections, the implementation of the LR is only briefly described, a detailed documentation of this LR can be found in [24]. First, the search process is described, followed by the study selection and quality assessment.

Search Process: To conduct a comprehensive and accurate LR, a well-designed search protocol is crucial in narrowing down the vast amount of available publications to a manageable set while ensuring that no relevant publications are missed or excluded. The search sequence is defined by first deriving the most relevant search terms from the research purpose and relevant publications already known, including titles, given keywords, and abstracts. Synonyms, alternative spellings, and related terms are then identified to broaden the search spectrum and cover variations. For this LR, the search sequence should refer to *business processes*, *models*, and *automated modeling*. The exact terms for the English and German language literature search can be found in [24] (register 2). When applied to title, abstract, and keywords in the Scopus database, this resulted in over 35,000 publications, which is too many to manage. To refine the search, further restrictions have been made to ensure that resulting publications refer to the area of business process modeling. For example, the query of terms from

---

[1] The term "generate" will be used to refer to the automatic model generation process.

the areas of *business process* and *model* is restricted to the title, while the query of terms from the area of *automated modeling* is still done on title, abstract, and keywords. This is because the term *model* is also used in other research fields, such as physics and medicine. By restricting search to title, publications obtained are more likely to be focused on business process modeling. Nevertheless, we aim to ensure the comprehensive coverage of relevant literature in our review, related LRs were considered on the one hand, and an extensive forward and backward search was performed on the other hand.

Study Selection and Quality Assessment: After reviewing previous LR in the field [11,23], relevant literature databases for this study were identified as Scopus, ACM Digital Library, IEEE Xplore, SpringerLink, Science Direct, Web of Science, DBLP, and Google Scholar. Explicit inclusion and exclusion criteria were established to ensure the relevance of the results. Relevant publications were checked based on their title, abstract, and/or full text to exclude those unrelated to business process model generation. The complete criteria are listed in [24]. The quality of the selected LR publications will be evaluated using carefully selected quality criteria. To achieve this, seven questions were formulated as quality criteria based on evaluation models from [2,8]. These questions will be used to assess the quality of the relevant publications and to ensure a minimum quality. This step is described in [24]. In addition to the quality assessment, certain characteristics of the publications will be examined to support a structured analysis and evaluation of the results. For example in the *Input Data* category, publications will be categorized according to the input data format and category used to generate a business process model.

4 Results

The LR was conducted in September 2022 in the selected databases. The initial literature search with the search sequence yielded 3617 publications, which formed the basis for subsequent reviews. Next, publications were excluded based on the predefined exclusion criteria, 3030 unique publications resulting in 382 publications, which were then further screened for relevance using the full text (remain 60). As a final step, a forward and backward search was conducted[2], resulting in the identification of an additional 65 relevant publications. In total, 125 publications from the research area of business process model generation were included in this study. A structured overview of the results according to further aspects such as qualitative evaluations of individual criteria of the approaches can be found in [24]. The following section presents the results of the LR. It begins by reviewing selected descriptive properties. Then, the publications are discussed to provide an overview of related work.

---

[2] The publications that utilize process mining algorithms based on event logs and process only standardized event logs were excluded during this step.

4.1 Descriptive Statistics

Figure 1 examines the results of the LR based on different descriptive properties. In particular, Fig. 1, located at the top left, demonstrates the results categorized by publication year. The oldest identified publication, dating back to 1998. In 2015, the most publications were published, indicating an active pursuit of research on model generation, with no sign of decline. However, the lower number of publications in 2022 can be attributed to the fact that only those published until August (in some cases September) were included due to the study's timeline, and publications before that may not have been fully available in the databases. Furthermore, Fig. 1, located at the top right, shows the development level of the publications. It indicates that most publications technically implement and evaluate the developed concepts. However, the quality of the evaluation was not taken into account in this figure. Some publications only present one concept or evaluate the concept based on a (mental) example without implementation. Figure 1, located at the bottom left, shows that most publications generate business process models based on event logs and text, as these are highly accepted and often readily available in companies. On the other hand, code and knowledge base inputs are less represented. Nevertheless, some publications are based on code that generates models, such as class or object diagrams. However, these were not considered in this LR, as they cannot be assigned to business process models but are developed and used primarily in the domain of software architecture and maintenance. Lastly, the figure on the bottom right shows that most algorithms offer models in the modeling languages Direct Follow Graph and BPMN, with Direct Follow Graphs often being the result of process mining algorithms. Meanwhile, other modeling languages were used less than three times and were grouped into the "Others" category.

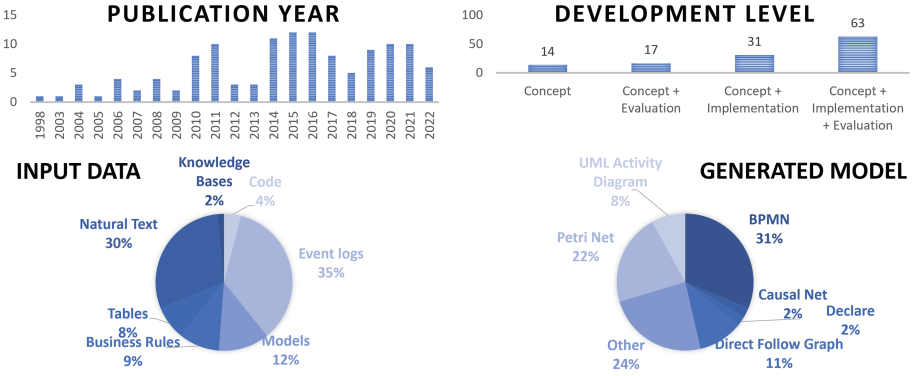


Fig. 1. Result statistics

4.2 Model Generation Approaches

This chapter will delve into the approaches described in the relevant publications, categorizing them based on the input data format. Thus, publications with similar methods for processing these data are considered together, while differences in the methods for the same input data formats are clarified. This approach allows for a clearer distinction between approaches, while avoiding a poor comparability that could result from structuring based on the modeling language used to generate the models. Although the output can be used for further consideration, it is not ideal for comparison purposes because different methods (such as process mining and natural language text-based models) have largely different approaches and are therefore only comparable to a limited extent. Furthermore, as outlined in Sect. 2, the category of carriers to which these input data belong is specified.

Approaches Based on Source Code: Five publications use code as input data to generate business process models. These input data are all enterprise related. One approach described in this context involves partitioning code into topics and subtopics based on the object model and identifying their connections. One approach use process discovery to create a knowledge discovery meta-model (KDM-model) from code to identify elements and their connections. By applying heuristic rules to the KDM-model, relevant business process elements are identified and then mapped to BPMN elements. In another approach, an abstract syntax tree is first extracted from code classes to derive business rules, which are then used to generate an activity diagram followed by a BPMN model.

Approaches Based on Business Rules: Eleven publications use business rules as input data to generate business process models. Although the majority of the data are related to process types, one publication provides data related to the enterprise. Five publications use Semantics of Business Vocabulary and Business Rules (SBVR) to formally describe business rules. SBVR can be used to formally describe business rules and terms. Seven approaches generate business process models from business rules that are not expressed in SBVR. The approaches based on business rules utilize meta-models or transformation rules to map the rules to model elements. Only one approach conducts a semantic analysis to identify relevant elements within the rules. Meanwhile, one approach generates event logs by instantiating these business rules.

Approaches Based on Tables: Ten publications have demonstrated the generation of business process models from tables, such as Excel spreadsheets, which store various types of data, including numbers and strings. The tables discussed in this context can represent a variety of data types such as use cases, business rules, business process descriptions, location data, and activity data. The majority of the data is related to process types, but the data also includes enterprise-related and object-related data, as well as communication data. In some cases, the activity data is manually entered by process participants using a web-based tool, providing details on input and output data, as well as optional or required activities. The approaches used in this context involve mapping rules, as well as transformation rules, process mining algorithms and machine learning techniques such as Latent Dirichlet Allocation.

Approaches Based on Models: 15 publications propose generating business process models from existing graphical models using structured exchange formats and transformation rules. This method allows for the mapping of elements from the original model to those in the output model. The input models can be diverse, such as data models, object relation models, product data models, DMN models, gain models, and use case models. Therefore, the majority of the data is related to process types or enterprise, but the data also includes one object-related data.

Approaches Based on Knowledge Bases: Two publications describe approaches for generating business process models using knowledge bases. These knowledge bases include ontologies of business rules and requirements, which are used for reasoning and generating BPMN models. In one approach, a manual ontology creation process is followed, and business rules are then defined based on this ontology. This approach aims to enhance the consistency and understanding of business rules, as well as improve their reusability. Therefore, the data is process type or enterprise related.

Approaches Based on Event Logs: 44 publications use event logs to generate business process models. Event logs are recorded by information systems and consist of a set of traces, which are defined as a sequence of events associated with the same process instance and case ID. Generally, these event logs are process instance or enterprise related, and some data is process type related. LR approaches that process event logs use process discovery algorithms or artificial intelligence techniques like clustering or convolutional neural network-based algorithms. Different algorithms exist in research and practice for generating business process models using process discovery algorithms, depending on the desired modeling language and properties of the event logs to be analyzed [1]. Approaches have delved into the generation of event logs through the utilization of various types of data sources. These sources range from code, rules, tables, models, emails, to text documents. Some approaches leverage extended databases as event logs and use either manually created path possibilities or logs describing state changes and resource usages to generate event logs. Additionally, one approach generates event logs with different perspectives by creating a metamodel based on different change logs of databases. Other approaches incorporate information about data exchange between organizational roles in their process discovery algorithms.

Approaches Based on Natural Text: 37 publications use natural text to generate business process models. Machine learning is commonly used to generate business process models from natural language information, which is often available in unstructured form in companies. Although natural language has the advantage of being easily understood by a larger group of people, it can also lead to misunderstandings and inconsistency in analysis due to its unstructured form. Several LRs on this topic have been conducted [6,18]. There are three distinct approaches to processing natural language texts, as identified in these LRs. The first is rule-based methods, which assume that sentences follow certain rules and are particularly useful for process extraction. The second is pattern matching, which involves developing speech patterns that consider basic speech patterns, syntactic trees, and anaphora resolution. However, these approaches often require domain-specific knowledge bases and are not suitable for open-domain or open-topic scenarios. The third approach is Natural Language Processing (NLP) methods, which involve extracting subject-verb-object constructs, participants, and splitting words (and/or/...) for processing. These procedures include tokenization, tagging, parsing, chunking, and Part-of-Speech marking. Most approaches to generating business process models from natural language use NLP, with input ranging from process descriptions to use case descriptions, user stories, business process rules, policy documents, software requirements, and cooking recipes. Therefore, the majority of the data is related to process types or enterprise, but the data also includes communication data. The modeling languages used include BPMN, Petri nets, UML activity diagrams, EPC, and process structure trees.

In addition to the approaches discussed above that generate models based on different input data, there are other approaches that derive input data from other sources. For instance, SBVR rules can be derived from use case diagrams [5] or natural language text [28]. Furthermore, some approaches aim to generate models from other data sources, such as creating a DMN model from text [13]. By combining these approaches, it may be possible to generate business process models from various input data sources. However, it should be noted that additional transformations in the process may result in the loss of important information.

5 Discussion

The results presented in this LR vary in terms of input data used, modeling language, and developed methods. Business process models are generated based on several types of input data, including code, rules, tables, models, knowledge bases, event logs, and natural language texts. The motivation behind generating models is to save time, reduce costs, and ensure consistent model quality. However, these approaches also have some limitations that need to be considered. By conducting an extensive analysis of the existing literature, this LR also presents the challenges and shortcomings of these approaches.

5.1 Challenges of Business Process Model Generation

Approaches that generate models depend on the *quality of the input data*. During model generation, only the information that is represented in the input can be included in the generated model. For instance, in process mining methods, it is crucial to identify the relevant systems and not remove any relevant aspects while cleaning up the input. If the input used does not include the relevant sequences of activities, the generated model will not accurately represent the process [1]. However, if the input is too extensive, generating models may become overly complex. These limitations are not just relevant for approaches using event logs.

In natural language approaches, sentences can be complex, and redundancies and ambiguities can be present in requirements documents used as input [16,19]. Additionally, different words can be used for the same thing, which can lead to differently generated activities in a model when processed. The level of detail of a description can also be different, and the input can contain errors and inaccuracies as well as a subjective perception of the person describing the process [9]. Texts, such as company guidelines, can be incomplete, leading to gaps in the generated business process models. To fill these gaps, additional data collections are required [16], and the approaches often have problems resulting from changes in the active and passive voice of the input text, possible rephrasing and changes of order and conditions that are explicitly noted [19]. Similarly, transforming meta-statements and example sentences that should not lead to model elements can also lead to errors. Moreover, the process is often described using multiple files, but some (for example natural language- and model-based) approaches restrict themselves only to one input file [17]. These restrictions can be reasonable, but often lack justification. Therefore, the quality of the approaches that generate models depend on the developed *algorithms* and how they process the data. Furthermore, currently used algorithms mainly use syntactic patterns, but capture little semantics in the sentences [16]. In addition to the excluded elements, complex patterns in the generated business process model are also excluded and often, only selected patterns such as the basic control flow pattern are supported during the transformation [14]. This restriction allows to focus on their implementation and, if necessary, to avoid too many additional restrictions [9]. If no such patterns are used, the constructed business process models may contain syntactic errors [16]. Consequently, the *quality of the generated models* can also be problematic.

5.2 Improvements Addressing the Challenges

Based on the identified deficiencies of existing approaches for generating business process models (the quality of input data and of the generated model, algorithm complexity) some improvement options have already been proposed:

Improvement of the Input Data Quality: One approach to improve the quality of input data is to *preprocess input data*. For example, [3] presents a method for pre-extracting information (activity, roles, relationships) from natural language texts using the language model Generative Pre-trained Transformer (GPT), thereby transforming them into structured process descriptions. Additionally, [10] demonstrates that language models such as GPT can be applied to model generation by transforming natural language texts into formal exchange formats. Another approach is to *add more information*, as suggested by [7], who transform disparate data such as activities, paths, event logs, and models into vectors. Using a neural network, the activities are then learned from the event logs, taking context into consideration. Another proposed approach is to *compose activities* in event logs by partial models [29].

Improvement of the Algorithms: To improve the algorithms used for generating business process models, *contextual information can be specified*, e.g., derived from behavioral similarities of process descriptions and contextual features, to enable improved relevance assessment of paths [12]. Additionally, *better incorporation of semantic features* can be achieved by using research results from other areas such as semantic web, ontology, and process reference models [16].

Improvement of the Quality of the Generated Model: Several approaches have been proposed to improve the quality of the generated model. One approach is to use *repair algorithms based on adding additional data*. For example, [20] proposes to improve a model based on the associated event log by first dividing the model into fragments with well-defined boundaries. The fragments that do not fit the event log are replaced with new fragments created using some well-known process detection algorithms. Then, the repaired model is assembled from the fragments. This method repairs the model locally to improve readability. [27] incorporates information on business goals for model improvement. Another approach is to use *repair algorithms based on model properties*, such as algorithms that correct Petri nets that do not satisfy the soundness property. [12] generates a small set of alternative models that resemble the original model but potentially contain fewer or no errors in the process behavior. Other approaches deal explicitly with improving cycles [4] or naming model elements. To *reduce complexity*, [22] presents an approach to consider the number of activities and data elements, the complexity of the control and data flow, and adapt the model to be improved. [31] develops an algorithm to simplify the model through abstraction. Additionally, *using patterns* can improve model quality, as suggested by [25], who generates patterns from model repositories to support modeling.

6 Conclusions

We conducted a LR on business process model generation, selecting 125 publications. We analyzed the selected papers from two perspectives: similarities and differences as well as challenges. Based on the input data, we categorized them into seven groups (code, tables, business rules, models, knowledge bases, event logs, natural text) and identified the main methods for generating models. Regarding challenges, few publications explored the quality of the generated models, and the limitations of these approaches concerning inputs, procedures, and generated models. We suggested future research possibilities and challenges for improvement. The LR shows diverse approaches for generating models, which vary based on input data, methods used, and modeling languages provided, but there is potential for improvement with new technologies. Furthermore, for example the increasing digitization of processes and organizations creates more data that can be utilized for generation.

References

1. van der Aalst, W.M.P.: Process Mining: Data Science in Action, 2nd edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. Adamo, G., Ghidini, C., Di Francescomarino, C.: What is a process model composed of?: a systematic literature review of meta-models in BPM. Softw. Syst. Model. **20**(4), 1215–1243 (2021). https://doi.org/10.1007/s10270-020-00847-w
3. Bellan, P., Dragoni, M., Ghidini, C.: Leveraging pre-trained language models for conversational information seeking from text (2022). arXiv:2204.03542
4. Chadli, N., Kabbaj, M.I., Bakkoury, Z.: An improved ad hoc approach based on active help method to detect data flow anomalies in a loop of business process modeling. In: Proceedings of the 1st International Conference of Computer Science and Renewable Energies, Ouarzazate, Morocco, pp. 284–290. SCITEPRESS - Science and Technology Publications (2018). https://doi.org/10.5220/0009775402840290
5. Danenas, P., Skersys, T., Butleris, R.: Enhancing the extraction of SBVR business vocabularies and business rules from UML use case diagrams with natural language processing. In: Proceedings of the 23rd Pan-Hellenic Conference on Informatics, Nicosia Cyprus, pp. 1–8. ACM (2019). https://doi.org/10.1145/3368640.3368641
6. De Almeida Bordignon, A.C., Thom, L.H., Silva, T.S., Dani, V.S., Fantinato, M., Ferreira, R.C.B.: Natural language processing in business process identification and modeling: a systematic literature review. In: Proceedings of the XIV Brazilian Symposium on Information Systems, Caxias do Sul Brazil, pp. 1–8. ACM (2018). https://doi.org/10.1145/3229345.3229373
7. De Koninck, P., vanden Broucke, S., De Weerdt, J.: act2vec, trace2vec, log2vec, and model2vec: representation learning for business processes. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 305–321. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_18
8. van Dinter, R., Tekinerdogan, B., Catal, C.: Automation of systematic literature reviews: a systematic literature review. Inf. Softw. Technol. **136**, 106589 (2021). https://doi.org/10.1016/j.infsof.2021.106589
9. Fan, S., Zhao, J.L., Dou, W., Liu, M.: A framework for transformation from conceptual to logical workflow models. Decis. Support Syst. **54**(1), 781–794 (2012). https://doi.org/10.1016/j.dss.2012.09.006
10. Fill, H.G., Fettke, P., Köpke, J.: Conceptual modeling and large language models: impressions from first experiments with ChatGPT. Enterp. Model. Inf. Syst. Archit. (EMISAJ) **3**, 1–15 (2023). https://doi.org/10.18417/EMISA.18.3
11. Fritsch, A., von Hammerstein, J., Schreiber, C., Betz, S., Oberweis, A.: Pathways to greener pastures: research opportunities to integrate life cycle assessment and sustainable business process management based on a systematic tertiary literature review. Sustainability **14**(18), 11164 (2022). https://doi.org/10.3390/su141811164
12. Gambini, M., La Rosa, M., Migliorini, S., Ter Hofstede, A.H.M.: Automated error correction of business process models. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 148–165. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23059-2_14
13. Goossens, A., Claessens, M., Parthoens, C., Vanthienen, J.: Extracting decision dependencies and decision logic from text using deep learning techniques. In: Marrella, A., Weber, B. (eds.) BPM 2021. LNBIP, vol. 436, pp. 349–361. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94343-1_27

14. Honkisz, K., Kluza, K., Wiśniewski, P.: A concept for generating business process models from natural language description. In: Liu, W., Giunchiglia, F., Yang, B. (eds.) KSEM 2018. LNCS (LNAI), vol. 11061, pp. 91–103. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99365-2_8
15. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. EBSE Technical report EBSE-2007-01(2.3) (2007)
16. Li, J., Wang, H.J., Zhang, Z., Zhao, J.L.: A policy-based process mining framework: mining business policy texts for discovering process models. IseB **8**(2), 169–188 (2010). https://doi.org/10.1007/s10257-009-0112-x
17. Maatuk, A.M., Abdelnabi, E.A.: Generating UML use case and activity diagrams using NLP techniques and heuristics rules. In: International Conference on Data Science, E-learning and Information Systems 2021, Ma'an Jordan, pp. 271–277. ACM (2021). https://doi.org/10.1145/3460620.3460768
18. Maqbool, B., et al.: A comprehensive investigation of BPMN models generation from textual requirements—techniques, tools and trends. In: Kim, K.J., Baek, N. (eds.) ICISA 2018. LNEE, vol. 514, pp. 543–557. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-1056-0_54
19. Mendling, L., Thom, L.H., van der Aa, H.: Natural Language Processing with Process Models (NLP4RE Report Paper). REFSQ Workshops 2376(2376) (2019)
20. Mitsyuk, A.A., Lomazova, I.A., Van Der Aalst, W.M.P.: Using event logs for local correction of process models. Autom. Control. Comput. Sci. **51**(7), 709–723 (2017). https://doi.org/10.3103/S0146411617070306
21. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. Data Knowl. Eng. **55**(3), 243–276 (2005). https://doi.org/10.1016/j.datak.2004.12.005
22. Müller, G., Bergmann, R.: Complexity-aware generation of workflows by process-oriented case-based reasoning. In: Kern-Isberner, G., Fürnkranz, J., Thimm, M. (eds.) KI 2017. LNCS (LNAI), vol. 10505, pp. 207–221. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67190-1_16
23. Moreno-Montes de Oca, I., Snoeck, M., Reijers, H.A., Rodríguez-Morffi, A.: A systematic literature review of studies on business process modeling quality. Inf. Softw. Technol. **58**, 187–205 (2015). https://doi.org/10.1016/j.infsof.2014.07.011
24. Schüler, S.: Results SLR Business Process Model Generation (2023). https://doi.org/10.5445/IR/1000158547
25. Smirnov, S., Weidlich, M., Mendling, J., Weske, M.: Action patterns in business process model repositories. Comput. Ind. **63**(2), 98–111 (2012). https://doi.org/10.1016/j.compind.2011.11.001
26. Stahl, T., Völter, M.: Model-Driven Software Development: Technology, Engineering, Management. Hoboken, Wiley (2006)
27. Takei, T., Horita, H.: Towards goal-oriented business process model repair. In: International Congress on Advanced Applied Informatics (IIAI-AAI), Niigata, Japan, pp. 691–696. IEEE (2021). https://doi.org/10.1109/IIAI-AAI53430.2021.00123
28. Tangkawarow, I., Sarno, R., Siahaan, D.: ID2SBVR: a method for extracting business vocabulary and rules from an informal document. Big Data Cogn. Comput. **6**(4), 119 (2022). https://doi.org/10.3390/bdcc6040119
29. Tsakalidis, G., Georgoulakos, K., Paganias, D., Vergidis, K.: An elaborate preprocessing phase (P3) in composition and optimization of business process models. Computation **9**(2), 16 (2021). https://doi.org/10.3390/computation9020016

30. Ullrich, M., et al.: Platform architecture for the diagram assessment domain. In: Proceedings of the Software Engineering 2021 satellite events, vol. 2814. Gesellschaft für Informatik e.V., Braunschweig (2021). https://doi.org/10.22028/D291-34119
31. Wang, N., Sun, S., Liu, Y., Zhang, S.: Business process model abstraction based on fuzzy clustering analysis. Int. J. Coop. Inf. Syst. **28**(03), 1950007 (2019). https://doi.org/10.1142/S0218843019500072
32. Webster, J., Watson, R.T.: Analyzing the past to prepare for the future: writing a literature review. MIS Q. **2**(26), xiii–xxiii (2002)
33. Weske, M.: Business Process Management: Concepts, Languages, Architectures, 3rd edn. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-59432-2

Navigating Event Abstraction in Process Mining: A Comprehensive Analysis of Sub-problems, Data, and Process Characteristic Considerations

Jungeun Lim and Minseok Song(✉)

Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang, South Korea
{je5719,mssong}@postech.ac.kr

Abstract. Process mining technologies assume event logs with an appropriate level of granularity, but many systems generate low-level event logs, resulting in complex process models. Event abstraction addresses this issue by transforming low-level event logs into abstracted event logs, enabling the derivation of business-level process models. However, practitioners often struggle to choose suitable event abstraction methods. This is primarily due to the lack of comparative studies that analyze the differences between methods and the insufficient information regarding the data and relevant process characteristics to be considered. This study conducts a comprehensive literature review on event abstraction to overcome these challenges. The review focuses on summarizing specific sub-problems in event abstraction, identifying types of data that can be utilized, and highlighting important process characteristics that should be considered. The insights and guidance provided by this review will be valuable to practitioners seeking to select and implement effective event abstraction techniques.

Keywords: Event Abstraction · Hierarchical Process Model · Literature Review

1 Introduction

The field of process mining deals with analyzing actual business processes by utilizing event logs [1]. Typically, event logs capture system-level activities rather than easily understandable business-level activities. This poses challenges for process discovery and modeling, as the detailed low-level activities may lead to complex process models that fail to capture the higher-level structure of the process. Event abstraction serves as a solution to address these challenges by identifying and abstracting low-level activities associated with the same high-level activity [2, 3].

While various event abstraction methods have been proposed in research studies, practitioners face challenges in selecting appropriate strategies for their specific situations and understanding the characteristics of their data for event abstraction. Additionally, there is no list of process characteristics to consider in event abstraction and comparative studies to guide the selection of abstraction methods based on process characteristics.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 174–185, 2024.
https://doi.org/10.1007/978-3-031-50974-2_14

In recent years, several relevant literature reviews have been conducted on event abstraction. Diba et al. [4] classified 20 event abstraction studies into clustering, supervised learning, behavioral patterns, and process model-based approach, focusing on technology. The data used in each study and assumptions for the application of technology were also explained, but they were not systematically organized. Marin-Castro & Tello-Leal [5] briefly introduced nine studies, categorizing them as unsupervised and supervised approaches. Zelst et al. [6] conducted an in-depth literature review on event abstraction, presenting seven taxonomies based on existing studies and providing valuable insights for understanding event abstraction from multiple perspectives.

However, existing literature reviews have not fully addressed the aforementioned challenges. Therefore, this study aims to present a conceptual framework that facilitates the understanding and comparison of event abstraction studies for practitioners. This paper is organized as follows. Section 2 defines the sub-problems of event abstraction, and Sect. 3 summarizes the data used in event abstraction research. Section 4 introduces methods for solving each sub-problem, and Sect. 5 defines process characteristics and conducts an analysis to show the importance of considering process characteristics in event abstraction. Finally, we conclude this paper in Sect. 6.

2 Event Abstraction Process

Typically, event abstraction involves several sub-problems that vary depending on specific conditions. This section presents an overview of the event abstraction process described in the literature, along with the individual problems that constitute this process. (For detailed information on the methods proposed to solve each sub-problem, please refer to Sect. 5.) The event abstraction process was designed by reviewing the existing literature that was covered in three review papers related to event abstraction published within five years. In total, 39 papers were considered, and after the abstract screening, 36 papers were selected.

The event abstraction process is summarized and organized like the BPMN model in Fig. 1. If a high-level process model, which represents the flow of activities, is available, solving the process model-based event abstraction problem is sufficient to obtain an abstracted event log. However, additional steps are required in cases where a high-level process model is not accessible. The first step, preprocessing, is optional and applied when the event cycle is too short. The subsequent step is to map an event class to an activity class (EC-AC) or an event instance to an activity class (EI-AC). Solving the EI-AC problem is particularly useful when different activity classes can be associated with the same event classes. Finally, the problem of mapping event instances to activity instances (EI-AI) is addressed to obtain an abstracted event log.

Process Model-Based Event Abstraction. This approach is employed when a high-level process model is available. In this method, studies [3, 7–9] focus on determining the activity class and activity instance of each event instance by considering the control-flow of activities within the high-level process model. If you performed process model-based event abstraction, there is no need to go through additional steps.

Preprocessing. The preprocessing step aims to distinguish event instances. In the literature [10, 11], this step was necessary because data collected from sensors or the web

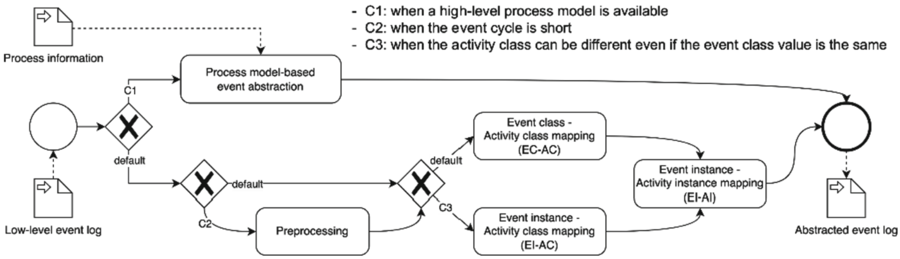

- C1: when a high-level process model is available
- C2: when the event cycle is short
- C3: when the activity class can be different even if the event class value is the same
Process information
Process model-based event abstraction
C1
default
default
C2
Preprocessing
default
C3
Event class - Activity class mapping (EC-AC)
Event instance - Activity instance mapping (EI-AI)
Event instance - Activity class mapping (EI-AC)
Low-level event log
Abstracted event log

Fig. 1. Event abstraction process with a BPMN-like notation.

often have very short intervals, unlike typical event logs. In such cases, multiple data points are treated together as a single event instance to enable subsequent analysis.

Event Class – Activity Class Mapping. The EC-AC problem [12–21] aims to find the mapping rule $\varphi : ec \rightarrow ac$ to identify event classes belonging to the same activity class. An event class *ec* is an attribute representing the event type, such as the event name, when an event is a low-level task. An activity class *ac* is an attribute representing the type of activity, when an activity is a high-level task, consisting of a low-level subprocess. The output of this problem in the literature shows either an n:1 or an n:m relationship between event classes and activity classes. In Fig. 2 (a), a conceptual example is provided: regardless of the attribute values, the activity class is determined solely based on the event class.

Event Instance – Activity Class Mapping. The EI-AC problem [3, 14, 19, 22–31] focuses on finding a mapping rule $\rho : ei \rightarrow ac$. An event instance *ei* is the occurrence of an event and it can be represented by a tuple of attribute values. The attributes representing the event instance do not include the activity class and activity instance identifier. This problem aims to identify patterns of event attributes that belong to the same activity class. The literature demonstrates the output of this problem as an n:1 relationship between attribute tuples and activity classes. In Fig. 2 (b), a conceptual example is provided: the activity class is determined by considering the event class and other attribute values.

Event Instance – Activity Instance Mapping. The EI-AI problem [3, 7–12, 14–28, 30–33] deals with finding a mapping rule $\psi : ei \rightarrow ai$. An activity instance *ai* is a trace of event instances, and attributes representing the event instance include the activity class. Even if multiple event instances share the same activity class values, they may either belong to a single activity instance or multiple activity instances created by performing the same activity multiple times. EI-AI addresses this situation. The literature presents the output of this problem as a 1:1 or an n:1 relationship between attribute tuples and activity instances. In Fig. 2 (c), a conceptual example is provided: the activity instance is determined by considering the event class and other attribute values, including the activity class, which can be derived from either EC-AC or EI-AC mapping results.

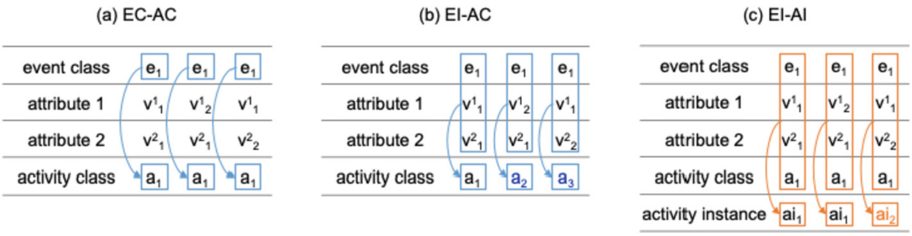


Fig. 2. Event log examples describing EC-AC, EI-AC, and EI-AI problems.

3 Data for Event Abstraction

The data used for event abstraction can be classified into three categories: event class sequence, event attributes, and activity-related data.

3.1 Event Class Sequence

The event class sequence represents a simple event log that considers only the class of events and their chronological order based on timestamps. This data is typically used to analyze the relationship between events, such as directly followed relations and causal relations. Alternatively, the data is transformed into position values, which indicate the relative or absolute position of an event within traces.

Several methods exist for calculating the position value of an event [12, 13, 23]. In the works of Günther et al. [12] and Rehse & Fettke [13], the position value is determined by measuring the distance between two events in the event sequence for all event class pairs. Folino et al. [23] expressed the position of an event as a vector with three values: 1) the distance of the event from the first event in the trace, 2) the length of the trace to which the event belongs compared to the longest trace in the event log, and 3) the distance of the event from the last event in the trace relative to the length of the longest trace.

3.2 Event Attributes

There have been studies using various event attributes. We collected all the attributes from the literature and classified them into four categories: event class attributes, event instance attributes, case attributes, and global attributes (Table 1).

Event class attributes are specific to the event class itself. Examples of event class attributes include the event name and event description [14–16]. Event instance attributes can vary across different event instances, even if they share the same event class. These attributes include resource, time, status, group, and others [3, 10, 11, 14, 15, 22, 23, 26, 27, 32]. Case attributes relate to the specific case or context in which the event occurs [14, 22, 27]. They provide information about the case at the time the event is performed, such as previous tasks performed and the condition of the case. Lastly, global attributes represent external factors that can dynamically change [14]. These attributes capture information that may not be directly associated with the event or case, but can still influence the event, such as case priority or temperature.

Table 1. Four categories of event attributes and their examples.

Event class attributes	event name, event description
Event instance attributes	group, role, resource, time, location, status, support team, functional division, organization line, organization country, activity class, sub-object
Case attributes	preceding events, succeeding events, case's clinical conditions
Global attributes	case priority

3.3 Activity-Related Data

Unlike event sequences and event attributes, activity-related data is often challenging to obtain. This data is typically held by organizations to support the business processes, or it may need to be created based on expert experience. Examples of activity-related data include the high-level process model, low-level subprocess model, activity set, and activity description [7, 8, 14, 15, 26, 27, 32, 34]. As mentioned earlier, when a high-level process model is available, event abstraction can be achieved through process model-based event abstraction. Other activity-related data serves as additional information for solving the EC-AC, EI-AC, and EI-AI problems.

4 Methods for Solving Event Abstraction Sub-problems

In this section, we introduce several studies with data and methods for each sub-problem. We categorize the methods into three classes: heuristic, unsupervised, and supervised approaches. The heuristic approach comprises methods that define abstraction rules based on domain knowledge. These methods rely on expert insights and predefined rules to guide the event abstraction process. The unsupervised approach encompasses methods where abstraction rules are derived from learning the event log without utilizing activity-related data. Lastly, the supervised approach includes methods that are similar to the unsupervised approach, but with the additional utilization of activity-related data.

4.1 Process Model-Based Event Abstraction Methods

The methods to solve process model-based event abstraction problems were primarily categorized as a supervised approach since a process model is a prerequisite for this problem. Mannhardt et al. [3] and Mannhardt & Tax [7] aimed to determine the best activity classes and activity instance identifiers that can effectively explain a given event sequence. These studies utilized both the high-level process model and the low-level subprocess model. Trace alignment techniques were employed to align the event sequence with potential event sequences derived from the models. On the other hand, Ferreira [34] solely relied on the high-level process model. They utilized the EM-algorithm to estimate the optimal alignment between a given event sequence and the high-level process model.

4.2 Preprocessing Methods

There were two studies that performed preprocessing: one study utilized data obtained from sensors [11], and the other study utilized data obtained from the web [10]. Both studies used the naive solution for preprocessing. Eck et al. [11], the user set the duration of each event instance, taking into consideration the sensor's data collection interval. Data points were then classified into event instances based on this duration. Leoni & Dündar [10] set a threshold for the time interval between data points, and data points were divided into different event instances when the time interval exceeded the threshold.

4.3 Event Class – Activity Class Mapping Methods

Unsupervised Approach. Clustering techniques were commonly used in the unsupervised approach to group event classes with similar properties into the same activity class [12, 13, 16]. Various combinations of data, including event position value, event class attributes, and event instance attributes were utilized to represent the properties of each event class. Another technique used in the unsupervised approach was inductive mining [17]. This algorithm was applied to the event class sequence, resulting in the identification of control-flow relationships between event classes in the form of a process tree. Events belonging to the same activity were then defined by grouping the sub-tree within the process tree.

Supervised Approach. Biar et al. [14, 15] employed a technique that is applicable when activity-related data is available. They utilized the name and descriptions of events and activities to calculate their similarity. By identifying pairs with high similarity, they established mappings between events and activities. Additionally, Biar et al. [15] utilized declarative mining, which involved deriving declarative rules from low-level event logs and simulated event logs generated using a high-level process model. These rules helped filter out EC-AC combinations that were considered impossible.

4.4 Event Instance – Activity Class Mapping Methods

Heuristic Approach. In studies employing the heuristic approach, domain experts defined criteria that event attributes must satisfy in order to map an event instance to a specific activity class [8, 14, 22]. For example, the resource attribute can be used as a mapping condition where event instances with the same resource value are mapped to the same activity class, regardless of other attribute values such as event class.

Unsupervised Approach. Fazzinga et al. [25] and Alharbi et al. [24] utilized Hidden Markov Models (HMM). They considered the activity class as a hidden state and, using the sequence of event classes as input data, learned the transition probability between activities and the emission probability of events associated with each activity. Based on the learned transition and emission probability, the activity class values that maximize the probability of the event class sequence were assigned to each event instance.

Folino et al. [23] employed a predictive clustering tree where event instance attributes served as predictors (x) and the relative positions of the events were the target variable (y). This study falls under the unsupervised approach since the relative position value used as the target variable is not the ground truth.

Supervised Approach. Tax et al. [26] addressed the problem by learning a conditional random field (CRF) model using event attributes as predictors (x) and activity classes as the target variable (y). Senderovich et al. [27] utilized the integer linear programming (ILP) by incorporating domain knowledge, such as the activity class set, characteristics of event instances that cannot be mapped to a single activity class, and prior relationships between activities, as constraints. Tello et al. [32] utilized a list of candidate low-level subsequences of activities as additional information. This list was used to calculate the probability of a direct-following relation between event activities. The calculated probability values were then used to divide the event sequence into a set of subsequences, selecting the result with the highest probability. The resulting subsequences were represented using event attributes, and finally, the subsequences were mapped to activity classes using the multivariate time series clustering method.

4.5 Event Instance – Activity Instance Mapping Methods

Naive approaches were most used for solving the EI-AI. In some studies, continuous event instances with the same activity class value were treated as a single activity instance [11, 12, 17, 24, 26]. Otherwise, studies considered each event instance as a separate activity instance [16, 23, 25, 27].

Heuristic Approach. Heuristic approaches were also widely adopted, where the user directly set the conditions that event instances must satisfy to be assigned to an activity instance. Studies introduced several parameters that can be used to set the condition, including the maximum number of event instances that can be assigned to a single activity instance [14, 15]; the maximum duration of an activity instance [10, 11, 14, 15]; the maximum number of event instances with different activity instance IDs occurring between event instances with the same activity instance ID (i.e., the maximum number of interleaved activity instances) [22]; and the maximum duration of the interleaved activity instances [22].

5 Process Property Consideration

Studies tried to perform event abstraction in consideration of characteristics such as interleaving and repeated use of activity in the process. However, there is no systematic summary of the characteristics of the process to be considered, making it difficult to compare studies. In this section, we organize the characteristics of the process and evaluate some methods with some process characteristics to present the need to consider process characteristics.

5.1 Process Properties

The first process property is the structure of a process. The high-level process and low-level subprocess are composed of a combination of sequential, exclusive, parallel, and redo structures (Fig. 3). Depending on the combination of these structures, the event

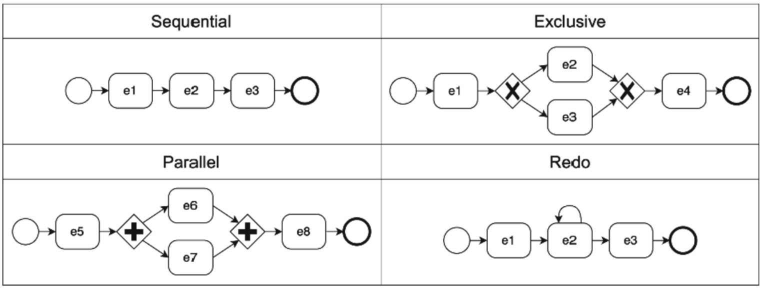


Fig. 3. Basic process structures of processes.

abstraction result may be affected. For example, if the high-level process includes a parallel structure, events required to perform parallel activities can be mixed in order, making it challenging to derive abstraction rules from the event log.

Another important process characteristic is the relationship between low-level and high-level activities (EC:AC). A low-level activity can be associated with only one high-level activity (n:1) or multiple high-level activities (n:m). This relationship between activities can significantly impact the performance of event abstraction, even when the process structure remains the same.

5.2 Analysis Method

We evaluated methods proposed in previous studies [12, 13, 23] that address the EC-AC problem through clustering based on event position values. To generate the data for evaluation, we created virtual process models with all possible process structure combinations where EC:AC = n:1. In total, we generated 16 virtual process models (SS, SE, SP, SR, ES, EE, EP, ER, PS, PE, PP, PR, RS, RE, RP, and RR: the first character represents the high-level structure with its first alphabet, and the second character represents the low-level structure with its first alphabet). We unified the structures of low-level subprocess models for simplicity. For example, the virtual process model PR is like Fig. 4.

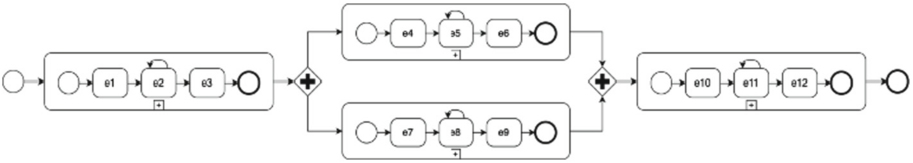


Fig. 4. Virtual process model consisting of a parallel structure at the high-level and a redo structure at the low-level.

We generated 1000 traces for each virtual process model by simulation. The position values were calculated following the formula provided in the literature, and we applied agglomerative clustering. We evaluated the method as effective if events belonging to the same activity were successfully grouped into one cluster. We conducted a qualitative analysis using the agglomerative clustering dendrogram and did not evaluate the result precisely based on strict thresholds. It is important to note that we only assessed the activity corresponding to the selected structure. For example, in the case of the model shown in Fig. 4, we evaluated the clustering result of (e4, e5, e6) and (e7, e8, e9), while (e1, e2, e3) and (e10, e11, e12) were not assessed.

5.3 Analysis Result

Across all three methods, the combinations of SS, SE, and any combinations involving high-level parallel structures (i.e., PS, PE, PP, and PR) did not yield satisfactory clustering results. In contrast, when both the high-level and low-level structures were redo structures, all three methods showed promising outcomes. Table 2 presents the results for the remaining combinations. Among these combinations, [12] and [13] consistently yielded similar results, except for the SP. The method of [23] did not work well in many combinations compared to the other two studies but performed well in the case of SR, where the other two studies struggled. Although this analysis is based on a limited dataset from a simplified process model, it shows the importance of considering process characteristics in event abstraction.

Table 2. Mapping results for each process structure combination.

	SP	SR	ES	EE	EP	ER	RS	RE	RP	RR
[12]	o	x	o	x	o	o	o	o	o	o
[13]	x	x	o	o	o	o	o	o	o	o
[23]	o	o	x	x	x	x	x	x	x	o

6 Discussion and Conclusion

Event abstraction plays a crucial role in converting low-level event logs, which hinders effective business process analysis. Therefore, it is a vital problem that must be addressed to achieve business process intelligence. In this study, we provide a comprehensive summary of the event abstraction process, sub-problems, data, and process characteristic considerations to facilitate the understanding and practical application of event abstraction studies. The insights presented in this study can be leveraged and further developed in several ways.

First, combining diverse methods to solve sub-problems allows us to explore possibilities for obtaining better event abstraction outcomes. Each method handles different process characteristics, so combining them holds the potential for improving event abstraction results. Furthermore, there is an opportunity to automate event abstraction. Optimal results can be derived by automatically selecting the most suitable event abstraction process and method for a given dataset. The evaluation method for event abstraction plays a crucial role in achieving automated outcomes. Although this aspect is not covered in this study, it is necessary for further investigation to enable automated event abstraction.

Second, comparing data and techniques across different approaches can lead to new research directions. For example, the heuristic approach, which defines rules based on diverse data, can inspire investigations into unsupervised or supervised approaches.

Third, regarding the process characteristics, this study conducted only small-scale experiments that show the necessity of considering these factors. To facilitate the comparison and evaluation of event abstraction methods, generating datasets with the suggested process characteristic outlined in this study, varying in size and conditions would be beneficial.

Finally, the recent review included in this study is from 2021, as the literature was selected from existing review papers. We need to analyze more recent literature using the framework introduced in this research.

Acknowledgment. This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2018-0-01441) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) and by a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (grant number: HI23C0061).

References

1. Van der Aalst, W.: Process mining: data science in action (2016)
2. Li, C.Y., Van Zelst, S.J., Van Der Aalst, W.M.P.: An activity instance based hierarchical framework for event abstraction. In: La Rosa, M., Loos, P., Pastor, O. (eds.) Business Process Management. BPM 2016. LNCS, vol. 9850, pp. 160–167. Springer, Cham (2021). https://doi.org/10.1007/978-3-319-45348-4_8
3. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: From low-level events to activities - a pattern-based approach. In: La Rosa, M., Loos, P., Pastor, O. (eds.) Business Process Management. BPM 2016. LNCS, vol. 9850, pp. 125–141. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_8
4. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **10**, 1–24 (2020)
5. Marin-Castro, H.M., Tello-Leal, E.: Event log preprocessing for process mining: a review. Appl. Sci. (Switzerland). **11**, 1–29 (2021)
6. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. Granul. Comput. **6**, 719–736 (2021)
7. Mannhardt, F., Tax, N.: Unsupervised event abstraction using pattern abstraction and local process models. arXiv preprint arXiv:1704.03520 (2017)
8. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: Guided process discovery – a pattern-based approach. Inf. Syst. **76**, 1–18 (2018)
9. Ferreira, D.R., Szimanski, F., Ralha, C.G.: Improving process models by mining mappings of low-level events to high-level activities. J. Intell. Inf. Syst. **43**, 379–407 (2014)

10. De Leoni, M., Dündar, S.: Event-log abstraction using batch session identification and clustering. In: Proceedings of the ACM Symposium on Applied Computing, pp. 36–44 (2020)
11. Van Eck, M.L., Sidorova, N., Van Der Aalst, W.M.P.: Enabling process mining on sensor data from smart products. In: Proceedings - International Conference on Research Challenges in Information Science. 2016-Augus, (2016)
12. Günther, C.W., Rozinat, A., Van Der Aalst, W.M.P.: Activity mining by global trace segmentation. In: Daniel, F., Sheng, Q., Motahari, H. (eds.) Business Process Management Workshops. BPM 2018. LNBIP, vol. 342, pp. 128–139. Springer, Cham (2010). https://doi.org/10.1007/978-3-030-11641-5_1
13. Rehse, J.-R., Fettke, P.: Clustering business process activities for identifying reference model components. In: Daniel, F., Sheng, Q., Motahari, H. (eds.) Business Process Management Workshops. BPM 2018. LNBIP, vol. 342, pp. 5–17. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11641-5_1
14. Baier, T., Mendling, J., Weske, M.: Bridging abstraction layers in process mining. Inf. Syst. **46**, 123–139 (2014)
15. Baier, T., Di Ciccio, C., Mendling, J., Weske, M.: Matching events and activities by integrating behavioral aspects and label analysis. Softw. Syst. Model. **17**, 573–598 (2018)
16. Sánchez-Charles, D., Carmona, J., Muntés-Mulero, V., Solé, M.: Reducing event variability in logs by clustering of word embeddings. In: Teniente, E., Weidlich, M. (eds.) Business Process Management Workshops. BPM 2017. LNBIP, vol. 308. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74030-0_14
17. Andritsos, P.: CJM-ab : Abstracting Customer Journey, vol. 1, pp. 49–56 (2018)
18. Richetti, P.H.P., Baião, F.A., Santoro, F.M.: Declarative process mining: reducing discovered models complexity by pre-processing event logs. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) Business Process Management. BPM 2014. LNCS, vol. 8659, pp. 400–407. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10172-9_28
19. Brzychczy, E., Trzcionkowska, A.: Process-oriented approach for analysis of sensor data from longwall monitoring system. Adv. Intell. Syst. Comput. **835**, 611–621 (2019)
20. Van Dongen, B.F., Adriansyah, A.: Process mining: fuzzy clustering and performance visualization. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) Business Process Management Workshops. BPM 2009. LNBIP, vol. 43, PP. 158–169. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_15
21. Begicheva, A.A., Lomazov, I.A.: Discovering high-level process models from event logs. Model. Anal. Inf. Syst. **24**, 125–140 (2017)
22. Leonardi, G., Striani, M., Quaglini, S., Cavallini, A., Montani, S.B.: Towards semantic process mining through knowledge-based trace abstraction. In: Ceravolo, P., van Keulen, M., Stoffel, K. (eds.) Data-Driven Process Discovery and Analysis. SIMPDA 2017. LNBIP, vol. 340, pp. 45–64. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11638-5_3
23. Folino, F., Guarascio, M., Pontieri, L.: Mining multi-variant process models from low-level logs. In: Abramowicz, W. (eds.) Business Information Systems. BIS 2015. LNBIP, vol. 208, pp. 165–177. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19027-3_14
24. Alharbi, A., Bulpitt, A., Johnson, O.: Towards unsupervised detection of process models in healthcare. In: Ugon, A., Karlsson, D., Dlein, gunnar O., and Moen, A. (eds.) Building Continents of Knowledge in Oceans of Data: The Future of Co-Create eHealth, pp. 381–385. IOS Press (2018)
25. Fazzinga, B., Flesca, S., Furfaro, F., Pontieri, L.: Process discovery from low-level event logs. In: Krogstie, J., Reijers, H. (eds.) Advanced Information Systems Engineering. CAiSE 2018. LNCS, vol. 10816, PP. 257–273. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91563-0_16
26. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.: Mining process model descriptions of daily life through event abstraction. Stud. Comput. Intell. **751**, 83–104 (2018)

27. Senderovich, A., Rogge-Solti, A., Gal, A., Mendling, J., Mandelbaum, A.: The ROAD from sensor data to process instances via interaction mining. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) Advanced Information Systems Engineering. CAiSE 2016. LNCS, vol. 9694, pp. 257–273. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39696-5_16
28. Fazzinga, B., Flesca, S., Furfaro, F., Masciari, E., Pontieri, L.: Efficiently interpreting traces of low level events in business process logs. Inf. Syst. **73**, 1–24 (2018)
29. Bose, R.P.J.C., Maggi, F.M., Van Der Aalst, W.M.P.: Enhancing declare maps based on event correlations. In: Daniel, F., Wang, J., Weber, B. (eds.) Business Process Management. LNCS, vol. 8094, pp. 97–112. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_9
30. Li, J., Bose, R.P.J.C., Van Der Aalst, W.M.P.: Mining context-dependent and interactive business process maps using execution patterns. In: Zur Muehlen, M., Su, J. (eds.) Business Process Management Workshops. BPM 2010. LNBIP, vol. 66, pp. 109–121. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20511-8_10
31. Folino, F., Guarascio, M., Pontieri, L.: Mining predictive process models out of low-level multidimensional logs. In: Jarke, M., et al. (eds.) Advanced Information Systems Engineering. CAiSE 2014. LNCS, vol. 8484, pp. 533–547. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07881-6_36
32. Tello, G., Gianini, G., Mizouni, R., Damiani, E.: Machine learning-based framework for log-lifting in business process. In: Hildebrandt, T., van Dongen, B., Röglinger, M., Mendling, J. (eds.) Business Process Management. BPM 2019. LNCS, vol. 11675. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_16
33. Sun, Y., Bauer, B.: A graph and trace clustering-based approach for abstracting mined business process models. In: ICEIS 2016 - Proceedings of the 18th International Conference on Enterprise Information Systems, vol. 1, pp. 63–74 (2016)
34. Ferreira, D.R., Szimanski, F., Ralha, C.G.: Mining the low-level behaviour of agents in high-level business processes. Int. J. Bus. Process. Integr. Manag. **6**, 146–166 (2013)

Timed Alignments with Mixed Moves

Neha Rino[1] and Thomas Chatain[2](✉)

[1] IRIF, ENS Paris-Saclay, CNRS, Université Paris-Saclay, Gif-sur-Yvette, France
`neha.rino@ens-paris-saclay.fr`
[2] LMF, ENS Paris-Saclay, CNRS, Université Paris-Saclay,
Inria, Gif-sur-Yvette, France
`thomas.chatain@ens-paris-saclay.fr`

Abstract. We study conformance checking for timed models, that is, process models that consider both the sequence of events that occur, as well as the timestamps at which each event is recorded. Time-aware process mining is a growing subfield of research, and as tools that seek to discover timing-related properties in processes develop, so does the need for conformance-checking techniques that can tackle time constraints and provide insightful quality measures for time-aware process models. One of the most useful conformance artefacts is the alignment, that is, finding the minimal changes necessary to correct a new observation to conform to a process model. In this paper, we solve the timed alignment problem where the metrics used to compare timed processes allow weighted mixed moves, i.e. an error on the timestamp of an event may or may not propagate to its successors, and we provide linear time algorithms for a large class of such weighted mixed distances, both for distance computation and alignment on models with sequential causal processes.

Keywords: Conformance checking · Alignments · Timestamps · Time Petri nets

1 Introduction

Conformance checking is the task of evaluating the quality of the discovered process models with respect to event logs, i.e. to determine how well a process model represents a system observed via the observed traces.

A central notion in conformance checking is the notion of an alignment [2], or, the minimal series of edits, usually insertions or deletions, that transform an observed trace into a process trace. Alignments thereby help pinpoint exactly where inevitable deviations from expected behaviour occur, and the more distant the aligning word of a model is to its observed trace, the worse the model is at reflecting real system behaviour.

Assuming the event logs are a list of words over a finite alphabet (the set of possible discrete events), the problem of calculating the alignment has been

---

Neha Rino was funded by the International Master's Scholarships Program IDEX of Université Paris-Saclay.

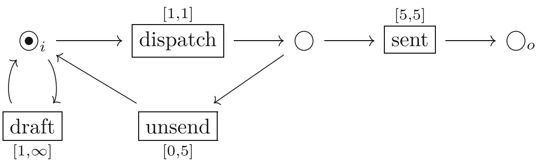
© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 186–197, 2024.
https://doi.org/10.1007/978-3-031-50974-2_15

extensively studied [2,4], usually using either Hamming distance or Levenshtein's edit distance. One can study process mining for explicitly timed systems [1,9,14], as by considering events along with their timestamps when mining processes, we can glean information about the minimum delay between two events, the maximum duration the system takes to converge upon a state, or check deadlines, all of which are highly relevant in real-world applications [8,10,12].

In order to study alignments for time-aware conformance checking, one requires a notion of distance over timed words, enabling us to measure how close two timed process traces are. Distances over timed words have been explored before, such as in [3,11] and [7], which capture differences between the timestamps attached to the same action label but do not link it to model events. Here we use and generalise the distances defined in [6]. As for process models, there are ways to use existing process mining notation to denote time constraints. BPMN 2.0 comes equipped with *timer events*. For our purposes, we use time Petri nets, an extension of Petri nets equipped with the ability to express constraints on the duration of time between an action being enabled, and its actual occurrence. In particular, in this paper, we restrict ourselves to time Petri nets with no branching points.

As time-aware process mining grows popular, new quality measures and conformance-checking techniques must be developed that are sensitive to temporal constraints, but so far in the study of alignments, we notice that the process model used is rarely time-aware. This paper seeks to investigate a class of timed distances that we call weighted mixed distance functions or d_α, parametrised by α, the ratio of different costs of types of timed edit moves. In doing so, we generalise the three distances defined in [6] and in this paper solve the alignment problem for a much more general class of metrics. These generalised weighted mixed distances allow users to decide what types of corrections in the observed behaviour should cost more to fix, as sometimes changing an event locally is the more viable option, while in other use cases, it might be more likely that shifts in local events necessary cause consequent events to reflect the same shift.

Example 1. Consider a model of composing and sending an email, where the intervals signify allowed durations between the current event and its immediate predecessor, depicted below with input and output places marked with an i and an o:



One process trace is (draft, 5)(dispatch, 6)(sent, 11), which depicts drafting the message in 5 units and then dispatching it, and having it send successfully. An example of an observed trace that does not conform to the process above would be (draft, 3)(draft, 5)(dispatch, 7)(dispatch, 7)(sent, 12). Clearly, there is an extraneous letter here, as the same message cannot be dispatched twice, and the timestamps for dispatch and sending come too late. One possible optimally close process trace for this observation is (draft, 3)(draft, 6)(dispatch, 7)(sent, 12), which deletes the extra dispatch event and extends the second draft event by a unit, allowing the rest of the trace to thereby be in time.

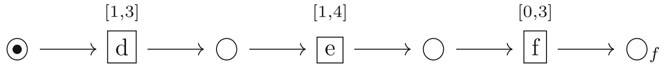
Here we recall the general alignment problem, posed in [6], which formalised the notion of aligning timed traces to timed process models.

Definition 1 (The General Alignment Problem). *Given a process model* N *denoted by a time Petri Net and an observed timed trace* σ *we wish to find a timed word* $\gamma \in \mathcal{L}(N)$ *such that* $d(\sigma, \gamma) = \min_{x \in \mathcal{L}(N)} d(\sigma, x)$ *for some distance function* d *on timed words.*

In this paper, we restrict ourselves to the study of the alignment problem to a specific class of time Petri nets, ones that we define as sequential process models. These lose the ability to express concurrent events the way time Petri nets can but retain the ability to reason about time constraints on the durations between any two consecutive events. The last assumption we make is that the untimed part of the observed trace conforms to the process, i.e., the only part that requires aligning is the timestamps. This allows us to focus on the timing aspect of the problem, although once this is solved we can easily adapt existing untimed alignment methods like those described in [2] or [4] to align words with both action labels and timestamps that need editing. This brings us to the following formal problem:

Definition 2 (The Purely Timed Alignment Problem for Sequential Process Models). *Given a sequential process model* N *denoted by a time Petri Net and an observed timed trace* (w, σ), *such that* $w \in Untime(\mathcal{L}(N))$ *we wish to find a valid timestamp sequence* γ *such that* $d(\sigma, \gamma) = \min_{x \in \mathcal{L}(N)} d(\sigma, x)$.

Example 2. Consider the process model N_1:



Say we observed the trace $(u_1, \sigma_1) = (d, 4)(e, 6)(f, 6)$ that did not fit the model, and we wished to analyse how best to modify them to fit them back into the model. The untimed part is already aligned, and in order to make the timestamps fit we can edit the timestamp of d to give $(v_1, \gamma_1) = (d, 3)(e, 6)(f, 6)$.

In Sects. 2 and 3.2 of this paper we will focus on the problem of computing a particular case of our mixed distances when costs are equal, d_1 and in Sect. 4, we return to the problem of aligning sequential process models to observed traces. In Subsect. 3.3 we also show that computing and aligning d_α for $\alpha \geq 2$ can be reduced to the delay only case, which was solved in [6]. We present linear time algorithms for both computing d_1 and solving the alignment problem in this setting. The full proofs for all of the results stated here can be found in [13].

2 Preliminaries: Edit Moves for Timed Words

Inspired by Levenshtein's edit distance, we view the definition of these distances as an exercise in cost minimisation over some set of transformations between words. We define *moves* as functions that map one time sequence to another. What are useful notions of transformation on a timed system?

Example 3. Say we seek to align the timed trace $(4, 6, 6)$ with $(3, 6, 6)$. It feels reasonable to say the distance between the two is 1. A way to arrive at this conclusion is that if the first timestamp were just shifted to fire at 3 instead, the whole process would match. This sort of local, almost typographical error can often happen in systems, and it is the simplest kind to fix.

Consider aligning the timed traces $(4, 8, 11)$ and $(3, 7, 10)$. When trying to compare these firing sequences, we can view it as before, as all of the timestamps firing later than they should, and so editing one by one, getting an aligning cost of 3. However, this cascading chain of errors can also be fixed if the first event is moved back to fire at 3, and all the relative relationships between it and its successors are preserved. This views the second two tasks as being dependent on when the first ends, which makes sense because they are causally linked in the model. Hence the switch from the timestamp series $(4, 8, 11)$ to $(3, 7, 10)$ can be viewed as only a cost 1 edit. This viewpoint is of practical use as if a delay at the beginning caused the whole process to occur too late, the important thing to fix is the delay at the beginning and the rest of the process will follow suit.

Based on the above example, we naturally arrived at two types of moves in [6]. We define a *stamp move* as a move that translates the timing function only at a point, i.e., that edits a particular element of the timestamp series τ. Calculating d_t between two traces can be viewed as a cost minimisation process in aligning the traces using only stamp moves.

Definition 3 (Stamp Move). *Given a timing function* $\gamma : \{1, \ldots n\} \to \mathbb{R}$*, formally, we define this as:*
$\forall x \in \mathbb{R}, i \le n : stamp(\gamma, x, i) = \gamma'$ *where*

$$\forall i \le n : \gamma'(j) = \begin{cases} \gamma(j) + x & j = i \\ \gamma(j) & otherwise \end{cases}$$

We define the cost of such a stamp move as the stamp cost per unit (c_s) times the magnitude of the move, i.e., $c_s \cdot |x|$.

The next type of move we describe is the more interesting *delay move*. By formulating this type of edit move, we seek to leverage the structure of the process model itself, by reflecting the causal relationships between events, yielding:

Definition 4 (Delay Move). *Given a timing function* $\gamma : \{1, \ldots n\} \to \mathbb{R}$*, formally, we define this as:*
$\forall x \in \mathbb{R}, i \le n : delay(\gamma, x, i) = \gamma'$ *where*

$$\forall i \le n : \gamma'(j) = \begin{cases} \gamma(j) + x & j \ge i \\ \gamma(j) & otherwise \end{cases}$$

We define the cost of this delay move as the delay cost per unit (c_d) times the magnitude of the move, i.e., $c_d \cdot |x|$. The cost of a sequence of moves is the sum of the costs of the moves. Armed with these definitions, up to scaling, all the distances we can build from these depend solely upon the ratio of stamp cost to delay cost, giving:

Definition 5 (Weighted mixed distance: d_α). *Given $c_s, c_d \in \mathbb{R}^+ \cup \{\infty\}$ and assuming that we always have $c_d \cdot c_s \neq 0$, for any two timing functions τ_1, τ_2 over the same causal process (CN, p) and $\alpha \in \mathbb{R}^+ \cup \{\infty\}$, we define the α weighted mixed distance d_α as follows*

$$d_\alpha(\tau_1, \tau_2) = \min\{cost(m) | \frac{c_s}{c_d} = \alpha \textit{ and } m \in (Stamp \cup Delay)^*, m(\tau_1) = \tau_2\}$$

From this definition, we can recover the stamp only (d_t), delay only (d_θ) and evenly mixed (d_N) distances defined in [6] as natural special cases, where we discount a type of move from the possible move set conceptually by setting its cost to ∞:

$$\begin{aligned}
d_0(\tau_1, \tau_2) &= d_t(\tau_1, \tau_2) = \min\{cost(m) | m \in Stamp^*, m(\tau_1) = \tau_2\} \\
d_1(\tau_1, \tau_2) &= \min\{cost(m) | m \in (Stamp \cup Delay)^*, m(\tau_1) = \tau_2\} \\
d_\infty(\tau_1, \tau_2) &= d_\theta(\tau_1, \tau_2) = \min\{cost(m) | m \in Delay^*, m(\tau_1) = \tau_2\}
\end{aligned}$$

Henceforward for consistency we will use d_0, d_1, and d_∞.

Example 4. Let us try to align the observed trace $(0, 3, 4)$ to the process trace $(0.5, 2.5, 3.5)$

The best d_0 alignment has cost 1.5, with stamp moves editing each position, so 0.5 cost for each of three edits.

The best d_∞ alignment is cost 1.5, as it requires a 0.5 delay edit at the first place to push the 0 forward to a 0.5, and then another -1 delay at the second position to pull the rest of the trace back from $(3.5, 4.5)$ back to $(2.5, 3.5)$.

Now, d_1. In sequential models, the cascading effect of a delay is not affected by the order of moves, we can align timestamps left to right. Our first move must incur a minimum cost of 0.5 as we try to align $(0, 3, 4)$ to our process trace at the first position. If any part of this initial move was a delay, we would push the second component even further away from 2.5 than it already was, and if any part of the delay were negative, then it seems to counter the effect of the stamp part, so we should only use stamp here. For move two, we again incur a minimum cost of 0.5. If the stamp part of this move were positive, it would leave the third component further off from the goal 3.5 than if the whole move used delay, and so a pure delay move seems best. This gives a cost 1 alignment, $(0.5, 0, 1)(0, 0.5, 2)$. All the intuitive reasoning given above will be justified in subsequent sections, and in fact, distance 1 is the best we can do here even in the mixed case.

3 Computing Mixed Distances

3.1 Notation and Setup

Given the nonconstructive nature of the definition of d_α, it is not clear how one can efficiently calculate the distance between two fixed timed traces, as a minimal cost sequence of moves is not obvious. Before we propose an algorithm that does calculate the minimal cost for sequential timed traces, we first define a few properties that seem to characterise classes of well-formed minimal cost runs, and then prove that these properties both improve the cost and are satisfied solely by the run calculated by the algorithm we provide. We start with some convenient notation for a common combination of the previously defined moves.

Definition 6 (Mixed Move). *We define mixed moves, which denote doing a stamp move and a delay move at the same position in the word. We define their effect as:* $(s, d, e)(\gamma) = stamp(delay(\gamma, d, e), s, e)$ *Let the set of all mixed moves be* $Moves$*.*

Given any move $m \in Moves$ we define the function $cost_\alpha : Moves \rightarrow \mathbb{R}^+$ that returns the cost of the move. The cost of a mixed move (s, d, e) is the sum of the cost of the stamp and delay moves it's made up of, $c_s \cdot |s| + c_d \cdot |d|$. By default by $cost((s, d, e))$ we are refering to the evenly mixed cost $cost_1((s, d, e)) = |s| + |d|$. We say a sequence of moves $m_1 m_2 \ldots m_n = m \in Moves^*$ *aligns* τ_1 to τ_2 if $m(\tau_1) = m_n(\ldots(m_2(m_1(\tau_1)) \ldots) = \tau_2$.

Now, we look at a new way to represent timing functions. There is of course the standard definition, $\tau : \{1, \ldots n\} \rightarrow \mathbb{R}^+$ that assigns to each event a timestamp that records exactly when the event occurs. Instead, thinking along the lines of delays and durations between events occurring, we define a way by which to view a timed word not in terms of its absolute timestamps, but by the delays between relevant timestamps.

Definition 7 (Flow Function). *Given a (not necessarily valid) time sequence,* $\tau : \{1, \ldots n\} \rightarrow \mathbb{R}^+$*, we first define the flow function of* τ*,* $f_\tau : \{1, \ldots n\} \rightarrow \mathbb{R}^+$ *such that*

$$f_\tau(i) = \begin{cases} \tau(i) & i = 1 \\ \tau(i) - \tau(i-1) & i > 1 \end{cases}$$

The flow function thereby is a dual representation of timing functions, much like the relationship between graphs and line graphs.

Example 5. Consider the linear timed trace $w = (1, 5, 9)$. The flow function measuring its successive delays is $(1, 4, 4)$.

As defined, f_τ produces exactly the time durations that the guards of each transition in the model check, i.e. the clock function values during the run. Hence, we see that if a word is in the language of the model, then its f function maps events to values that lie within the constraint that the event's corresponding transition demands. Also note that given the underlying causal process and the resulting f_τ, we can reconstruct τ quite straightforwardly as $\forall i \leq n : \tau(i) = \sum_{j \leq i} f_\tau(j)$.

The following lemma characterises how moves look in flow function notation, highlighting how they are both relatively "local" in effect now:

Lemma 1. *Given a flow function f_{τ_1} and a mixed move (s, d, i), the consequence of performing the move is the flow function defined below, where $\tau_2 = (s, d, i)\tau_1$.*

$$f_{\tau_2}(j) = \begin{cases} f_{\tau_1}(j) + s + d & j = i \\ f_{\tau_1}(j) - s & j = i - 1 \end{cases}$$

In order to compute the mixed distance between two timed traces, we wish to cull down the space of possible sequences of moves that transform the two words to each other to some move sequences that still capture all the behaviours we're interested in. Hence, we introduce the following properties, which are designed intuitively to be sequences of moves that still perform the same effective transformations, and generally lower the cost of the transformation as well, by choosing moves wisely to avoid inefficiency. Chronology, the first property, represents the fact that moves can commute in a sequence. This holds for all d_α, irrespective of α. The second, co-operation, represents the fact that doing an unnecessarily expensive move in the moment will not give unintuitive future gains, and we can show this holds for all d_α where $\alpha \geq 2$. The final property, stability, pinpoints exactly at what point the optimal move for the evenly mixed distance, d_1 is.

Definition 8 (Chronology). *A sequence of moves aligning γ to σ is said to be chronological if for all positions $i < j \leq n = |\gamma|$, all the moves at position i are performed before any move at position j and exactly one mixed move (where one or both components may be zero) takes place at each position, that is $\rho \in Moves^*$ is chronological iff $\forall i \in \{1, 2, \ldots n\}, \exists s_i, d_i \in \mathbb{R} : \rho = (s_1, d_1, 1)(s_2, d_2, 2) \ldots (s_n, d_n, n)$. Such that $\forall i \in \{1, 2, \ldots n\} : \sigma_i = \gamma_i + s_i + \sum_{j=1}^{i} d_j$.*

Definition 9 (Reverse Chronology). *A sequence of moves aligning γ to σ is said to be reverse chronological if for all positions $i < j \leq n = |\gamma|$, all the moves at position i are performed after any move at position j and exactly one mixed move (where one or both components may be zero) takes place at each position, that is $\rho \in Moves^*$ is chronological iff $\forall i \in \{1, 2, \ldots n\}, \exists s_i, d_i \in \mathbb{R} : \rho = (s_n, d_n, n)(s_{n-1}, d_{n-1}, n-1) \ldots (s_1, d_1, 1)$. Such that $\forall i \in \{1, 2, \ldots n\} : \sigma_i = \gamma_i + s_i + \sum_{j=1}^{i} d_j$.*

Example 6. Given $\gamma = (1, 1, 2, 4, 5)$ and $\sigma = (1, 2, 2.5, 4.2, 5)$, an example of a cost 5.3 non-chronological sequence of evenly mixed moves aligning them is as shown: $\gamma \xmapsto{(-1,0,1)} (0, 1, 2, 4, 5) \xmapsto{(0,2,1)} (2, 3, 4, 6, 7) \xmapsto{(0,-1,1)} (1, 2, 3, 5, 6) \xmapsto{(0.3,-0.8,3)} (1, 2, 2.5, 4.2, 5.2) \xmapsto{(0,-0.2,5)} \sigma$

There is an improved cost 3.3 reverse chronological sequence of moves obtained by reordering the above sequence and combining moves at the same position: $\gamma \xmapsto{(0,-0.2,5)} (1,1,2,4,4.8) \xmapsto{(0,0,4)} (1,1,2,4,4.8) \xmapsto{(0.3,-0.8,3)} (1,1,1.5,3.2,4) \xmapsto{(0,0,2)} (1,1,1.5,3.2,4) \xmapsto{(-1,1,1)} \sigma$.

Definition 10 (Co-operation). *A mixed move is said to be co-operative if its delay edit and stamp edit are in the same direction, that is, (s,d,i) is co-operative iff $sd \geq 0$.*

A chronological (or reverse chronological) sequence of moves such as $m = m_0 m_1 \dots m_{n-1}$ aligning γ to σ is said to be co-operative if each of its moves is co-operative.

Example 7. The previous example (that was a cost 3.3 evenly mixed run, recall) can hence be further improved to become a reverse chronological co-operative cost 1.7 run as follows (on flow vector): $f_\gamma \xmapsto{(0,0,5)} (1,0,1,2,1) \xmapsto{(0.2,0,4)} (1,0,1,2.2,0.8) \xmapsto{(0.5,0,3)} (1,0,1.5,1.7,0.8) \xmapsto{(1,0,2)} (1,1,0.5,1.7,0.8) \xmapsto{(0,0,1)} f_\sigma$.

Once again, one can always convert an evenly mixed non-cooperative run into an equivalent cooperative one with a smaller cost, and in fact, this holds for a large number d_α as well.

For the rest of this subsection assume we are always aligning using evenly mixed distance d_1.

The notion of co-operation and reverse chronology leaves only one degree of freedom so to speak in our choice of aligning moves, that is, the ratio of stamp to delay at any given position, and this optimal ratio is what stability pinpoints.

Definition 11 (Stability). *In a reverse chronological, co-operative sequence of moves, let $m_i = (s,d,i)$ be a co-operative move seeking to correct the partially aligned flow function $f_\gamma = (f_\gamma(1), \dots f_\gamma(i), f_\gamma(i+1), f_\sigma(i+2) \dots f_\sigma(n))$ (the result of the run of m all the way up to and including the last stamp move) to $f_\sigma = (f_\sigma(1), f_\sigma(2), \dots f_\sigma(n))$.*

Let $e_i = f_\sigma(i) - f_\gamma(i)$, and $e_{i+1} = f_\sigma(i+1) - f_\gamma(i+1)$.

We say m_i is stable if

$$s = \begin{cases} 0 & e_i \cdot e_{i+1} \geq 0 \\ e_i & \textit{Else If } |e_i| < |e_{i+1}| \\ -e_{i+1} & \textit{Otherwise} \end{cases}$$

A co-operative sequence of moves is said to be *stable* if each of its moves is stable.

Example 8. The earlier cost 1.7 example can be even further improved to be stable and have cost 1.5, thereby giving us the following: $f_\gamma \xmapsto{(0,-0.2,5)} (1,0,1,2,0.8) \xmapsto{(0,-0.3,4)} (1,0,1,1.7,0.8) \xmapsto{(0,0,3)} (1,0,1,1.7,0.8) \xmapsto{(0.5,0.5,2)} (1,1,0.5,1.7,0.8) \xmapsto{(0,0,1)} \sigma$

As the running example suggests, a stable chronological run (or reverse chronological run) is unique, but moreover, we claim that stability, co-operation, and chronology improve the cost of the sequence of moves.

Lemma 2. *There is a stable minimal cost sequence of moves aligning any two sequential timed traces.*

Algorithm 1. d_1 Computation Algorithm

```
Input: $\sigma, \gamma$
Output: $d_1(\gamma, \sigma)$
$cost \leftarrow 0$
$i \leftarrow n$
while $i > 1$ do
    $a \leftarrow f_\sigma(i) - f_\gamma(i)$
    $b \leftarrow f_\sigma(i-1) - f_\gamma(i-1)$
    if $a \cdot b \geq 0$ then                                  $\triangleright$ $(0, a, i)$
        $\gamma \leftarrow (0, a, i)\gamma$
    else if $|a| < |b|$ then                                   $\triangleright$ $(-a, 0, i-1)(0, 0, i)$
        $\gamma \leftarrow (-a, 0, i-1)\gamma$
    else                                                       $\triangleright$ $(b, 0, i-1)(0, a-b, i)$
        $\gamma \leftarrow (b, 0, i-1)(0, a-b, i)\gamma$
    end if
    $cost \leftarrow cost + |a|$
    $i \leftarrow i - 1$
end while
$cost \leftarrow cost + |\gamma_1 - \sigma_1|$                 $\triangleright$ $(0, |\sigma_1 - \gamma_1|, 1)$
```

3.2 Computing the Evenly Mixed Distance d_1 on Traces

Lemma 3. *Given two time sequences* $\sigma = (\sigma_1, \sigma_2, \ldots \sigma_n)$ *and* $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ *with sequential underlying causal processes, the sequence of moves the above algorithm calculates (m) corresponds to the unique stable sequence of moves that aligns* σ *to* γ*.*

Theorem 1. *Algorithm 1 is correct, that is, its result (cost)* $= d_1(\gamma, \sigma)$

3.3 Computing and Aligning d_α for $\alpha \geq 2$

A surprising fact is that for this entire class of distances, we get our distance computation and alignment algorithms almost for free. Since both chronology and co-operation improve cost when computing d_α, whenever the cost of the stamp move is at least twice as much as the cost of the delay move, any sequence of α weighted mixed moves can be converted to one with lesser or equal cost. In fact, in this case, any sequence of such moves can be converted to a delay-only sequence of moves with the cost improving at most. Hence, assuming $c_d = 1$ and $c_s = \alpha$ (which up to scaling changes nothing), we have the following lemma:

Lemma 4. *When $\alpha \geq 2$ for any sequence of α weighted mixed moves aligning any two sequential timed traces, there is a sequence of delay only moves that aligns the same two traces, and has at most equal cost.*

Algorithm 2. d_1 Alignment Algorithm

```
Input: $\sigma$, $N$
Output: $\gamma = \text{argmin}_{x \in \mathcal{L}(N)} d_1(x, \sigma)$
for $i <\in \{1, \ldots, n\}$ do
    $(a, b) \leftarrow (Eft(t_i), Lft(t_i))$
    $f_\gamma(i) = \text{argmin}_{x \in [a,b]} |x - f_\sigma(i)|$
    $i{+}{+}$
end for
```

As a consequence of this lemma, both distance computation and alignment for this large class of d_α's can be done using the naive greedy algorithm described in [6], which runs in linear time.

4 Purely Timed Alignment Under d_1 for Sequential Models

Definition 12 (Sequential Process Models). *We define a sequential process model N of length n to be a sequence of intervals $\{[a_i, b_i] | a_i \in \mathbb{R}, b_i \in \mathbb{R} \cup \{\infty\}, i \leq n\}$. In addition, we define its language $\mathcal{L}(N)$ as follows: $\{(t_1, \ldots t_n) | \forall i \leq n : t_i - t_{i-1} \in [a_i, b_i]\}$ where $t_0 = 0$.*

We depict them as follows: $\odot \longrightarrow \underset{[a_1,b_1]}{\square} \longrightarrow \bigcirc \longrightarrow \underset{[a_2,b_2]}{\square} \ldots \underset{[a_n,b_n]}{\square} \longrightarrow \bigcirc$.

Example 9. Consider the below underlying sequential process model and a new observed trace $\sigma = (3, 4, 5) \notin \mathcal{L}(N)$.

$$\odot \longrightarrow \underset{[0,1]}{\square} \longrightarrow \bigcirc \longrightarrow \underset{[2,2]}{\square} \longrightarrow \bigcirc \longrightarrow \underset{[1,1]}{\square} \longrightarrow \bigcirc$$

The best d_0 alignment for the example in the diagram below is $\gamma = (1, 3, 4)$ with minimum cost $d_0(\sigma, \gamma) = 4$. The best d_∞ alignment for the example in the diagram below is also $\gamma = (1, 3, 4)$, but this time with minimum cost $d_\infty(\sigma, \gamma) = 3$, evidenced by the move sequence $(delay(-2, 1)delay(+1, 2))$. And lastly, the best d_1 alignment for the example in the diagram below is also $\gamma = (1, 3, 4)$, the sequence of moves being simply one stamp and one delay move at the start, $m = stamp(-1, 1)delay(-1, 1)$, and now with minimum cost $d_1(\sigma, \gamma) = 2 < \min\{d_0(\sigma, \gamma), d_\infty(\sigma, \gamma)\}$.

Theorem 2. *Given a sequential process model N of a time Petri Net N and a sequential observed trace σ, the word $\gamma \in \mathcal{L}(N)$ such that $f_\gamma(i) = \operatorname*{argmin}_{x \in [Eft(t_i), Lft(t_i)]} |x - f_\sigma(i)|$ also has the property $\gamma = \operatorname*{argmin}_{x \in \mathcal{L}(N)} d_1(x, \sigma)$.*

This of course means that we have a linear time algorithm for aligning sequential processes under d_1, by locally choosing the best flow vector as shown in Algorithm 2. This, incidentally, is exactly the aligning word obtained by the algorithm developed for delay-only distance, d_θ in [6].

5 Implementation

We have implemented the d_1 computation algorithm and the d_1 alignment algorithm for sequential process models in Python, available at https://github.com/NehaRino/TimedAlignments. Both algorithms have linear time complexity, and so, they have efficient running times, as evidenced by the table below:

Trace Length	Running Time (seconds)
10	0.00003
100	0.00024
1000	0.00259
10000	0.02811
100000	0.33131
1000000	3.44314

6 Perspectives and Conclusion

In this paper, we studied the alignment problem for timed processes using the weighted mixed moves distance d_α. We mainly focused on the evenly mixed distance d_1, and solved both distance computation and the purely timed alignment problem for the same. We also proved that for $\alpha \geq 2$, this reduces to the already solved delay-only case, which has a linear algorithm. As far as we know, this (along with [6]) is the first step in conformance checking for time-aware process mining, and much further work can be inspired from this point. Firstly, we have only solved the mixed distance alignment problem for a subset of all possible mixed distances d_α, and that too only over sequential process models, which are rather structurally restricted. As seen in this paper, even over sequential process models the problem is subtle, and while considering models that branch outwards might retain our method's efficiency, we suspect models with synchronisation between different events in particular would be tougher to tackle. It is essential to develop practical techniques for more complex classes of models, which will provide the foundation needed to consider extensions of time Petri nets with data or resources. Secondly, further investigation into the general timed alignment problem is necessary, as our proposed approach here is rather rudimentary and can certainly be improved. Lastly, there are a number of other conformance artefacts that can be set and studied in the timed setting, such as anti-alignments [5], and it would be very interesting to better develop all such conformance checking methods in a manner that accounts for timed process models.

References

1. van der Aalst, W.M.P., Santos, L.: May i take your order? - on the interplay between time and order in process mining. In: Marrella, A., Weber, B. (eds.) BPM 2021. LNBIP, vol. 436, pp. 99–110. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94343-1_8
2. Adriansyah, A.: Aligning observed and modeled behavior. Ph.D. thesis, Technische Universiteit Eindhoven (2014). https://doi.org/10.6100/IR770080
3. Asarin, E., Basset, N., Degorre, A.: Distance on timed words and applications. In: Jansen, D.N., Prabhakar, P. (eds.) FORMATS 2018. LNCS, vol. 11022, pp. 199–214. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00151-3_12
4. Boltenhagen, M., Chatain, T., Carmona, J.: A discounted cost function for fast alignments of business processes. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 252–269. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_17
5. Chatain, T., Boltenhagen, M., Carmona, J.: Anti-alignments - measuring the precision of process models and event logs. Inf. Syst. **98**, 101708 (2021). https://doi.org/10.1016/j.is.2020.101708
6. Chatain, T., Rino, N.: Timed alignments. In: 4th International Conference on Process Mining, ICPM 2022, pp. 112–119. IEEE (2022). https://doi.org/10.1109/ICPM57379.2022.9980687
7. Chatterjee, K., Ibsen-Jensen, R., Majumdar, R.: Edit distance for timed automata. In: 17th International Conference on Hybrid Systems: Computation and Control (part of CPS Week), HSCC'14, Proceedings, pp. 303–312. ACM (2014). https://doi.org/10.1145/2562059.2562141
8. Cheikhrouhou, S., Kallel, S., Guermouche, N., Jmaiel, M.: The temporal perspective in business process modeling: a survey and research challenges. SOCA **9**, 75–85 (2014)
9. Conforti, R., La Rosa, M., ter Hofstede, A.H.M., Augusto, A.: Automatic repair of same-timestamp errors in business process event logs. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 327–345. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_19
10. Eder, J., Panagos, E., Rabinovich, M.: Time constraints in workflow systems. In: Jarke, M., Oberweis, A. (eds.) CAiSE 1999. LNCS, vol. 1626, pp. 286–300. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48738-7_22
11. Gupta, V., Henzinger, T.A., Jagadeesan, R.: Robust timed automata. In: Maler, O. (ed.) HART 1997. LNCS, vol. 1201, pp. 331–345. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0014736
12. Nguyen, A., Chatterjee, S., Weinzierl, S., Schwinn, L., Matzner, M., Eskofier, B.: Time Matters: Time-Aware LSTMs for Predictive Business Process Monitoring, pp. 112–123, March 2021. https://doi.org/10.1007/978-3-030-72693-5_9
13. Rino, N., Chatain, T.: Timed alignments with mixed moves. CoRR abs/2210.15209 (2022). https://doi.org/10.48550/arXiv.2210.15209
14. Rogge-Solti, A., Mans, R.S., van der Aalst, W.M.P., Weske, M.: Repairing event logs using timed process models. In: Demey, Y.T., Panetto, H. (eds.) OTM 2013. LNCS, vol. 8186, pp. 705–708. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41033-8_89

16th International Workshop on Social and Human Aspects of Business Process Management (BPMS2 2023)

Social and Human Aspects of Business Process Management (BPMS2 2023)

The involvement of human aspects in Business Process Management occurs both on a social and individual level. Social information systems, such as social media, Enterprise 2.0, and social platforms, are spreading quickly in society, organizations, and economics. Integrating business process management and social information systems is becoming more widespread. New approaches for using social information systems and business process management appear frequently.

Social information systems are utilized in both internal and external business operations. Companies can collaborate with customers in product development to gather ideas and enhance services. This leads to two-way communication between the company and its customers. Social information systems distinguish themselves from traditional information systems by allowing for emergent interactions. These interactions occur during runtime and involve multiple stakeholders without requiring approval from a supervisor or management. Emergent interactions facilitate the transformation of personal knowledge into collective knowledge, making them an effective tool for harnessing collective intelligence in the digital era.

By combining business process management with social information systems, innovative business models can be developed using social platforms. TripAdvisor, Uber, and AirBnB serve as excellent examples of this. The value-creating mechanisms of social information systems have made it possible to achieve business models that were previously unattainable. For instance, AirBnB has implemented a unique quality control model by utilizing user reviews of apartments. This approach has made it feasible to assess the quality of products and services in a cost-effective manner.

Business process management involves human and social aspects, with individuals becoming an essential part of many processes. Digital assistants like Alexa can integrate people into processes that traditional computers cannot handle. The human aspect includes creating process models, communication, collaboration, coordination, and cooperation. It also encompasses specific human-related aspects and their representation in models.

Given this background, the goal of the BPMS2 workshop is to explore how social information systems integrate with business process management, and how business process management may profit from this integration. Furthermore, the workshop investigates the human aspects of Business Process Management by involving human actors. Examples are using crowdsourced knowledge and tasks and the need for new user interfaces, e.g., augmented reality and voice bots.

Three papers were accepted for presentation at the BPMS2 2023 workshop. Joklan Imelda, Camelia Goni, and Amy van Looy in their paper with the title ‘Towards a Measurement Instrument for Assessing Capabilities when Innovating Less-Structured Business Processes’ develop a measurement instrument to assess capabilities for innovating less-structured business processes (LSBPs). Through a Delphi study and two illustrative case studies, the researchers build and demonstrate an LSBP innovation capability (PIC-LSBP) measurement tool.

The instrument covers 6 main capability areas, 18 sub-areas, and 55 measurement items identified as important for LSBP innovation. Areas include employee skills, process specifics, information systems, entrepreneurial strategy, knowledge management, and external partnerships.

Case studies in a manufacturer and airport service company highlight differences across sectors in current capabilities and gaps. Findings emphasize the value of a tailored tool to evaluate an organization’s innovation capabilities for a specific process type like LSBP.

Mehran Majidian Eidgahi, Anne-Marie Barthe-Delanoë, Dominik Bork, Sina Namaki Araghi, Guillaume Mace-Ramete, and Frédérick Bénaben integrate social media and business process management (BPM) to improve agility in their paper “Integrating Social Media and Business Process Management: Exploring the Role of AI Agents and the Benefits for Agility”

A new social media-based BPM platform has been suggested which allows stakeholders to be dynamically involved in the design and run-time aspects of BPM lifecycles. The platform has demonstrated implementing an agility framework within short and mid-term timeframes. However, automation is needed for long-term agility exceeding human capabilities. Introducing AI agents is proposed to address this gap. AI agents are identified to contribute through real-time monitoring, adaptive decision-making, intelligent task execution, and adaptive resource allocation. This would enhance reliability, speed, and adaptation of the platform.

In their paper “Design Principles for Using Business Process Management Systems” Sebastian Dunzer, Willi Tang, Nico Höchstädter, Sandra Zilker and Martin Matzner explore how BPMS features can aid operations in organizations. Through an action design research project with a medium-sized fashion company, the researchers helped adapt the company’s issue-tracking system into a tailored BPMS to improve their software testing and promotion-code handling processes.

The key findings indicate that organizations can repurpose existing IT systems, like issue trackers, to function as bespoke BPMSs without major IT changes. The adapted BPMS improved the company’s processes by enabling user management, customizable workflows, automation, monitoring, and case handling.

Based on their experiences, the researchers propose seven socio-technical design principles for BPMS features that provide guidance on mechanisms and aims of each feature. A key principle is integration, highlighting that BPMSs should connect across systems to support end-to-end processes.

We wish to thank all the people who submitted papers to BPMS2 2023 for sharing their work with us, the many participants creating fruitful discussions, and the BPMS2 2023 Program Committee members, who made a remarkable effort in reviewing the submissions. We also thank the organizers of BPM 2023 for their help with organizing the event.

Selmin Nurcan
Rainer Schmidt

Organization

Program Committee

Jan Bosch	Chalmers University of Technology, Sweden
Norbert Gronau	University of Potsdam, Germany
Holger Günzel	Munich University of Applied Sciences, Germany
Kathrin Kirchner	Technical University of Denmark, Denmark
Michael Möhring	Hochschule Reutlingen, Germany
Selmin Nurcan	Université Paris 1 Panthéon-Sorbonne, France
Mohammad Rangiha	Moody’s Corporation, UK
Flavia Santoro	UERJ, Brazil
Rainer Schmidt	Munich University of Applied Sciences, Germany
Miguel-Angel Sicilia	University of Alcalá, Spain
Pnina Soffer	University of Haifa, Israel
Johannes Tenschert	FAU Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Irene Vanderfeesten	KU Leuven, Belgium
Moe Thandar Wynn	Queensland University of Technology, Australia
Alfred Zimmermann	Reutlingen University, Germany

Integrating Social Media and Business Process Management: Exploring the Role of AI Agents and the Benefits for Agility

Mehran Majidian Eidgahi[1](✉), Anne-Marie Barthe-Delanoë[1], Dominik Bork[2], Sina Namaki Araghi[4], Guillaume Macé-Ramète[3], and Frederick Benaben[1]

[1] Centre de Génie Industriel, Université Toulouse - Mines Albi, Campus Jarlard, Route de Teillet, 81000 Albi, France
{Mehran.Majidian,Anne-marie.Barthe,Frederick.BENABEN}@mines-albi.fr
[2] TU Wien, Vienna, Austria
Dominik.Bork@tuwien.ac.at
[3] ITEROP, 1 bis rue Antoine Lavoisier, 31770 Colomiers, France
Guillaume.MACE-RAMETE@3ds.com
[4] Laboratoire Génie de Production de l'École Nationale d'Ingénieurs de Tarbes (LGP-INP-ENIT), Université de Toulouse, 65016 Tarbes, France
Sina.Namakiaraghi@enit.fr

Abstract. Business Process Management (BPM) faces increasing societal and business challenges. Societal issues include effectively managing unexpected changes and ensuring employee engagement during process modifications. The former category involves the limitations of technology when introducing process changes while Agile BPM and Social BPM concepts have been explored in previous studies to tackle the latter. These investigations have resulted in the development of a Social-Media (SM) based BPM platform and an agility framework. The SM platform draws inspiration from popular social media platforms such as Twitter and Instagram. Its primary objective is to merge the design-time and run-time phases of the business process (BP) lifecycle as well as to actively engage stakeholders in the dynamic design and implementation of BPs. By simplifying the response to potential changes and maintaining stakeholder involvement throughout the process, the platform addresses the aforementioned challenges. This article establishes a correlation between SM platform concepts and BPM concepts, emphasizing how they facilitate the concurrent design and execution of BPs. Furthermore, the article demonstrates the application of the SM platform in implementing the agility framework through a use case analysis involving a scientific paper submission process. To overcome the current emphasis on stakeholder involvement and human capabilities in applying the platform, we finally explore the possibilities of employing AI agents as automated assistants.

Keywords: Business Process Management · Agile BPM · Social BPM · Agility · AI agent

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 205–216, 2024.
https://doi.org/10.1007/978-3-031-50974-2_16

1 Introduction

Business Process Management (BPM) focuses on managing the lifecycle of business processes (BPs) and the associated technologies within organizations that aim to drive their business by executing and optimizing the underlying processes. These processes are crucial for delivering valuable services or products to end-users [1,2,17]. While BPM can provide significant benefits, such as increased efficiency and productivity, it also faces a number of social and business challenges that need to be addressed.

Regarding societal challenges, modern businesses heavily depend on their capacity to effectively address both anticipated and unforeseen changes. For example, as organizations introduce new technologies, systems, and tools to their processes, they must ensure that those changes are integrated smoothly, reactively, and effectively. Failure to manage changes reactively and effectively can result in disruptions to processes, increased costs, and delays in achieving business objectives [5–7]. Another concern revolves around the employees, who are the primary catalysts for initiating changes. Organizations need to ensure that employees are informed and engaged throughout the change process and that they have the support and resources they need to adapt to the changes [3]. As a result, BPM has long focused on agility in order to be ready for either foreseen or unpredicted uncertainties and changes.

In terms of BPM business challenges, one of the earliest challenges is related to the limitations of technology. In the early days of BPM, tools and systems were often rigid and inflexible, making it difficult to introduce changes to processes. Organizations had to rely on IT professionals to implement changes, which was often time-consuming and expensive [12]. BPM orchestrators were not exempt from this problem. Despite the advent of agile orchestrators, challenges persist, including the complexity of systems, the importance of robust governance, and the integration with external systems and applications. These challenges add complexity to the redesign phase of the BP lifecycle.

To address the aforementioned BPM challenges, this research attempts to answer the following research question:

- **In what ways can tools, technology, and AI bring agility to BPs and BPM orchestrators?**

This research is part of a series of studies that address the aforementioned challenges [13,14]. In the remainder of this paper, the basic concepts related to BPM, agile BPM, and Social BPM are presented in Sect. 2. Furthermore, Sect. 3, examines how the developed platform integrates the design-time and run-time phases of the BP lifecycle and subsequently showcases the utilization of the SM platform in implementing the agility framework. Section 4 explores the potential contributions of AI agents within the platform. Eventually, the conclusion of this paper is available in Sect. 5.

2 Background

In the following, we provide background information on Agile BPM and Social BPM, two concepts that have emerged as partial solutions to address the challenges mentioned at the outset.

2.1 Agile BPM

Agility is a reactive and efficient process of discovering changes and deciding on and implementing a response strategy [5–7]. Agility has been explored and developed in a variety of industries, including manufacturing, software development, and corporate management. To the best of our knowledge, there are three major agility abstraction levels: **discovering** a need for change [7], **deciding** a response strategy [9], and **implementing** the chosen strategy [7].

The previous abstraction levels remained the basics for agile BPM as well. Agile BPM is an approach to BPM that emphasizes flexibility, collaboration, and continuous improvement. One of the key characteristics of agile BPM is the focus on collaboration and teamwork. Agile BPM emphasizes the involvement of all stakeholders in the BPM process, including business users, IT staff, and other relevant parties. This collaborative approach is designed to ensure that BPM processes are aligned with business needs and that they can be adapted quickly and efficiently in response to changing requirements [4]. Despite the fact that agile BPM emphasizes incorporating stakeholders in orchestrating BPs, or, to put it another way, socializing BPs, this incorporation remains a difficulty [8].

2.2 Social BPM

Social BPM is an approach to BPM that incorporates social media and collaboration technologies into the design, execution, and monitoring of BPs to facilitate stakeholder engagement. This approach enables organizations to improve communication, collaboration, and knowledge sharing among employees, customers, and partners [3,11]. According to our understanding, Social BPM may be viewed as a solution to the agile BPM challenge of socializing BPs.

Several studies have explored the benefits and challenges of Social BPM. For example, Social BPM can lead to increased collaboration and process visibility, as well as improved process efficiency and flexibility [16]. However, there are challenges such as privacy and security concerns, cultural resistance, and difficulty in measuring the impact of social media on BPs [15].

Some studies examined the role of social media in the design and execution of BPs. They found that social media can be used to support collaboration, knowledge sharing, and feedback gathering throughout the process lifecycle. However, the studies also identified challenges related to the integration of social media into existing process management systems [3].

3 SM-Based Process Management

An innovative Social Agile BPM tool has been developed to tackle the aforementioned challenges of Agile BPM while leveraging the benefits of Social BPM. It is a text-based platform inspired by Twitter and Instagram. The main objective of this platform is to dynamically involve the relevant stakeholders in the design and implementation of each BP. This platform merges the design and execution phases of BPM (Design-time and Run-time), thereby aiming to simplify the response to potential changes. By engaging the stakeholders throughout the BP lifecycle, they are constantly informed about the current state of the process and are ready to take necessary actions in case of any required modifications. Our earlier paper [14], contains further information on the application.

3.1 Mapping Between SM Concepts and BPM Concepts

The platform incorporates four key concepts to enable stakeholders to design and execute BPs simultaneously: *Post*, *Comment*, *Reaction*, and *Notification.*

Figure 1 presents the metamodel depicting the links between the notions of BPM design-time, run-time, and the platform concepts of the SM platform. In other words, it explains how the platform concepts enable users to design and orchestrate BPs simultaneously. Here, the main idea is to fulfill a BP by determining its primary goal, the requirements to achieve it, and the way of achieving it. Within the SM platform's package, the Post class represents the main goal, encapsulating process information such as deadlines and involved people. The Comment class defines the requirements associated with the goal specified in the post. The Reaction and Notification classes facilitate the process of achieving the goal, working in conjunction with the Comment class. In traditional BPM, these three aspects are addressed in two complementing, distinct phases: Design-time and Run-time. In the Design-time package, the Process class represents the goal. To address the question of what is required to achieve the goal, the Task class from Design-time and Participants from Run-time define tasks and their associated resources. Additionally, the Gateways and Event classes from Design-time, along with the Execution Context and Token classes from Run-time, define and track the sequence or timing of tasks. Figure 1 demonstrates how the platform merges these two phases and fulfills their respective duties through its own classes, answering the questions necessary for fulfilling a BP.

Use Case Analysis. The offered use case highlights the steps of a high-granularity scientific paper submission procedure. During this procedure, a Ph.D. student begins by supplying the necessary information such as *controllers* who have the authority to react to and validate the post as well as assign tasks to resources, *contributors* who can only track and comment on posts, and conference information. The student will then get validation of the process from controllers, as well as duties to finish the procedure on a regular basis. This procedure was executed using the developed tool. All of the process's participants communicate within the process post, such as assigning tasks, making recommendations, amending documents, or responding to the process's ongoing events.

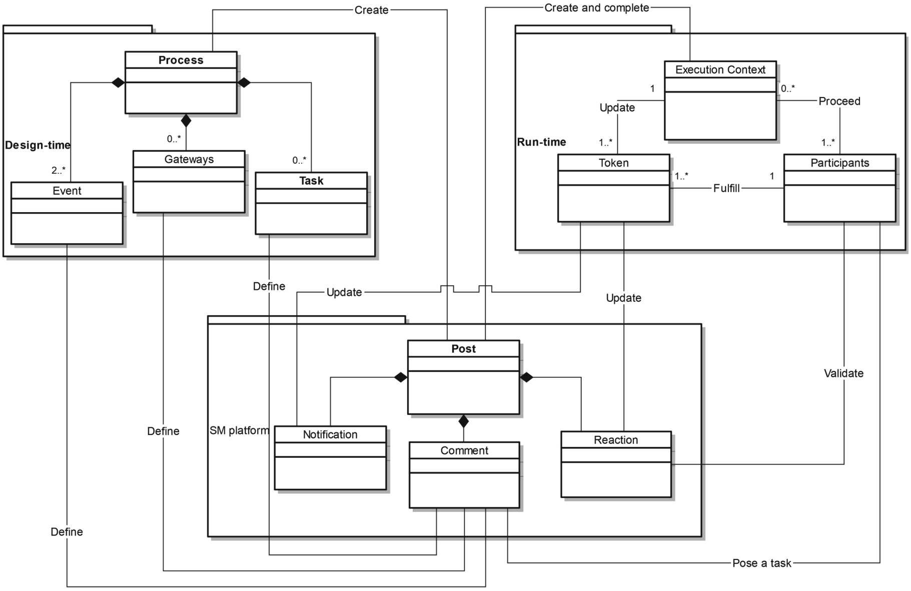


Fig. 1. Relationships between design-time, run-time and platform's concepts

The first column in Table 1 contains all procedure steps. This table also shows in the second column which platform concept(s) were utilized to complete each step. The third and fourth columns define the design-time and run-time concepts required in conventional BPM for each step that the platform covers simultaneously or independently as part of the merging design-time and run-time approach. In step 5, for example, one of the controllers comments under the post to "check the format of the conference and send the first draft in two weeks." It made advantage of the Comment platform concept. This comment assigns two tasks to the student and creates a parallel gateway. So, among design-time concepts, tasks, and gateways are required. It uses the participant concept to specify the resource to accomplish the task, which is one of the run-time concepts.

3.2 Application of SM in Implementing the Agility Framework

The agility framework depicted in Fig. 2 is developed to be a guide for implementing or improving agility in BPM that covers agility abstraction levels, the requirements for each abstraction level, and their correlation with BPM lifecycles. The abstraction levels and their requirements are influenced by a review of the definitions and contributions of agility in the literature. The association between abstraction levels and BP lifecycle was then inferred by creating a clear description for each abstraction level and BP lifecycle and assessing the literature on agile BPM. Time scale is a new dimension that adds agility diagnostic ability to the framework. At first glance, Fig. 2 is divided into two parts of Run-time

Table 1. Use-case steps and their relationship to concepts

Step	SM platform	Design-time	Run-time
1- Create a post "Submit paper"	Post	Process	Execution context
2- Notify Controllers	Notification	Event	Participants
3- Validate the post by controllers	Reaction	Event	Participants, Token
4-Notify contributors	Notification	Event, Task	Participants
5- Comment "check the format of conference and send the first draft in two weeks" by controllers	Comment	Task, Gateways	Participants
6- Finish the "format checking" task	Reaction	Event	Participants, Token
7- Comment "Write the paper on Overleaf" by contributors	Comment	Event	–
8- Notify Controllers about "Overleaf" suggestion	Notification	Event	Participants
9- Validate the suggestion by controllers	Reaction	Task, Gateways	Participants, Token
10- Finish the "Overleaf" and "First draft" tasks	Reaction, Comment	Event	Participants, Token
11- Send the reviewed paper and ask creator to revise it by controllers	Comment	Task	Participants
12- Finish the "Revision" task and send the revised paper	Comments, Reaction	Event	Participants, Token
13- Accept the paper and ask to submit by controllers	Comments	Event	Participant
14- Finish the "Submit" task	Reaction	Participants	Token

(yellow part on the top) and Design-time (green part on the bottom), which are the stages of the BPM lifecycle. Run-time contains three abstraction levels of agility represented as three gray vertical columns, which implies that agility's abstraction levels execution is devoted to the run-time while agility's abstraction levels preparation is devoted to the design-time. The definition of these abstraction levels is as follows:

- **Discovering:** finding need for changes by gathering information about the system's situation.
- **Deciding:** making a decision on how to respond to the discovered need for change according to the capabilities of the system.
- **Implementing:** an action to trigger the compositions of the system to embody the decision and respond to the discovered change.

These abstraction levels overlap with the time scale representing three distinct timespans used to assess the duration of abstraction levels performed: the *agility objective time scale* that defines what perspective is required to respond to the occurred change regarding the context. The agility objective time scale is equivalent to the highest available time scale and corresponding functions for abstraction levels. Our earlier study [13] explores the framework in further detail.

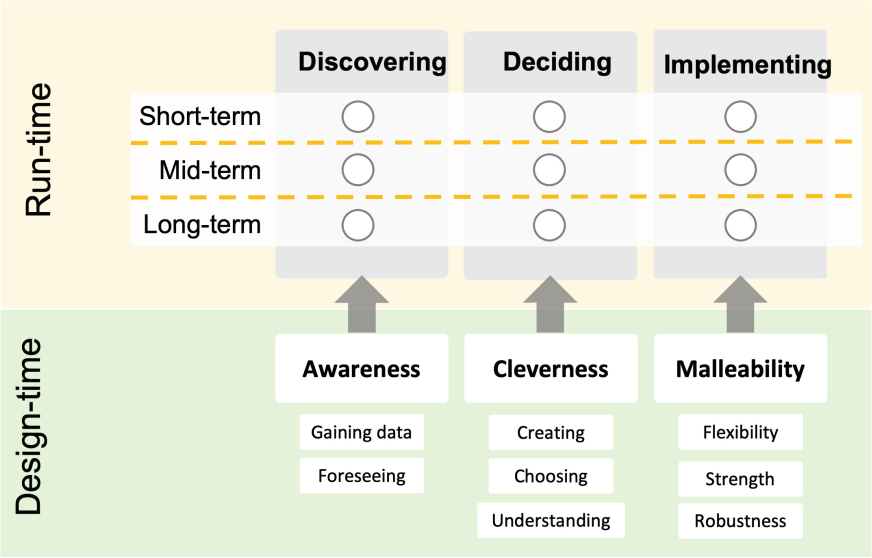


Fig. 2. The agility framework

Use-Case Analysis. Different change scenarios that can occur throughout the submission of a paper use case are defined here to analyze how the use of the SM platform implements specific components of this agility framework. In the **first scenario there is timing issue**. The controllers have not observed any advancement in the process for a considerable period subsequent to submitting their review. Therefore, the controllers opt for the short-term agility objective to reactively address this change. The controllers checked the student's delay in revising the paper. They sensed a time problem in the *discovering* abstraction level because of the significant delay and the impending conference date. This sensing was made possible by the platform's given *awareness* and *gaining data* option. After a discussion under the post as a reflex to the problem, it was discovered that the writer is new to Overleaf, and they determined in the *deciding* abstraction level that one of the contributors assists the student with Overleaf. The chosen contributor was added to the resources of the revising task in the *implementing* abstraction level since the platform kept updating the involved people and making them available for the process. In the **second scenario there is a process deviation**. The controllers feel that the paper did not match the conference well after the first draft review. So the controllers choose the mid-term agility objective to respond to such change effectively and reactively. Based on the *awareness* they gained from prior similar processes and the real-time *gaining data* of the present process they detected unmatch parts of the paper in the *discovering* abstraction level. Then, depending on the supplied *understanding* of the process, involved people offer new conferences and provide the *choosing* option for the *deciding* abstraction level. The new conference was decided based on the availability of *understanding* and *choosing* options in the *deciding* abstraction level. Taking into account the platform's *flexibility* and the *malleability* of the post information, adjustments were implemented and post information was updated in the *implementing* abstraction level.

Taking into account recent use cases, the platform demonstrates its suitability for implementing the agility framework in response to changes for both *short-term* and *mid-term* agility objectives. The platform not only facilitates the implementation of each abstraction level but also fulfills the requirements associated with each level. This requirement fulfillment is not a one-time task but an ongoing process that strengthens and expands the requirements in alignment with the progress of the process. The platform's capability in this regard can be attributed to two primary factors. Firstly, it involves the relevant individuals in the process, ensuring their active participation. Secondly, it keeps these individuals informed and updated about the process using simple and understandable human language, thus facilitating effective communication. Regarding the *long-term* agility objective, while the platform offers data and information storage capabilities, it is not a task that can be solely accomplished by individuals alone. To illustrate, the *discovering* abstraction level of the *long-term* agility objective may necessitate the utilization of data mining techniques to delve into historical data and identify patterns. The capabilities and challenges highlighted earlier in relation to the platform present an opportunity to consider integrating AI into the platform to address the identified gaps.

4 Robust Agility Framework Through AI Agents in SM

In many aspects, an AI agent may be viewed as assisting involved people of a BP to facilitate the BP's execution and response to changes. In this section, we elucidate the potential contributions of an AI agent integrated into the SM platform. Firstly, we explore the benefits that such an AI agent can bring to the platform. Secondly, we investigate the specific areas of the agility framework that the AI agent could augment. Additionally, we analyze how the AI agent can enhance the existing components of the agility framework that have not already been implemented.

4.1 Benefits of an AI Agent in the SM Platform

AI agents can play a vital role in proposing suggestions during the creation and execution of BPs. These suggestions could come from analyzing historical process data and identifying patterns to improve the platform in the following areas:

- **Process Optimization**: By leveraging machine learning algorithms, agents can provide recommendations on tasks, resource allocation, task prioritization, and process sequencing, enabling the platform to dynamically adapt and optimize the BPs [10].

- **Intelligent Process Automation**: AI agents automate repetitive or rule-based tasks within the BPM platform. By learning from historical data, agents can autonomously perform tasks, reducing manual effort and improving process efficiency. This automation frees human participants to focus on more complex and strategic aspects of the processes, thereby enhancing overall productivity.
- **Collaborative Process Guidance**: AI agents act as virtual assistants within the BPM platform, providing guidance and support to human participants. They answer questions, provide context-specific information, and suggest the next steps or alternative actions. This collaborative guidance facilitates effective communication and collaboration among participants, leading to improved user experience and adherence to process guidelines.

Thus far, the potential of an AI agent to facilitate the design and execution of BPs has been recognized. Subsequently, the focus shifts towards exploring the AI agent's potential in assisting with responding to changes, or in other words, enhancing agility. Based on our current understanding, the assistance provided by the AI agent should aim to execute agility abstraction levels in a more reactive and efficient manner.

4.2 Augmentation of the Agility Framework by the AI Agent

The first way an AI agent might aid in the improved deployment of the agility framework on the platform is through real-time process monitoring. AI agents continuously monitor the execution of BPs in real-time. They analyze data from various sources, such as IoT devices and user interactions, to track process performance, detect anomalies, and provide timely alerts or notifications. Real-time monitoring enables proactive intervention and corrective actions, ensuring smooth process execution, and could be a great contribution to the *discovering* abstraction level. For example, in the first scenario of the use case, if there was a timing issue, the AI agent might detect this change based on a specified delay policy at a suitable time without the danger of human error. Another aspect that might be seen as a contribution to the *deciding* abstraction level is adaptive decision-making. AI agents analyze data, contextual information, and predefined rules to make informed decisions during process execution. By considering factors such as user preferences, workload, and resource availability, agents dynamically adjust process flows, allocate tasks, or escalate issues, ensuring efficient and agile process execution. AI agents might add to the *implementation* abstraction level in two ways: Intelligent Task Execution and Adaptive Resource Allocation. For the first one, AI agents can autonomously execute tasks within the BPs. They can leverage their learning capabilities and knowledge of process requirements to perform tasks accurately and efficiently. AI agents can also handle exceptions or deviations from the predefined process flow, ensuring smooth process execution. For the second one, AI agents can analyze resource availability, workload, and skill requirements to dynamically allocate resources during process execution. They can make intelligent decisions on resource assignments based on real-time data and optimize resource utilization. This helps in ensuring that the right resources are allocated at the right time, enhancing efficiency and productivity. Overall, the contributions of the AI agent have the potential to enhance the agility of the platform in multiple ways. Firstly, it helps eliminate the risk of human error, thereby improving the overall reliability and accuracy of the system. Secondly, it enables the implementation of abstraction levels within shorter timeframes, allowing for more efficient and timely responses to changes. This increased speed and responsiveness enhance the platform's adaptation to evolving circumstances.

Regarding the long-term timeframe that was missing in the platform from the agility framework, AI agents contribute to discovering, deciding, and implementing abstraction levels by leveraging advanced data mining, analysis, decision support, and adaptive automation capabilities. AI agents can realize these contributions in each abstraction level in the following:

1. **Discovering.** *Mining and Analysis:* AI agents can leverage machine learning algorithms to mine and analyze large volumes of process-related data. They can identify patterns, correlations, and insights from historical and ongoing process execution data, user interactions, and other relevant sources. This analysis helps in discovering hidden process knowledge, identifying bottlenecks, trends of changes, and detecting unexpected or unforeseen situations.
2. **Deciding.** *Complex Analysis and Decision Support:* AI agents can employ predictive models to forecast process outcomes and potential bottlenecks. By considering historical data, external factors, and process dynamics, AI agents can anticipate future process states and provide insights to support decision-making. This helps in proactive planning, and risk mitigation.
3. **Implementing.** *Evolutionary Process Improvement:* AI agents can analyze process performance, collect user feedback, and learn from past executions to suggest incremental improvements. AI agents can identify process inefficiencies, propose changes, and guide the implementation of process updates ensuring that processes remain agile and aligned with changing business needs.
   Adaptive Automation: AI agents can dynamically adapt process automation based on changing requirements and conditions. They can learn from historical data and user interactions to identify opportunities for automation or augmentation. AI agents can recommend when and how to automate tasks or decisions, ensuring that the implemented automation aligns with the evolving needs of the organization.

5 Conclusion

This study continues an ongoing stream of research that aims to assess the integration of design-time and run-time concepts within a Social BPM platform. The findings of the study revealed that the platform effectively addresses both the design-time and run-time concepts. The platform effectively tackles the societal and business challenges associated with BPM and seamlessly integrates design-time and run-time aspects, while actively engaging stakeholders throughout the entire BP lifecycle. Moreover, it was observed that the platform demonstrates strong applicability for the agility framework, particularly in the short-term and mid-term time spans, regarding the implementation of abstraction levels and meeting their requirements. However, the long-term time span remained a challenge at the current stage, as it often exceeds human capabilities and requires automation. The introduction of an AI agent was identified as a potential solution to this challenge, with the added benefit of enhancing agility through intervention in abstraction levels. The AI agent can mitigate or reduce the risk of human error while also accelerating the performance on all abstraction levels.

In summary, AI agents can serve as a valuable component within the platform, aiding individuals in the seamless achievement of BPM's primary objectives, namely the design and execution of BPs. The AI agent can fulfill this role by providing recommendations for upcoming tasks, task assignments, process monitoring, and issuing timely alerts. As a potential area for future research, the integration of an AI agent into the platform holds promise. We anticipate that the AI agent could contribute to various domains, such as process optimization and process monitoring within the platform. However, implementing the AI agent across all aspects of the platform would necessitate careful scheduling and prioritization to ensure regular and comprehensive integration.

References

1. van der Aalst, W.M.P.: Business process management: a comprehensive survey. Int. Sch. Res. Notices **2013** (2013)
2. van der Aalst, W.M.P., van Hee, K.M.: Workflow Management: Models, Methods, and Systems. Cooperative Information Systems, MIT Press, Cambridge (2002). ISBN 0-262-01189-1
3. Ariouat, H., Hanachi, C., Andonoff, E., Bénaben, F.: A conceptual framework for social business process management. In: Zanni-Merk, C., Frydman, C.S., Toro, C., Hicks, Y., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference KES-2017, Marseille, France, 6–8 September 2017, Procedia Computer Science, vol. 112, pp. 703–712, Elsevier (2017). https://doi.org/10.1016/j.procs.2017.08.151
4. Badakhshan, P., Conboy, K., Grisold, T., vom Brocke, J.: Agile business process management: a systematic literature review and an integrated framework. Bus. Process. Manag. J. **26**(6), 1505–1523 (2020). https://doi.org/10.1108/BPMJ-12-2018-0347
5. Barthe-Delanoë, A., Truptil, S., Bénaben, F., Pingaud, H.: Event-driven agility of interoperability during the run-time of collaborative processes. Decis. Support Syst. **59**, 171–179 (2014). https://doi.org/10.1016/j.dss.2013.11.005
6. Barthe-Delanoë, A.M., Montarnal, A., Truptil, S., Benaben, F., Pingaud, H.: Towards the agility of collaborative workflows through an event driven approach. Int. J. Disaster Risk Reduct. **28**, 214–224 (2018). https://doi.org/10.1016/j.ijdrr.2018.02.029
7. Bénaben, F., Mu, W., Boissel-Dallier, N., Barthe-Delanoë, A., Zribi, S., Pingaud, H.: Supporting interoperability of collaborative networks through engineering of a service-based mediation information system (MISE 2.0). Enterp. Inf. Syst. **9**(5–6), 556–582 (2015). https://doi.org/10.1080/17517575.2014.928949

8. Bruno, G., et al.: Key challenges for enabling agile BPM with social software. J. Softw. Maint. Res. Pract. **23**(4), 297–326 (2011). https://doi.org/10.1002/smr.523
9. Gong, Y., Janssen, M.: Agent-based simulation for evaluating flexible and agile business processes: separating knowledge rules, process rules and information resources. In: Barjis, J. (ed.) EOMAS 2010. LNBIP, vol. 63, pp. 41–58. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15723-3_4
10. Ilieva, R., Nikolov, Y.: AI integration in business processes management (2019). https://doi.org/10.1109/CREBUS.2019.8840086, cited by: 4
11. Kocbek, M., Jošt, G., Polančič, G.: Introduction to social business process management. In: Uden, L., Heričko, M., Ting, I.-H. (eds.) KMO 2015. LNBIP, vol. 224, pp. 425–437. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21009-4_33
12. Lacity, M.C., Willcocks, L.P.: An empirical investigation of information technology sourcing practices: Lessons from experience. MIS Q. **22**(3), 363–408 (1998). http://misq.org/an-empirical-investigation-of-information-technology-sourcing-practices-lessons-from-experience.html
13. Majidian-Eidgahi, M., Bork, D., Barthe-Delanoë, A., Namaki-Araghi, S., Mace-Ramete, G., Benaben, F.: An agility framework for business process management. Bus. Process Manag. J. under review (2023)
14. Majidian-Eidgahi, M., Namaki-Araghi, S., Bork, D., Barthe-Delanoë, A., Mace-Ramete, G., Benaben, F.: A social bpm approach to deal with agility. In: Proceedings of the 15th International Conference on Management of Digital EcoSystems, MEDES 2023 (2023)
15. Prodanova, J., Looy, A.V.: How beneficial is social media for business process management? A systematic literature review. IEEE Access **7**, 39583–39599 (2019). https://doi.org/10.1109/ACCESS.2019.2903983
16. Suša Vugec, D., Tomičić-Pupek, K., Bosilj Vukšić, V.: Social business process management in practice: overcoming the limitations of the traditional business process management. Int. J. Eng. Bus. Manag. **10** (2018). https://doi.org/10.1177/1847979017750927
17. Weske, M.: Introduction, pp. 3–23. Springer, Berlin, Heidelberg (2012). ISBN 978-3-642-28616-2, https://doi.org/10.1007/978-3-642-28616-2_1

Design Principles for Using Business Process Management Systems

Sebastian Dunzer, Willi Tang(✉), Nico Höchstädter, Sandra Zilker, and Martin Matzner

Friedrich-Alexander-Universität Erlangen-Nürnberg, Fürther Str. 248,
90429 Nürnberg, Germany
{sebastian.dunzer,willi.tang,nico.hochstadter,sandra.zilker,
martin.matzner}@fau.de

Abstract. Organizations aim to achieve operational excellence to reduce costs and improve the quality of their business processes. Business process management (BPM) enables continuous improvement of business processes. Business process management systems (BPMS) serve as an entry point to BPM activities and afford firms to manage, execute, and automate business processes. This study follows an action design research approach to design a BPMS in use together with a medium-sized German fashion company. We concurrently evaluated the artifact-in-use by tracking performance indicators that are aligned with the company's objective. As a result of our formalization of learning, we propose seven design principles for using BPMSs to achieve continuous improvement of business processes. These design principles comprise user management, process modeling, automation, logging, monitoring, integration, and case handling.

Keywords: Business process management systems · action design research · design principles

1 Introduction

An organization's operational excellence (i.e., its emphasis on efficiency and reliability in a firm's daily business activities) and the resulting firm performance depend on its ability to align its business strategy, business processes, and information technology (IT) [35]. Businesses have traditionally turned to business process management (BPM) for guidance in handling their business processes with IT, which highlights – among others – the need for business process orientation, continuous process improvement, and appropriation of technology to achieve high process performance and operational excellence [2,39]. For instance, reducing the throughput time of business processes is one aim of successfully managing structured business processes [36]. As a result, an organization may increase customer satisfaction or reduce the costs of producing an outcome. Accordingly, designing, enacting, controlling, and analyzing operational processes should be at the center of attention for businesses that want to use IT to support business processes [1].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 217–228, 2024.
https://doi.org/10.1007/978-3-031-50974-2_17

Organizations can use a business process management system (BPMS) to facilitate BPM [37]. A BPMS is defined as “a generic software system that is driven by explicit process designs to enact and manage operational business processes” [2]. Despite the variety of existing BPMSs (e.g., ARIS, Camunda, Bizagi), companies, especially small and medium enterprises (SME), lack guidance in how to effectively utilize their BPMS to improve operations [37]. Research remains inconclusive on how companies can identify appropriate use cases of BPM technology. Prior work suggests that a BPMS is most suited for organizational and administrative processes [8] and requires good integration with other systems, for example, enterprise resource planning systems or case management systems [34]. However, most research is written at a technology-oriented level for a technical audience with a STEM background to inform engineering decisions [26]. They specify individual technical objects or components that a BPMS comprises (e.g., [6,27]).

We believe that present expositions on BPMS could benefit from socio-technical descriptions to inform practice-oriented readers and scholars that focus on how organizations can put BPM principles into practice using IT. BPMSs are socio-technical systems [32], and their design should reflect the notion that the technical components do not lead to a fixed outcome [28]. Against this background, we investigate the following research question:

RQ: How do BPMS features aid operations in organizations?

Following an action design research (ADR) approach, our research article reports on the case of a medium-sized fashion company that initially used spreadsheet software to support their software testing processes and later appropriated their existing issue tracking systems (ITS) *Jira* as their “bespoke” BPMS to continuously improve their business processes. We followed an ADR approach, a hybrid of action research and design science research, to guide the design of the company’s bespoke BPMS. Based on the artifact design, we derived seven design principles following the schema proposed by Gregor et al. [11].

Our approach contributes to theory and practice as follows. First, we take on a socio-technical stance on IT. In formulating the design principles we explicitly consider the mutability of IT and the affordances they provide [11,21]. We specify the targeted user groups, the aims of a certain design principle, and the mechanisms and rationales involved to account for the general intent underlying the structural, “technical” set of features and the nondeterministic nature of the artifact when human actors use them [11]. Second, we address business process execution and suggest that organizations can use different systems, for example, their ITS, as bespoke BPMS without changes to their IT landscape.

The remainder of this paper is structured as follows: Sect. 2 provides an overview of BPMS and design principles in BPM research. After outlining the ADR approach described in Sect. 3, we propose our design principles in Sect. 4. We discuss the results and implications in Sect. 5.

2 BPMS and Design Principles in BPM

Workflow management systems (WfMS) build the foundation for BPMS [10], which are systems that define, create, and manage the execution of workflows through software [19]. They achieve those capabilities by interpreting process models and interacting with workflow participants. However, workflow participants must achieve any additional functionality by invoking other software systems. And this soon proved to restrict BPM [41]. As a result, companies started to offer software solutions that aim to cover all parts of the BPM lifecycle. BPMS today may refer to individual systems or a collection of (process-aware) systems used in combination. However, an extensive system is not always necessary, as perceived complexity can hinder technology acceptance among employees [13].

Design principles aim to pass on knowledge in a prescriptive manner. They are abstractions and do not refer to concrete instances and should be formulated in a way that they can be understood and can be applied to achieve the desired outcome [3,11]. One example is: "For *Implementer I* to achieve or allow for *Aim A* for *User U* in *Context C*, employ *Mechanisms M1, M2, ... Mn* involving *Enactors E1, E2, ... En* because of *Rationale R*" [11, p. 1633]. In research, design principles have been developed for multiple aims. Some examples are design principles for enterprise architecture management [18], gamification [20], or the assessment of human competencies [7].

In BPM research, design principles have been developed for different purposes and by applying different research methods. Design principles have been proposed in BPM for comprehensible process discovery [33], for reflecting sustainability in business models [30], or to derive theoretically grounded design principles for process guidance systems to support employees' process execution [24].

3 Method

Our study aims to design principles that increase the utility of a BPMS. We chose an ADR approach where we iteratively designed and supervised BPMS use in an organization [31]. ADR combines action research [5] and design science research [14]. It recognizes the influence of the organizational context on the design of artifacts [31]. ADR projects draw from knowledge emerging through intervention together with and in an organization. Our participating organization is a medium-sized German fashion company selling women's outerwear "WomOut".[1]

The ADR team comprised the researchers, a testing coordinator, and a person responsible for providing software internally. The end users of the BPMS were, at first software testers, extended to online marketing, and later involved other functions.

---

[1] Due to reasons of confidentiality, we can neither disclose the company's name, specific *Jira* configurations nor additional source code.

3.1 Stage I: Problem Formulation

"WomOut" operates different online shops catering to international distribution channels of their products. The decision to self-host online shops saves platform costs but introduces other expenses like developing and testing each online shop release. Therefore, WomOut decided to marginally employ testers (approx. 10) to reduce software testing costs. Every software tester received a spreadsheet containing cases for testing and result documentation. A test coordinator collected the spreadsheets, assembled them, and reported the test results to the software engineering teams. WomOut observed that their development teams had to work overtime before every release to address late test results. The online shop was released with many untested cases and, thereby, sometimes included issues. WomOut identified a need for (1) a live overview of test cases executed and pending execution, (2) flexible test case re-assignment, and (3) eliminating the manual and error-prone assembly of test cases.

Deployment of the artifact was later extended to the marketing department to support the creation of promotions. Promotions involved activities such as discounts which customers could redeem by entering a coupon code during checkout. Both internal and external requests for coupon codes were handled by the online marketing team via email. However, creating promotions requires the coordination of different stakeholders, which made this unstructured procedure infeasible for WomOut. Creating promotion codes required multiple systems, and three different departments (i.e., business intelligence, customer service, and finance) needed to be informed about the details.

3.2 Stage II: Building, Intervention Evaluation

The practitioners responsible for the internally used software in the ADR team took a bilateral role engaging more with the three researchers in the ADR team. The researchers suggested utilizing a BPMS to continuously improve their software testing process. The practitioner, who was placed among the researchers, raised the idea of extending the company's issue-tracking system *Jira* for use as a BPMS. Starting from this initial idea, we chose *Jira* as our BPMS for managing the software testing process. As *Jira* qualifies as a BPMS according to the generic architecture of a BPMS [10,15] and a BPMS's definition [2] it appeared as a satisficing entry point.

The ADR practitioners monitored the running processes. Our monitoring procedure is grounded on measuring different indicators related to quality-related goals defined by the WomOut. WomOut wanted to reduce the time to detect a bug measured from the point in time when a new feature was available to testers. WomOut also aimed to achieve higher test coverage before a new feature's deployment to productive instances of the online shop. The assembly of the overall test coverage was measured in the workdays of a full-time employee.

Three organization-dominant building intervention and evaluation cycles reciprocally shaped the artifact. We chose the organization-dominant approach to attend to the socio-technical nature of a BPMS. The first two cycles comprise building the BPMS for software testing, whereas the third phase arose during the project and focuses on the promotion code handling.

In the first cycle, the ADR team opened a new *Jira* project. Under the guidance of the researcher, the practitioners configured workflows and designed a notification scheme when new test cases were to be tested. Furthermore, they defined automation patterns consisting of business rules to assign certain software testing tasks to testers in different countries. The monitoring requirement of the testing coordinator was immediately available as the board view with different filtering options showed test cases were open and who should execute them. After rolling out the project to software testers, problems related to the workflow, the notifications scheme, and case re-assignment patterns arose and could be resolved quickly. Problems related to user management arose, as privacy and re-distribution of test cases still posed issues. Looking at the defined metrics for achieving WomOut's goals showed that the decrease in manual assembly of all test cases and test coverage were about the same as before the intervention.

In the second cycle, we focused on providing automation capabilities for the business process by extending the *Jira* instance with the plugin *Adaptivist ScriptRunner* for this purpose. The company opted for this extension to extend the automation and process modeling capabilities of *Jira*. With the help of the extension, WomOut extended *Jira*'s workflow screens, conditions, validators, and POST functions by a couple of predefined, configurable functions, as well as the ability to write custom code. The custom code uses *Jira*'s built-in functionalities, allowing the testing coordinator to read and manipulate any data related to individual cases, send emails, and make network requests.

The ADR team enabled, thereby, automating the assembly of the overall test case results. Additionally, a new type of view on a *Jira* project for monitoring the current state of the process instances. The ADR team also automated the initial distribution of test cases to testers in the according countries. Additionally, we established three user roles to aggravate surveillance and comparison to other software testers and to allow for extending the number of testers from different teams during run-time. After rolling out the changes in the business process, we saw that the automation reduced manual effort in distributing and assembling test cases. At this point, the ADR team had elaborated a BPMS, which fulfilled WomOut's expectations.

In the third cycle, we drew from the software-testing business process and adjusted a copy of it to fit the promotion code creation process. We could set up a workflow in which every promotion was created as one case in the *Jira* project. Event listeners were introduced to react to events across projects, jobs to run code periodically, and behaviors to manipulate the screens users see when creating, editing, or transitioning a case. Using the previous extension, the BPMS could automatically perform a duplicate and logical check. Moreover, notifying the three different departments could be enabled based on the event listeners. A checklist in the cases of the promotion code handling process cases for those systems where a specific promotion code needs to be added to work. The workflow made sure that everyone knew what to do and when.

3.3 Stage III: Reflection and Learning

Reflection and learning are characterized by guided emergence, which refers to deriving knowledge from a design and its use in an organizational context. The practitioners on the ADR team reported problems on the types of problems that arose during the use of the BPMS and reported if and how they could counteract these problems. On the contrary, if the researchers of the ADR team proposed adjusting the BPMS configuration based on theory or reasoning, the practitioners in the ADR team made the adjustments in the artifact. In quarterly meetings, the ADR team practitioners reported how the BPMS had been used by the software testers and if small operative problems occurred which had to be adjusted. Furthermore, we learned that generalizing the outcome could benefit from expanding the BPMS to further functions in the company. As WomOut saw the BPMS's success in software testing, the company wanted to expand the BPMS to other units. The online marketing department expanded the scope of the ADR project.

3.4 Stage IV: Formalization of Learning

The present study set out to derive a BPMS design following a socio-technical perspective that balances appropriateness for use in research and practice. We, therefore, follow Gregor et al.'s [11] anatomy of a design principle to generalize our findings (generalized outcome). In the following section, we present seven design principles for using BPMS as the outcome of our formalization and learning from the ADR project.

4 Design Principles

User Management. BPMSs distribute work based on digital user identity, permissions, and authorization. Process managers require permission to change business processes and manage their department's progress while IT personnel require read-permissions on process data to build dashboards for analyzing process performance indicators. Both process managers and IT personnel should be able to allocate users in groups and roles, and IT personnel can create roles, groups, and permissions. For instance, a software tester in an English-speaking country may access all test cases for the English online shop.

Process Modeling. A BPMS requires functionalities for modeling business processes that are adjustable and configurable during run-time [1,10]. The executable process models must be free of ambiguities and machine-readable [40]. Business rules must be added to executable process models and they must be machine-readable, ranging from conditions to automation [17,32]. The ability to adjust workflows and business rules during run-time [29] allows for quick adaptations to process requirements, enabling cases contradicting the refined model to be migrated to a compliant new execution route.

Automation. Automation capabilities are considered a key driver for using BPMS [36]. We found that automation should consider at least two triggers to be helpful: (1) based on case-related conditions and (2) based on a fixed schedule. BPMS and connected systems can execute automated actions (e.g., notifications, report generation, or input validation) directly in their environment.

Logging. Users may want to log processed cases for different reasons, for example, task handovers, process debugging, performance measurement, compliance checking, auditing, and process analysis. In our ADR project, logging facilitated reporting software-testing results and tracing a promotion code's status. As promotion codes had to be entered into different systems, a case log contained which systems already contained the promotion code.

Monitoring. Monitoring enables an organization to gain transparency about their running processes [16]. Managers and IT personnel can, for instance, set up dashboards based on logged data or the current state of process instances. The monitoring capabilities allow for the creation of personalized views on the process state. For example, testing coordinators, online marketing, and process workers required different views on a business process.

Integration. Business processes span across functions of an organization [39]. Process workers need to work in different IT systems. System integration enables the delegation of tasks and case state retrieval from other IT systems (e.g., the promotion code input in our ADR project).

Case Handling. A process participant's daily work depends on handling cases [2]. All required information for case handling must be associated with a case, e.g., via forms, files, or integration of another system [9]. To execute business processes, BPMS must allow process workers to complete their work (i.e., process their cases). A key concern of BPM in case processing is the flexibility of processes [1]. *A priori* flexibility allows process workers to select different execution paths of a process model [29]. Occasionally, users need to edit cases to achieve further flexibility. A BPMS facilitates collaboration between users, for example, by passing work to a colleague or leaving comments on a task [23]. A BPMS's user interface allows users to interact with their cases [2]. Users identify cases based on certain properties. In our ADR project, that referred to, for instance, test cases belonging to a tester, which have been executed more than once, and previously revealed an issue at least once.

Table 1 summarizes all our design principles.

Table 1. Design Principles for Using BPMS.

DP1: User Management	Aim, implementer, user	Allow managers (*user*) to create and delete users and allow IT personnel to control access to assets and tasks via roles and permissions (*aim*)
	Context	An organization wants to execute its business processes in a BPM to achieve organizational excellence
	Mechanism	Create users with roles and permissions in a company's user management or a likewise system that is mutable by managers. The system should comprise at least three roles: manager, process worker, and administrator
	Rationale	Because controlling user access prevents unauthorized read or write operations, it decreases the risks, such as errors and compliance issues
DP2: Process Modeling	Aim, implementer, user	Allow managers, IT personnel or process workers (*implementers*) to intuitively create business process models and business rules (*aim*)
	Context	An organization wants to execute its business processes in a BPMS to achieve organizational excellence
	Mechanism	Provide means for storing and changing representations of a business process (e.g., flow objects, informational objects, connections, artifacts, or logic) via (pseudo-)code or graphical elements. Make process models changeable during run-time and support migration between different execution paths
	Rationale	The system can interpret business process models and execute activities according to the predefined sequences of activities. The process model includes information on required resources, involved stakeholders or execution conditions. Process modeling facilitates structured operations and automation
DP3: Automation	Aim, implementer, user	Allow process workers (*users and implementers*) to create automated actions (*aim*) based on rules, conditions or a schedule
	Context	An organization wants to execute its business processes in a BPMS to achieve organizational excellence
	Mechanism	Provide an interface to define (1) case-related conditions or (2) schedules to trigger automated execution of the process
	Rationale	Automation capabilities are key drivers for organizations using a BPMS
DP4: Logging	Aim, implementer, user	Log the processing of cases for future processing (*aim*) by process workers (*users*), e.g., for task handovers, workflow debugging, compliance checking, auditing, and process analysis
	Context	In the backend of the system
	Mechanism	Ensure that every activity the system performs is documented and stored persistently with enough detail
	Rationale	Logs serve as input for further analyses, software testing or other supporting activities
DP5: Monitoring	Aim, implementer, user	A BPMS enables the analysis of process-performance-related (user or pre-defined) indicators and the current state of cases (*aim*) by managers and process workers (*users and implementers*)
	Context	An organization wants to execute its business processes in a BPMS to achieve organizational excellence
	Mechanism	Provide an interface that allows viewing and defining performance indicators and the current state of work based on logged or case data
	Rationale	Process workers and managers use dashboards to react to information at run-time, which is essential for controlling knowledge-intensive processes and informing business decisions
DP6: Integration	Aim, implementer, user	Allow IT personnel (*implementer*) to integrate other programs or external services for extending the system's usefulness (*aim*) for process workers and managers (*users*)
	Context	An organization wants to execute its business processes in a BPMS to achieve organizational excellence
	Mechanism	Provide interfaces that facilitate integration with other systems or functions, for example, via data integration, service-oriented architecture or REST APIs
	Rationale	Business processes usually span across multiple functions of an organization. For end-to-end support of business processes, integration is required to enable the delegation of tasks and retrieve case information from other IT systems
DP7: Case Handling	Aim, implementer, user	Allow cases to be created and processed by process workers (*user*) and enable them to work flexibly on an identifiable task (*aim*)
	Context	An organization wants to execute its business processes in a BPMS to achieve organizational excellence
	Mechanism	Provide functionalities to identify and edit cases, to add information to cases, such as files or forms, and to work on cases collaboratively, such as by using a worklist handler for prioritization or case assignment to process participants
	Rationale	Business processes can be flexible and need to be identified and processed by process workers (sometimes collaboratively)

5 Discussion and Conclusion

The present study explores how BPMS features can aid operations in organizations. We aimed to generate prescriptive knowledge on BPMS design from a socio-technical perspective that accounted for the relationship between artifact features and users and which supports organizations in applying the design knowledge in practice. We subsequently followed Gregor et al.'s anatomy of a design principle that links IT artifacts and users through functional relationships and aims at a level of presentation that is "understandable and useful in real-world design contexts" [11]. Our design principles contribute to the literature on BPMS design which has been dominated by technology-oriented studies targeted at computer scientists [26]. While our focus is on portraying the incremental reconfiguration of a company's ITS to a bespoke BPMS, it also provides some insights into BPM literature in general. IT affects how organizations execute and coordinate activities in a process. It influences an organization's capabilities to automate tasks, gain insights through monitoring and analyzing processes, and change its coordination mechanisms [22]. Opportune use of technology is therefore integral to BPM success [39]. Organizations should not put too much thought into finding the best system which fits certain system types or functionality "labels" but instead prioritize matching their employees' needs about BPM-related tasks (i.e., diagnosis, process design, system configuration, and process enactment) [2] with what IT functionally affords them to do. As system functionality and extendability increase and as their autonomy and agency increase [4,22], more and more systems can support multiple tasks without being explicitly designed to do so. Analyzing an IT artifacts material properties alone will prove increasingly difficult and insufficient. Instead, future research on IT supporting BPM can benefit from also looking at generative mechanisms of artifacts that emerge in relation to the users [11,38] and that could give rise to process dynamics in the organization and digital innovation [12,25].

The findings of this study have to be seen in light of some limitations. Our key limitation stems from the design of our research as ADR study. This paper aimed to derive generalizable knowledge (design principles) from one case [31]. The application in other firms, however, may not be equally relevant or feasible. The studied firm had a running issue tracking system, *Jira*, to build on and could capitalize on internal expertise with respect to using *Jira* to support workflows. However, other organizations using different IT systems could require more effort to enact the design principles, and extending and adapting these systems to function as BPMS might prove resource intensive. Furthermore, the involvement of the researchers introduced a bias towards searching for BPM-related solutions. While we made every decision to the best of our knowledge, other solutions to the company's problem outside of BPM research could have been more suited to inform the company's IT design. Future research could investigate other industries or company sizes and if these contextual factors influence the design of IT systems for supporting BPM. The present study indicates that organizations could explore their existing IT systems for supporting BPM activities. For the debate to be moved forward, a better understanding regarding the impact of the extendability and level of appropriation of existing IT on effective use and BPM success needs to be developed.

References

1. van der Aalst, W.M.P.: Business process management: a comprehensive survey. ISRN Softw. Eng. 1–37 (2013)
2. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business process management: a survey. In: van der Aalst, W.M.P., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-44895-0_1
3. van Aken, J.E.: Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. J. Manag. Stud. **41**(2), 219–246 (2004)
4. Baird, A., Maruping, L.M.: The next generation of research on is use: a theoretical framework of delegation to and from agentic is artifacts. MIS Q. **45**(1), 315–341 (2021)
5. Baskerville, R.: Investigating information systems with action research. Commun. Assoc. Inf. Syst. **2**, 1–32 (1999)
6. Baumgraß, A., et al.: Towards a methodology for the engineering of event-driven process applications. In: Reichert, M., Reijers, H.A. (eds.) BPM 2015. LNBIP, vol. 256, pp. 501–514. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42887-1_40
7. Coenen, T., et al.: An information system design theory for the comparative judgement of competences. Eur. J. Inf. Syst. **27**(2), 248–261 (2018)
8. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. J. Data Semant. **4**(1), 29–57 (2015)
9. Djedovic, A., Zunic, E., Alic, D., Omanovic, S., Karabegovic, A.: Optimization of the business processes via automatic integration with the document management system. In: Proceedings of the 2016 International Conference on Smart Systems and Technologies, pp. 117–122. IEEE (2016)
10. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, Berlin, Heidelberg (2018). https://doi.org/10.1007/978-3-662-56509-4
11. Gregor, S., Chandra Kruse, L., Seidel, S., et al.: Research perspectives: the anatomy of a design principle. J. Assoc. Inf. Syst. **21**(6), 1622–1652 (2020)
12. Grisold, T., vom Brocke, J., Gross, S., Mendling, J., Röglinger, M., Stelzl, K.: Digital innovation and business process management: opportunities and challenges as perceived by practitioners. Commun. Assoc. Inf. Syst. **49**(1), 556–571 (2021)
13. Hasan, B.: Examining the effects of computer self-efficacy and system complexity on technology acceptance. Inf. Resour. Manag. J. **20**(3), 76–88 (2007)
14. Hevner, M., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28**(1), 1–33 (2004)
15. Hollingsworth, D.: The workflow reference model. workflow management coalition TC00-1003 (1995)
16. Kirchmer, M., Laengle, S., Masias, V.: Transparency-driven business process management in healthcare settings [Leading Edge]. IEEE Technol. Soc. Mag. **32**(4), 14–16 (2013)

17. Knolmayer, G., Endl, R., Pfahrer, M.: Modeling processes and workflows by business rules. In: van der Aalst, W., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 16–29. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45594-9_2
18. Lange, M., Mendling, J., Recker, J.: An empirical analysis of the factors and measures of enterprise architecture management success. Eur. J. Inf. Syst. **25**(5), 411–431 (2016)
19. Lawrence, P.: Workflow Handbook, 1 edn. Wiley, Hoboken (1996)
20. Liu, D., Santhanam, R., Webster, J.: Toward meaningful engagement: a framework for design and research of gamified information systems. MIS Q. Manag. Inf. Syst. **41**(4), 1011–1034 (2017)
21. Markus, M., Silver, M.: A foundation for the study of IT effects: A new look at DeSanctis and poole's concepts of structural features and spirit. J. Assoc. Inf. Syst. **9**(10), 1–26 (2008)
22. Mendling, J., Decker, G., Hull, R., Reijers, H., Weber, I.: How do machine learning, robotic process automation, and blockchains affect the human factor in business process management? Commun. Assoc. Inf. Syst. **43**(1), 297–320 (2018)
23. Mendling, J., Recker, J.C., Wolf, J.: Collaboration features in current BPM tools. EMISA Forum **32**(1), 48–65 (2012)
24. Morana, S., Kroenung, J., Maedche, A., Schacht, S.: Designing process guidance systems. J. Assoc. Inf. Syst. **20**(5), 6 (2019)
25. Pentland, B., Vaast, E., Wolf, J.R.: Theorizing process dynamics with directed graphs: a diachronic analysis of digital trace data. MIS Q. **45**(2), 967–984 (2021)
26. Pourmirza, S., Peters, S., Dijkman, R., Grefen, P.: A systematic literature review on the architecture of business process management systems. Inf. Syst. **66**, 43–58 (2017)
27. Pourmirza, S., Peters, S., Dijkman, R., Grefen, P.: BPMS-RA: a novel reference architecture for business process management systems. ACM Trans. Internet Technol. **19**(1), 13:1–13:23 (2019)
28. Sarker, S., Chatterjee, S., Xiao, X., Elbanna, A.: The sociotechnical axis of cohesion for the IS discipline: its historical legacy and its continued relevance. MIS Q. **43**(3), 695–719 (2019)
29. Schonenberg, H., Mans, R., Russell, N., Mulyar, N., van der Aalst, W.: Process flexibility: a survey of contemporary approaches. In: Dietz, J.L.G., Albani, A., Barjis, J. (eds.) CIAO!/EOMAS -2008. LNBIP, vol. 10, pp. 16–30. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68644-6_2
30. Schoormann, T., Stadtländer, M., Knackstedt, R.: Designing business model development tools for sustainability-a design science study. Electron. Mark. **32**, 1–23 (2021)
31. Sein, H.: Purao, Rossi, Lindgren: action design research. MIS Q. **35**(1), 37–56 (2011)
32. Shaw, D.R., Holland, C.P., Kawalek, P., Snowdon, B., Warboys, B.: Elements of a business process management system: theory and practice. Bus. Process. Manag. J. **13**(1), 91–107 (2007)
33. Stierle, M., Zilker, S., Dunzer, S., Tenscher, J.C., Karagegova, G.: Design principles for comprehensible process discovery in process mining. In: Proceedings of the European Conference on Information Systems (2020)
34. Szelągowski, M., Berniak-Woźny, J., Lupeikiene, A.: The direction of the future development of ERP and BPMS: towards a single unified class? In: Ivanovic, M., Kirikova, M., Niedrite, L. (eds.) Digital Business and Intelligent Systems. Baltic

DB&IS 2022. CCIS, vol. 1598, pp. 111–124. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-09850-5_8

35. Tallon, P., Queiroz, M., Coltman, T., Sharma, R.: Business process and information technology alignment: construct conceptualization, empirical illustration, and directions for future research. J. Assoc. Inf. Syst. **17**(9), 563–589 (2016)
36. Trkman, P.: The critical success factors of business process management. Int. J. Inf. Manag. **30**(2), 125–134 (2010)
37. Veldhuizen, R., van Ravesteijn, P., Versendaal, J.: BPMS implementations in SMEs: exploring the creation of a situational method. In: Proceedings of the 25th Bled eConference eDependability: Reliable and Trustworthy eStructures, eProcesses, eOperations and eServices for the Future (2012)
38. Volkoff, O., Strong, D.M.: Critical realism and affordances: theorizing IT-associated organizational change processes. MIS Q. **37**(3), 819–834 (2013)
39. vom Brocke, J., Schmiedel, T., Recker, J., Trkman, P., Mertens, W., Viaene, S.: Ten principles of good business process management. Bus. Process. Manag. J. **20**(4), 530–548 (2014)
40. Weber, I., Hoffmann, J., Mendling, J.: Beyond soundness: on the semantic consistency of executable process models. In: 2008 Sixth European Conference on Web Services, pp. 102–111 (2008)
41. Weske, M., van der Aalst, W., Verbeek, H.: Advances in business process management. Data Knowl. Eng. **50**(1), 1–8 (2004)

Towards a Measurement Instrument for Assessing Capabilities When Innovating Less-Structured Business Processes

Joklan Imelda Camelia Goni(✉) and Amy Van Looy

Department of Business Informatics and Operations Management, Faculty of Economics and Business Administration, Ghent University, Ghent, Belgium
`Joklanimeldacamelia.Goni@UGent.be`, `Amy.vanlooy@Ugent.be`

Abstract. Many contemporary organizations deal with different types of business processes, including less-structured business processes (LSBP). The latter are characterized by more unpredictable situations, ad-hoc tasks, limited process details, and a stronger emphasis on the human aspects as compared to the more structured business processes (SBP). However, existing BPM approaches often encounter challenges when driving innovation within organizations that have LSBP characteristics. To address this issue, organizations still lack a measurement tool to assess their process innovation capability (PIC) that is tailored towards LSBP (as opposed to generic BPM measurement instruments). Our study fills this gap by gradually building and testing a PIC-LSBP measurement instrument. We rely on empirical data from an international Delphi study, followed by a demonstration in two real-life organizations (i.e., one manufacturing organization in Asia and one service organization in Europe). The study aims at showing the practical application of the uncovered measurement instrument, seeking to identify and effectively address process innovation in more dynamic and complex business environments. The resulting instrument consists of six main capability areas, 18 sub-areas, and 55 formative measurement items. The findings highlight the significance of leveraging an instrument as a means to assess an organization's current innovation capability with respect to a specific process type. Based on this insight, our research contributes to a broader understanding of how organizations can assess their existing process situation and derive action plans.

Keywords: Delphi · illustrative case study · process innovation capability · less-structured business processes · brainstorming tool · measurement instrument

1 Introduction

Organizations typically strive to continually innovate their business processes [1] for enabling effective operations, enhancing productivity, increasing operation efficiency, streamlining processes, and delivering superior products or services [2, 3]. While this struggle exists in different business contexts [4], it remains relevant in more dynamic business environments that are usually characterized by less-structured business processes (LSBPs) [5, 6]. So far, business process management (BPM) as a discipline

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 229–240, 2024.
https://doi.org/10.1007/978-3-031-50974-2_18

especially provides tools, methods, frameworks and measurement tools for analyzing, designing, implementing, and continuously improving the more structured business processes [5]. However, despite BPM's widespread adoption, organizations often encounter challenges and fail in endeavors when attempting to apply structured BPM practices in diverse business contexts with LSBP [7]. Although some methods and tools have appeared that are more geared towards LSBP (e.g., case management), a maturity model or measurement instrument that comprehensively assesses the required capability areas when innovating LSBP is still lacking.

One of the main reasons for the lower success rates of BPM in LSBP is because traditional BPM approaches, methods, and models are less suitable to accommodate to diverse requirements of businesses operating in dynamic business contexts [7], and particularly those involving LSBP [8–10]. LSBP characteristics are heavily affected by agility and rely on people (i.e., as process participants and process improvement team members) in order to respond to dynamic market conditions for their organization to embrace change and emerging opportunities [11]. This includes acknowledging that each organization operates in a unique business context with varying market dynamics, changing regulations [12], as well as changes in customers' attitudes, technology disruption, or natural disasters [13].

To overcome these challenges, there is a need to develop a dedicated BPM measurement instrument for accommodating the diverse needs of organizations and aligning with different business context factors that typify LSBP [7, 14, 15] in order to help organizations build their process innovation capability (PIC). The purpose of this study is to address the following research question:

- RQ. How to measure an organization's capabilities for innovating less-structured business processes?

Our intended measurement instrument will be developed through an international Delphi study and tested by two real-life cases. The instrument helps organizations acquire an overview of their existing (AS-IS) situation and serves as a foundation for a related gap analysis to enhance their PIC in LSBP. Our main scientific contribution implies adding a novel artefact to the body of knowledge since (to the best of our knowledge) there is currently no specific measurement instrument available for managing PIC in LSBP.

2 Theoretical Background

2.1 Underlying Theories

This study combines the theoretical foundation of RBV and dynamic capabilities to better understand the need for PIC in LSBP. More specifically, in rapidly evolving business environments, organizations need to deal with different types of business processes, including those that are less structured. LSBPs are characterized by a higher likelihood of encountering unpredictable situations, by ad-hoc tasks and by having a rather limited availability of detailed flows [8–10], but also by more complex, dynamic, interactive, socially constructed, and more people-centric than typical structured business processes

(SBP) [11]. Such business processes frequently involve user collaboration, and contextual changes may necessitate at execution [9] to adjust to specific circumstances [16]. Moreover, organizations with LSPB characteristics often face challenges in driving innovation. We see process innovation is a key driver of organizational success for enabling the organization to operate efficiently and effectively by encompassing an organization's knowledge, resources, and strategies [17–19]. Hence, PIC is defined as "*a firm's ability to acquire, assimilate, transform, and exploit technically related resources, procedures, and knowledge for process innovation purposes*" [20]. By cultivating strong innovation capabilities in LSBP, organizations have acknowledged that they can gain a sustainable competitive advantage in dynamic business environments [21, 22]. However, to effectively harness their PIC in LSBP, organizations also need to align their innovation capabilities with their unique resources, as emphasized by RBV theory.

RBV theory emphasizes the importance of leveraging an organization's unique resources and capabilities to gain competitive advantages [23]. For instance, these resources provide an organization with a competitive advantage in developing creative solutions and adapting to unforeseen challenges. Additionally, a dynamic capability extends the RBV concept by emphasizing the optimization or development of resources and capabilities in a more dynamic business environment [24–26]. In LSBP, dynamic capabilities thus provide organizations with the ability to learn, reconfigure resources, and sense and seize new opportunities to navigate through uncertain and rapidly change environments while remaining agile and innovative in their business processes.

2.2 BPM-Related Capabilities

Numerous models and approaches (such as maturity models) have been proposed to develop capability frameworks and measurement instruments in the field of BPM. For example, the Capability Maturity Model Integration (CMMI) provides a framework and measurement instrument for assessing and improving process maturity levels [27]. Alternatively, the Business Process Maturity Model (BPMM) assesses an organization's process maturity across multiple dimensions [28]. Previous studies have also described different dimensions of BPM capability areas. For instance, de bruin and Rosemann [29] have identified six competence categories, while McCormack and Johnson [30] propose three categories. In a meta-analysis, Van Looy et al. [31] have constructed a conceptual model by comparing 69 maturity models, after which Van Looy [32] has developed a measurement instrument for capabilities when managing business process. Although such models offer valuable insight and understanding of BPM capabilities and their measurement, they are not finetuned to LSBP nor recognize the specifics of dynamic business contexts.

Alternatively, to operationalize the capability areas of PIC in LSBP, this study builds upon a recent literature review that has proposed a combination of the three-pillar framework of "People-Process-Technology" (PPT) [33], along with an organizational, knowledge-related and ecosystem pillar. In sum, despite extensive research in BPM, there is still a gap in the body of knowledge about how to measure specific innovation capabilities in LSBP and about how to put it into practice [32].

3 Research Methods

Our research methods were in line with the Information Systems (IS) literature ([34–36] in the sense that we started our study from the body of knowledge. More specifically, we found a recent systematic literature review (SLR) [37] that provided use with an initial capability framework for understanding the PIC-LSBP topic. Building upon this knowledge, a three-round international Delphi study was conducted to validate the proposed framework [38, 39]. Next, to operationalize the framework, an extended and empirically validated capability framework was concretized into binary measurement questions that are formative in nature. Finally, we employed an illustrative case study approach in two real-life organizations, representing both the manufacturing and service sector in Asia and Europe to capture diverse practices when assessing and innovating LSBP. The overall research model depicted in Fig. 1 provides a visual representation of the study.

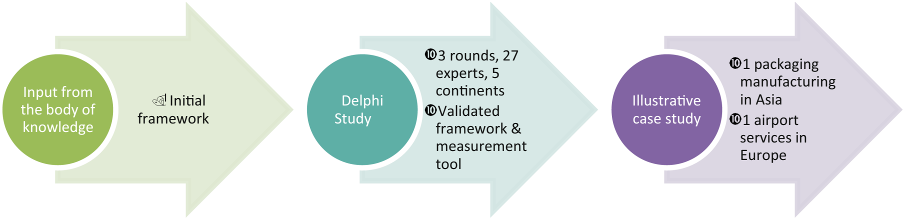

Input from the body of knowledge
Initial framework
Delphi Study
3 rounds, 27 experts, 5 continents
Validated framework & measurement tool
Illustrative case study
1 packaging manufacturing in Asia
1 airport services in Europe

Fig. 1. Research model underlying this study.

3.1 Delphi Study for Instrument Development

We conducted a modified Delphi approach, following the guidelines of [39] and [38] to decide on expert selection, expert satisfaction evaluation [6], consensus criteria [40], and validity and reliability [41]. This research phase took nine months in total. We carried out a three rounds of expert interviews, including 27 experts from both academia and practitioners to ensure a balance of views from both theoretical and practical perspectives [29] using LinkedIn to contact experts. The scholars were required to have a PhD and had published at least two academic papers in reputable peer-reviewed journals. As for the practitioner-oriented participants, we sought senior managers with a minimum of 7 years of experience in the BPM field and working within medium- to large-sized organizations.

The measurement tool was developed over three Delphi rounds by gradually integrating each capability based on the initial definitions derived from the SLR and extended with experts' inputs. Throughout the study, the experts were asked to comment on the capability coverage and the measurement items. Additionally, we also asked the experts to evaluate the framework's overall coverage to ensure comprehensiveness and for combining rigor with relevance.

3.2 Two Illustrative Case Studies for Instrument Testing

A careful case selection was based on specific selection criteria: (1) involving different sectors to capture different industry settings, (2) working with LSBPs, (3) having prior experience with innovating such LSBPs, and lastly (4) operating in an international or at least regional operational setting to recognize cross-cultural differences. Such criteria helped ensure that the selected organizations had already actively engaged in innovation within LSBP, while their diversity would help represent the global applicability of our instrument to provide a robust evaluation of the measurement tool.

Besides showing a demonstration, the purpose of conducting a comparative analysis was to gain valuable insight into industry-specific strategies, best practices, lessons learned, and adaptability to the specific contextual condition [42] of their business processes. By having these two illustrative case studies for which senior managers acted as respondents, we were able to identify not only the commonalities that drive successful process innovation and foster adaptability in terms of LSBP capabilities, but differences may also exist due to the unique challenges and dynamics within each industry [43]. An examination of commonalities and differences was thus deemed valuable [44], and provided us with the possibility of deriving actionable recommendations that can enhance process innovation across various settings, and ultimately partly contributing to organization's competitiveness in a rapidly evolving market.

Our demonstration specifically focused on preparing the instrument for wider use in the sense of a pilot test. The measurement tool was not prescriptive in nature and did not yet intend to assign specific maturity levels in its current form. Hence, its scope differs from typical maturity models.

To develop the measurement tool for our formative constructs, we followed the established research methods of Peter et al. [35]. First, we arranged a one-hour online video meeting (using Zoom or MS Teams) to present a comprehensive overview of the research project and to introduce the measurement instrument using MS PowerPoint and MS Excel document formats. Afterwards, we inquired to what extent the organization had prior experience with innovating LSBP in order to further orient the respondents to the scope of the measurement instrument. We offered the organization different options to fill in the tool together with the researcher, but they preferred to complete the questionnaire separately without guidance. Hence, we conducted two follow-up sessions, each lasting two to three hours, during which a thorough review and discussion of their responses took place. The interviews were conducted between April and May 2023. During this period, the first researcher also responded to any additional questions via e-mail, ensuring clear information on the organization's practice. For the manufacturing company, one interviewee provided the required input, while for the service company, there was one main interviewee who also involved other relevant stakeholders when filling out the measurement instrument in order to jointly review and discuss the organization's AS-IS situation.

4 Result

The Delphi results were used as a starting point for further case studies in real-life business scenarios.

4.1 Delphi Results

The measurement instrument covers a total of six main capabilities, 18 sub-capabilities and 55 measurement items that organizations can use. After the first Delphi round, we gathered insight on the differences between SBP and LSBP. However, as from the second round, the experts expressed the need for a practical demonstration of how the capabilities could be implemented within concrete organizations. As a response to this feedback, we prepared a practical measurement instrument that incorporated all found capabilities (https://osf.io/e2vrd/?view_only=2c204401575c4387833aca8f745445c3). Although each item can be measured using a 5-point Likert scale (1 = very unlikely; 5 = very likely), the current instrument deliberately turned to a binary scale (absent versus present) to act more as a brainstorming instrument within the organization, without necessarily being associated with the idea of a maturity model that emphasizes a diversity of maturity levels (e.g., for the purpose of benchmarking or certification). During the third and final round, the average experts' overall satisfaction score had increased from 7.52 out of 10 in the second round to 8.36 out to 10 in the third round.

4.2 Illustrative Case Study Results

After consolidating the input gathered from both manufacturing and service organizations, their assessment results were visualized using a spider chart in Fig. 2 and Fig. 3, respectively. These charts served as a visual presentation of the consolidated data on the level of the six main capability areas and offer a clear overview of the current state (AS-IS) within each organization. The percentages reflect capability coverage, namely the number of items being present (i.e., with a yes-score) within a certain capability area. For instance, 100% would indicate that all measurement items within a certain capability area are covered. For further use, similar charts can also be drawn on the level of the 18 sub-capabilities or even per measurement item.

We first present an overall analysis of both charts, after which we delve deeper into a comparison of both cases. More specifically, for the manufacturing company, the analysis of Fig. 2 revealed a balance between the capability areas of Employee skills and Process-specific features at 60%, followed closely by the Entrepreneurial strategy with coverage at 56%. The IS-related area accounted for 44% of its overall capability, whereas Network external relationship was situated at 33% of the measurement items in this area. Finally, Knowledge management was represented as the lowest capability with a coverage of 22%. In contrast, the service organization demonstrated a slightly different distribution of capability coverage. Employee skills stood out with the highest percentage, reaching 70% of capability coverage. IS and Knowledge Management reached the same percentage, each with 67% coverage, and closely followed by Process-specific features at 60% of the items covered. Although the Entrepreneurial strategy lagged at 33% of its overall capability, there was currently no established Network of external relationship (0%).

Furthermore, we structured the results using the PIC-LSBP capability framework resulting from the Delphi study for incorporating another view in terms of initial pillars (i.e., PPT, organization-related, knowledge-related and ecosystem pillars). This analysis

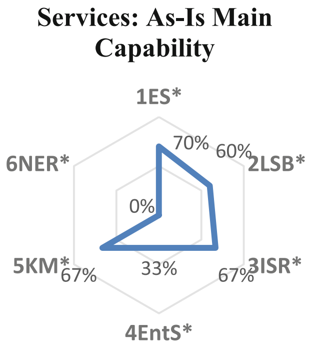


Fig. 2. Packaging manufacturing in Asia

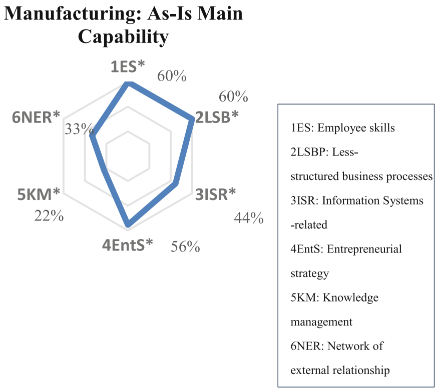


Fig. 3. Airport services in Europe

allowed us to uncover interesting differences between the service organization and the manufacturing organization.

First, in terms of PPT pillar, the service organization exhibited a higher capability in Employee skills (70%), potentially indicating their emphasis on knowledge-intensive workers that require high-quality services to meet diverse customer needs. On the other hand, the manufacturing organization demonstrated a more balanced capability in Employee skills and LSBP (60%), indicating their focus on adaptability and flexibility to respond quickly to market changes and to optimize production efficiency. Moreover, the service organization emphasized IS-related aspects (67%), reflecting their focus on leveraging technology to enhance service delivery through efficient data management. During the discussion, the expert also mentioned that each technician in the field is equipped with tablets to streamline the business process of updating work progress per activity. This technological integration allowed for real-time data recordings along with more efficient and immediate monitoring of the tasks performed by the technician.

Next, within the Organization-related pillar, the service organization demonstrated a higher capability in Knowledge management (67%), which was due to their recognition of capturing and leveraging organizational knowledge in delivering high-quality services and fostering continuous improvement. On the other hand, the manufacturing organization had a higher capability in Entrepreneurial strategy (56%), indicating its proactive and innovative approach for pursuing growth opportunity. During the discussion, the senior manager highlighted a shift in customer expectations. Customers would initially simply request what they wanted, while the company is being asked to produce proposals for more efficient product creations. This increased focus within the manufacturing sector highlights the need to identify new market trends, technologies, and business opportunities.

Finally, within the Ecosystem pillar, the manufacturing company exhibited a higher score regarding Network of external relationships (33%), indicating their established collaborations with supply chain partners. This observation was driven by the manufacturing industry's reliance on external partnerships for sourcing materials, expanding market reach, and optimizing logistics. During the discussion, the senior expert provided an example in which a supplier asked them to try their materials in the factory, and in return, they would collaborate for joined sales. On the other hand, this aspect was completely absent in the service organization's capability area, leaving room for further investigation and potential opportunities for the service organization to develop external relationships.

The goal of this AS-IS PIC-LSBP analysis was not to achieve complete 100% coverages across all capability area within each organization, but rather to encourage a more systematic way of thinking about all potentially critical areas that are aligned with an organization's strategy. This demonstration thus provided valuable insight into the current state (AS-IS) of both organizations and can serve as a basis for future improvement and strategic decision making within each organization.

5 Discussion

Our main contribution is a comprehensive PIC-LSBP framework along with a related measurement instrument. Next, we present the theoretical implications (Sect. 5.1), practical implications (Sect. 5.2), research limitations and avenues (Sect. 5.3).

5.1 Theoretical Implications

Two key findings arise when comparing the existing literature with our identified capability areas for measuring process innovation in LSBP. First, the existing studies on BPM and innovation have not adequately captured the unique characteristics and requirements of LSBP [7], leading to the research gap that we have tackled to some extent. While we have refined the PIC-LSBP capability overview and further concretized it in a measurement approach, more research is still needed to address the growing importance of LSBP in order to better accommodate to dynamic and complex business environments. Secondly, while various capabilities from different dimensions, including competence categories, have been identified in the existing literature, their empirical evidence remains limited [32], and to which we have contributed by building a practical instrument together with experts and practitioners.

Our research has offered a comprehensive PIC-LSBP framework along with a related measurement instrument. This instrument incorporates the well-known PPT pillar [33], as well as other organization-related, knowledge-related and ecosystem pillars, specifically tailored to meet the needs of measuring innovation capabilities in LSBP. While these pillars are aligned with the literature review that we considered as our starting point, we have further validated and finetuned them in terms of main and sub-capability areas. Moreover, the translation of this empirically validated framework into a practical measurement instrument facilitates a more comprehensive approach to capture the LSBP-related characteristics and enable organizations to systematically assess their innovation capabilities. For instance, the application of this measurement instrument in two cases from different sectors suggests some generalizability of our assessment approach and broadens its applicability in practice. It offers valuable insights into their strengths and potential areas for improvement within a varied set of main PIC-LSBP capability areas for both sectors to leverage and further develop their competitive edge.

5.2 Practical Implications

The measurement instrument available in this paper is open for other LSBP-related organizations that wish to innovate their business processes and would like to assess their current capability coverage. Simultaneously, identifying areas of improvement will enable organizations to prioritize resources and initiatives aimed at enhancing those capabilities. A comprehensive understanding of all PIC-LSBP capabilities can help organizations make well-thought strategic decisions and guide their efforts to enhance their most important capabilities and positioning organizations for sustainable growth in dynamic business environments. For instance, the insight derived from spider graphs (such as Fig. 2 and Fig. 3) can be leveraged to brainstorm about and derive more step-by-step guidance for a specific organization. In our systematic approach for conducting PIC-LSBP assessments, an organization can first utilize the instrument to answer the measurement items in a comprehensive manner. Secondly, the organization can utilize the assessment results to draw a spider graph for visually presenting the current (AS-IS) best practices across different capability areas. This graph intends to provide a clear overview of current strengths and weaknesses, and thus purely facilitating a better understanding of the overall assessment outcomes. Thirdly, the organization can critically analyze the spider graph to assess whether the current (AS-IS) situation is optimal or if there are areas that require improvement. Lastly, by identifying improvement areas (e.g., in a workshop with distinct roles), the organization can strategically decide to focus their improvement efforts on enhancing certain areas (e.g., currently weaker aspects) and to think about an action plan. Hence, the provided measurement instrument can help realize a more systematic and effective assessment that closely aligns with the organization's specific objectives and business context.

5.3 Research Limitations and Future Research Avenues

Our phased research approach is part of larger design-science research (DSR) study with several other iterative phases to further build and test the measurement instrument as a valid and reliable instantiation or tool artefact. The Delphi study represents a first iteration, while the illustrative case studies constitute a second iteration. We will continue with applying and evaluating different versions of the measurement tool in subsequent DSR iterations. Meanwhile, the major limitation of our measurement instrument is that its purpose is descriptive instead of being prescriptive in nature, which means that we do not intend to assign specific maturity levels and we do not measure its performance outcomes. At this moment, the instrument focuses on assessing current situations, allowing flexibility without imposing predefined levels to organizations. However, its descriptive nature might also limit the tool's ability to provide specific guidance for those organizations seeking a more prescriptive aid to progress their journey. Additionally, the tool has been built in an MS Excel format, which might present potential challenges for users in terms of usability and data entry. To address these limitations, future research can focus on enhancing the measurement instrument by incorporating a prescriptive component such as maturity level guidance with performance measurement, while also exploring alternatives for obtaining more user-friendly formats.

6 Conclusion

This study has offered a major step towards the creation of a valid and reliable measurement instrument for assessing PIC in LSBP, providing a valuable tool for both BPM scholars and practitioners. Our instrument helps enhance the BPM discipline by strengthening its theoretical foundation about an under-researched subject and fostering future empirical BPM research. The instrument includes six main capabilities, 18 subcapabilities and 55 measurement items. While these capability areas comprehensively cover the required areas for innovating LSBP, the measurement items are formulated in a generic way for ensuring their usability across different audiences and business contexts. By utilizing this measurement tool, organizations can identify and evaluate their current capabilities before launching process innovation projects. The systematic approach provided by the measurement tool facilitates a comprehensive reflection on an organization's existing best practices in LSBP to gain a clear overview of their current situation (AS-IS) and assess their strengths and weakness more accurately while also determining the future capabilities required to achieve successful innovation in LSBP. The next step involves conducting a gap analysis between the current state and the desired future state. Based on this analysis, an organization can concretize an action plan that aligns their strategic initiatives with their innovation goals in order to make their LSBP more future proof, while also considering their business context.

References

1. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30409-5

2. Zimmermann, R., Ferreira, L.M.D.F., Carrizo Moreira, A.: The influence of supply chain on the innovation process: a systematic literature review. Supply Chain Manage.: Int. J. **21**(3), 289–304 (2016)
3. Jansen, J.J.P., Van Den Bosch, F.A.J., Volberda, H.W.: Exploratory innovation, exploitative innovation, and performance: effects of organizational antecedents and environmental moderators. Manage. Sci. **52**(11), 1661–1674 (2006)
4. Harmon, P., Wolf, C.: The state of business process management 2014. BPTrends (2014)
5. Dumas, M., et al.: Introduction to business process management. Fundam. Bus. Process Manage. 1–33 (2018)
6. Kerpedzhiev, G.D., et al.: An exploration into future business process management capabilities in view of digitalization. Bus. Inf. Syst. Eng. **63**(2), 83–96 (2021)
7. Vom Brocke, J., Zelt, S., Schmiedel, T.: On the role of context in business process management. Int. J. Inf. Manage. **36**(3), 486–495 (2016)
8. Szelągowski, M., Berniak-Woźny, J.: The knowledge and process continuum. Knowl. Process. Manag. **26**(4), 308–320 (2019)
9. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. J. Data Semant. **4**, 29–57 (2015)
10. Kemsley, S.: The changing nature of work: from structured to unstructured, from controlled to social. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, p. 2. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23059-2_2
11. Karande, A.M., Kalbande, D.: Business process analyzed factors affecting business model innovation. In: 2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE). IEEE (2015)
12. Weber, M.: The Theory of Social and Economic Organization. Simon and Schuster (2009)
13. Bandara, W., et al.: A call for ‘holistic’ business process management. In: Proceedings of the International Workshop on BPM Problems to Solve Before We Die (PROBLEMS 2021). Sun SITE Central Europe (CEUR) (2021)
14. Pöppelbuß, J., et al.: Service innovation capability: proposing a new framework. In: 2011 Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE (2011)
15. Roeser, T., Kern, E.-M.: Surveys in business process management–a literature review. Bus. Process. Manag. J. **21**(3), 692–718 (2015)
16. Boissier, F., Rychkova, I., Grand, B.L.: Challenges in knowledge intensive process management. In: 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW), pp. 65–74. IEEE Computer Society (2019)
17. Damanpour, F.: Organizational innovation: a meta-analysis of effects of determinants and moderators. Acad. Manag. J. **34**(3), 555–590 (1991)
18. Davenport, T.H.: Process Innovation: Reengineering Work Through Information Technology. Harvard Business Review Press (1993)
19. Camisón, C., Villar-López, A.: Organizational innovation as an enabler of technological innovation capabilities and firm performance. J. Bus. Res. **67**(1), 2891–2902 (2014)
20. Frishammar, J., et al.: Antecedents and consequences of firms’ process innovation capability: a literature review and a conceptual framework. IEEE Trans. Eng. Manage. **59**(4), 519–529 (2012)
21. Damanpour, F.: Organizational Innovation: Theory, Research, and Direction (2020)
22. Trkman, P., et al.: From business process management to customer process management. Bus. Process Manage. J. (2015)
23. Barney, J.: Firm resources and sustained competitive advantage. J. Manag. **17**(1), 99–120 (1991)
24. Bowman, C., Ambrosini, V.: How the resource-based view and the dynamic capability views of the firm inform corporate-level strategy. Br. J. Manage. **14**, 289–303 (2003)

25. Teece, D.J.: Business models and dynamic capabilities. Long Range Plan. **51**(1), 40–49 (2018)
26. Peteraf, M.A.: The cornerstones of competitive advantage: a resource-based view. Strateg. Manag. J. **14**(3), 179–191 (1993)
27. Paulk, M.C., et al.: Capability maturity model for software, version 1.1. Technical report. Carnegie Mellon University/Software Engineering Institute), CMU/SEI-93-TR (1993)
28. OMG, Business process model and notation (BPMN) version 2.0. OMG Standard (2011)
29. de Bruin, T., Rosemann, M.: Using the Delphi technique to identify BPM capability areas. In: ACIS 2007 Proceedings - 18th Australasian Conference on Information Systems (2007)
30. McCormack, K.P., Johnson, W.C.: Business Process Orientation: Gaining the e-Business Competitive Advantage. CRC Press (2001)
31. Looy, A.V., Backer, M.D., Poels, G.: A conceptual framework and classification of capability areas for business process maturity. Enterp. Inf. Syst. **8**(2), 188–224 (2014)
32. Van Looy, A.: Capabilities for managing business processes: a measurement instrument. Bus. Process Manage. J. (2019)
33. Leavitt, H.: Applied Organizational Change in Industry. New Perspective in Organization Research. Wiley, New York (1964)
34. Moore, G.C., Benbasat, I.: Development of an instrument to measure the perceptions of adopting an information technology innovation. Inf. Syst. Res. **2**(3), 192–222 (1991)
35. Petter, S., Straub, D., Rai, A.: Specifying formative constructs in information systems research. MIS Q. 623–656 (2007)
36. Recker, J., Rosemann, M.: A measurement instrument for process modeling research: development, test and procedural model. Scand. J. Inf. Syst. **22**(2), 1 (2010)
37. Goni, J.I.C., Van Looy, A.: Process innovation capability in less-structured business processes: a systematic literature review. Bus. Process. Manag. J. **28**(3), 557–584 (2022)
38. Delbecq, A.L., Van de Ven, A.H., Gustafson, D.H.: Group techniques for program planning: a guide to nominal group and Delphi processes. In: Delbecq, A.L., Van de Ven, A.H., Gustafson, D.H. (eds.) Management Applications Series. Scott, Foresman, Glenview (1975)
39. Okoli, C., Pawlowski, S.D.: The Delphi method as a research tool: an example, design considerations and applications. Inf. Manage. **42**(1), 15–29 (2004)
40. Paré, G., et al.: A systematic assessment of rigor in information systems ranking-type Delphi studies. Inf. Manage. **50**(5), 207–217 (2013)
41. Hasson, F., Keeney, S.: Enhancing rigour in the Delphi technique research. Technol. Forecast. Soc. Chang. **78**, 1695–1704 (2011)
42. Yin, R.K., Case Study Research: Design and Methods, 5 edn. SAGE, Thousand Oaks (2014)
43. Yin, R.K.: Case Study Research: Design and Methods. SAGE Publications (1994)
44. Gerring, J.: Case Study Research: Principles and Practices. Cambridge University Press, Cambridge (2006)

2nd International Workshop on Data-Driven Business Process Optimization (BPO 2023)

Second International Workshop on Data-Driven Business Process Optimization (BPO)

Remco Dijkman[1], Arik Senderovich[2], and Willem van Jaarsveld[1]

[1] Eindhoven University of Technology, The Netherlands
`{r.m.dijkman, w.l.v.jaarsveld}@tue.nl`
[2] York University, Canada
`sariks@yorku.ca`

Preface

Business process management is a very promising paradigm for optimizing the way in which work is performed in an organization. Decision-making is key in managing processes, and supporting decisions by combining insights from data and corresponding process models is the focus of this workshop. Decisions in business processes include assigning resources to the tasks for which they are most suited, ordering the execution of tasks to best meet customer deadlines, etc. While such questions are important in administrative processes, they are even more important in processes that have a physical component in domains such as transportation, production, and healthcare. In these domains, assigning tasks to the wrong resource or performing them in the wrong order immediately leads to increased costs, dissatisfied customers, and even health risks.

Traditionally, the research area of operations research has studied techniques for modeling and solving optimization problems in much detail. At the same time, the research area of business process management has studied techniques for aggregating the data that is needed for modeling, analyzing, and in the end optimizing business processes. Combining techniques from both areas makes it possible to solve optimization problems in practice, using models that are based on real-world data, with fewer assumptions. In particular, it allows us to create clear and realistic data-driven models of the way in which customer orders pass through an organization and of the behavior and performance of resources. While this provides clear benefits in terms of more realistic models and analysis, it also brings challenges in terms of the computational complexity of the used analysis and optimization techniques.

The goal of the Data-Driven Business Process Optimization workshop was to bring together researchers from the areas of Business Process Management and Operations Research as well as other related areas, with the overall goal of developing techniques for optimizing business processes in an organization based on models that are created from real-world data.

The workshop covered both presentations on techniques for optimizing business processes and applications of such techniques to real-world problems. It received four paper submissions, out of which three were accepted for presentation. In addition, a keynote speaker was invited to deliver a presentation.

Marlon Dumas delivered a keynote ‘On the Central Role of Simulation in Business Process Optimization’. Hans Weytjens, Jochen De Weerdt, and Wouter Verbeke presented the paper ‘Timed Process Interventions: Causal Inference vs. Reinforcement Learning’. Zeping Wang, Rehan Syed, and Chun Ouyang presented the paper ‘Towards Data-driven Business Process Redesign through the Lens of Process Mining Case Studies’. Ignacio Velásquez and Marcos Sepúlveda presented the paper ‘Analyzing the Devil’s Quadrangle of Process Instances through Process Mining’.

We hope that the reader will find the selection of papers useful to get an insight into how operations research and business process management can be combined to solve business process optimization problems.

Workshop Organization

Organizing Committee

Arik Senderovich	Rotman School of Management, University of Toronto, Canada
Remco Dijkman	Eindhoven University of Technology, The Netherlands
Willem van Jaarsveld	Eindhoven University of Technology, The Netherlands

Program Committee

Akhil Kumar	Penn State University, USA
Anna Kalenkova	University of Adelaide, Australia
Avigdor Gal	Technion, Israel
Chiara Di Francescomarino	University of Trento, Italy
Chiara Ghidini	FBK, Italy
Cristina Cabanillas	Universidad de Seville, Spain
Fernanda Gonzalez-Lopez	Pontificia Universidad Católica de Chile, Chile
Henrik Leopold	Kühne Logistics University, HPI, Germany
Marcos Sepúlveda	Pontificia Universidad Católica de Chile, Chile
Martin Matzner	Friedrich Alexander Universität, Germany
Massimiliano de Leoni	University of Padua, Italy
Matthias Weidlich	Humboldt University, Germany
Orlenys López Pintado	University of Tartu, Estonia

Timed Process Interventions: Causal Inference vs. Reinforcement Learning

Hans Weytjens(✉), Wouter Verbeke, and Jochen De Weerdt

Research Centre for Information Systems Engineering (LIRIS), Faculty of Economics and Business, KU Leuven, Leuven, Belgium
{hans.weytjens,wouter.verbeke,jochen.deweerdt}@kuleuven.be

Abstract. The shift from the understanding and prediction of processes to their optimization offers great benefits to businesses and other organizations. Precisely timed process interventions are the cornerstones of effective optimization. Prescriptive process monitoring (PresPM) is the sub-field of process mining that concentrates on process optimization. The emerging PresPM literature identifies state-of-the-art methods, causal inference (CI) and reinforcement learning (RL), without presenting a quantitative comparison. Most experiments are carried out using historical data, causing problems with the accuracy of the methods' evaluations and preempting online RL. Our contribution consists of experiments on timed process interventions with synthetic data that renders genuine online RL and the comparison to CI possible, and allows for an accurate evaluation of the results. Our experiments reveal that RL's policies outperform those from CI and are more robust at the same time. Indeed, the RL policies approach perfect policies. Unlike CI, the unaltered online RL approach can be applied to other, more generic PresPM problems such as next best activity recommendations. Nonetheless, CI has its merits in settings where online learning is not an option.

Keywords: Prescriptive Process Monitoring · Process Optimization · Timed Interventions · Causal Inference · Reinforcement Learning

1 Introduction

Moving from predicting the outcome of a running process to optimizing it with respect to a goal implies making decisions about actions that will change its course. *Prescriptive Process Monitoring* (PresPM) [3,9,14,19] is a young subfield of Process Mining (PM) studying such business process optimization methods. Optimization in the PresPM context concerns decisions an agent has to take to optimize the outcome of a running case given certain goals (metrics). It does not concern enhancing the underlying process itself, as practiced in PM. Our PresPM (see below) literature review reveals that two methods, *reinforcement learning* (RL) [16] and *causal inference* (CI) [8,12], emerge as pathways. However, a quantitative comparison is currently missing. Most of the research on

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 245–258, 2024.
https://doi.org/10.1007/978-3-031-50974-2_19

PresPM works with offline historical data creating two limitations: Online RL is not possible and experimental results prove difficult to quantify accurately for lack of counterfactuals.

This research gap defines our contribution. In our experiments, we introduce online RL to business processes and benchmark it against CI. Our use of synthetic data, rather than historical event logs as in earlier PresPM research, is not only instrumental in permitting both online RL and CI, but also enables deeper insights, a correct evaluation of the experiments' results, and the calculation of perfect policies as an absolute benchmark.

Solutions to *timed process interventions*, such as the escalation of a customer complaint process to higher management echelons, a customer call to speed up an administrative process or to maximize turnover, an additional test to conduct to reduce a patient's length of stay at hospitals, etc., can be seen as a gateway to solving the more generic problem of recommending the next best activities in a process. In timed interventions, the agent has one chance to make an intervention sometime during the process, whereas, in next best activity problems, the agent has to choose between all possible activities at every step in the process. The relative simplicity of timed process interventions in terms of combinatorial possibilities (state space) results in fewer data and computational requirements. It will permit easier insights into the characteristics of the used models and find faster real-world adoption. Furthermore, a vast number of relevant applications for timed process interventions exist. For these reasons, our experiments focus on timed process interventions, rather than next best activity recommendations.

This paper is structured as follows. Section 2 introduces concepts pertaining to PresPM and refers to related work around them. We then move on to the experimental Sect. 3 comparing RL to CI using two synthetic datasets. The insights gained from the literature review and experiments lead to a deeper discussion of the two PresPM methods in Sect. 4. We conclude this paper and suggest avenues for future work in Sect. 5.

2 Background and Related Work

In line with the very large majority of PresPM research, we limit our focus to the optimization of a single process in isolation (assuming process independence), even though in practice, many processes affect and even interact with each other.

2.1 Causal Inference

By default, CI [8,12] works with offline, logged data. The field can be subdivided into two components. The first concerns the detection of causal relationships: "Which *treatment(s)* have an effect on the process' outcome?". A treatment is a multi-class action. In the CI literature, treatments are often, but not always, binary (two classes: take or don't take the action). The second CI component involves estimating the effect of treatments. We concentrate on the *individual treatment effect* (ITE) [13], which is the difference between predicted outcomes of (possible) treatment(s) and non-treatment for a given sample. For example, when our model predicts that calling customer x (treatment) will increase revenue by 200€, while x is expected to reduce sales by 100€ if not called (non-treatment), then the ITE_x is 300€. Note that the ITE is an expectation, not a hard-coded causality. Usually, a threshold (e.g., 50€ in our example) is determined to arrive at a policy for selecting (non-) treatments. The main challenge is the absence of *counterfactuals* in the dataset. A counterfactual is the unobserved outcome of a case assuming another treatment than the one factually applied. In the absence of *randomized controlled trials* (RCTs), realized by a policy of random interventions, *selection bias* will occur as the data-gathering policy leads to different distributions of treatment and non-treatment samples in the datasets. Combating selection bias is an important aspect of CI (e.g., [13]). Real-world CI applications include marketing (e.g., churn reduction, discounting, ...), education, recommender systems, etc. Most of these applications, however, are cross-sectional rather than longitudinal: There are no timing issues, let alone sequential treatments as seen in processes.

In line with the overview provided by [9], we found no papers in the sparse PresPM literature published before 2020 claiming to use CI for process outcome optimization. [6] and [19] did apply a form of *indirect CI* with that aim, albeit without carrying the CI label. The indirect approach consists of first predicting the most likely (or distribution of) process continuation(s) (*suffix*) for every possible treatment given a certain ongoing case (*prefix*). In the second step, another model predicts outcomes for all these suffixes, which will then be used to choose a treatment. In the *direct* CI approach, the process outcomes for all possible treatments for a given prefix are directly predicted. Direct CI implementations can be found in [2] (without timing considerations) and [3] and [14] (including timing). With the exception of [2], none of the PresPM papers addresses selection bias. [1], in contrast, use a sequence-to-sequence recurrent neural network that automatically builds a treatment-invariant representation of the prefixes to combat the selection bias in a medical treatment problem.

The lack of counterfactuals in the test set hinders the accurate evaluation of CI methods' results: For a given prefix, the action recommended by the CI model may be absent from the cases in the test set. Researchers cope with this problem by relying on a predictive model to estimate outcomes, a distance-minimizing algorithm to find the nearest case in the training or dataset, or a generative model that produces augmented data [10].

2.2 Reinforcement Learning

RL [16] is an important class of ML algorithms learning policies that guide an agent's behavior or sequence of actions in an environment in order to maximize an expected cumulative *reward*. Early successes in computer games drew much attention to RL, which has since then expanded into industrial processes, robotics, autonomous driving, healthcare, engineering, finance, etc. RL comes in many flavors. We will discuss and use the widespread *Q-learning* variant. In processes, the most important reward is often the process outcome that becomes known at the conclusion (last event) of the case. Regardless, intermediate rewards could be easily included in RL should they occur. The cost of actions can be viewed as a negative reward. At its core, RL assumes an *online* environment that the agent can interact with. RL does not need an environment (→ process) model. Instead, real *episodes* (→ cases) are executed and their rewards (→ outcomes) are observed. For every encountered *state* (→ prefix), a *state-action value* (Q) is learned for every possible *action*. Q represents the state-action value for the next state (→ prefix) plus the reward minus the cost of that action to get to that next state (a *transition*). For any given prefix, the state-action values can be interpreted similarly as the effects of the possible treatments learned by CI. The difference between the state-action value for a treatment and the one for the non-treatment corresponds to the ITE at that state (→ prefix). The state-action value of the last prefix of a (completed) process is its final outcome. Given the size of the state space (→ number of possible prefixes) in most processes, these state-action values cannot be stored in tabular form (*Q-table*). Instead, they are approximated by an artificial neural network (NN). This is called *deep reinforcement learning*. At every state (→ prefix), the policy will be to choose the action with the highest relative state-action value. Learning is achieved by playing out many processes and iteratively updating the Q-table NN after each (batch of) observed rewards (→ outcomes). In order to explore all areas of the state space and to prevent prematurely settling into a sub-optimal policy, a certain degree of exploration is introduced: The agent will sometimes overrule the policy and choose another action, especially at the beginning of the learning process. RL has found many applications in process outcome optimization, but few researchers [4,5] apply RL to PresPM process optimization.

In practice, the online, real-life form of data gathering is often too slow and too expensive. It can even be dangerous at the early stages of learning when the NN is insufficiently trained and significant exploration happens. An entire spectrum of alternative data-gathering methods at different proximities to reality exists. For instance, [11] work with synthetic data for industrial process control. [17] use simulation to train robots and investigate pathways to close the *reality gap*, the mismatch between the reality and the simulation. *Offline* RL, finally, allows working with existing datasets (supervised data). PM discovery techniques, for example, yield grid graphs of business processes as representations of the RL agent's environment in [4] and [5].

2.3 Research Gap

There exists no quantitative comparative analysis of CI and RL process outcome optimization problems. This is the main research gap we address in this paper. As explained in Subsect. 2.1, the use of historical data for the test sets hampers the evaluation of CI methods for lack of counterfactuals. A similar issue appears in the RL literature that exhibits a prevalence of non-real-life work. Here, simulations or models based on reality are used to train and test online models without considering the performance on the original problems, thus ignoring the reality gap (exception: [15]). We also address this issue by making use of entirely artificial synthetic data in our experiments. This form of data allows us to accurately evaluate CI, to test online RL and eliminate the reality gap, and to share the same test set between both methods. Additionally, none of the aforementioned papers compared their results to *perfect policy* results needed to gain an intuition for the absolute performance of their methods. A perfect policy leads to better expected process outcomes (as defined by the chosen metric) than any other policy. The majority of the discussed papers treat rather complex problems for which computing such a perfect policy is intractable, hence the need for techniques such as CI and RL. We opt for relatively simple timed interventions, so that we can easily compute results for a perfect policy. The next Sect. 3 describes our contribution: making an accurately evaluated CI-RL-perfect-solution comparison based on synthetic datasets.

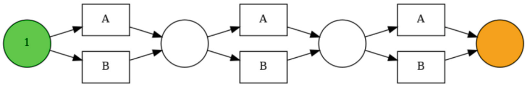
3 Experimental Comparison of CI and RL

In the following three subsections, we describe our data generation, experimental setup, and results.

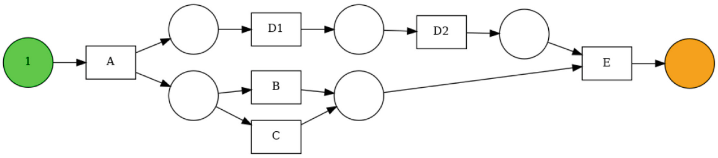
3.1 Data Generation

We work with two synthetic processes generating both the offline dataset for CI and the online environment for online RL. We first describe these two processes and then motivate our choice.

Two Synthetic Processes. The process models as Petri nets and key features of our two synthetic processes are shown in Fig. 1 and Table 1 respectively. `Process_1` is a sequence of three activities, either "A" or "B" with an according integer attribute, representing an arbitrary event attribute (e.g., age, amount, ...). At one of the three events, a (free) intervention can be made. The outcome of the process is the sum of the attributes, where the attribute of the event where the intervention took place is multiplied by 2 if activity "A" occurred at



(a) Process model representing `Process_1`



(b) Process model representing `Process_2`

Fig. 1. Synthetic generative processes: Petri net visualization

Table 1. Synthetic generative processes: key features

	`Process_1`	`Process_2`
Structure	sequence of OR	AND with OR in lower branch
Activities	every event activity from $[A, B]$ with prob $[.25, .75]$	as per graph. "B" and "C" with prob $[.2, .8]$
Attributes (Att)	uniformly random from $[0, 1, 2, 5]$	uniformly random from $[1, 2, 3]$ for "$D1$" and from $[1, 2, 3, 4]$ for "$D2$", else 0
Case variables (CV)	–	uniformly random integer from [1–10]
State space size	512	720
Interventions	at all steps or not at all	at all steps or not at all
Intervention cost	0	5
Intervention effect (intervention at event n)	Multiply Att_n by 2 in case an "A" occurred in the process, otherwise by -2.	Att corresponding to next of "$D1$" or "$D2$" multiplied by 2 (in case of "B"-process) or -4 (in case of "C"-process)
Outcome	$\sum Att$	$CV * \sum Att$

least once in the process, otherwise by -2. `Process_2` consists of five events and includes both an AND and an XOR construct. Every `Process_2` case carries an integer case attribute known from the start. Event attributes are integers as well, and an intervention can be made once in a process at any event. When an intervention is made (at a cost of 5), the attribute corresponding to the first of "$D1$" or "$D2$" to occur thereafter (if any) will be multiplied by 2 in case the process passes through its "B" branch, by -4 otherwise. The final outcome is the sum of the attributes times the case attribute.

Motivation. These processes and interventions were designed to be simple for clear insights, yet representative of real-world processes by incorporating their main challenges. Since PresPM concerns actionable decisions, we can reduce sub-processes that do not contain any decision points (and are not a branch of a parallel structure with another branch containing such a decision point) into one event, thus significantly shrinking the process model. In our experiments, the interventions only change event attributes but in reality, they may alter the control-flow as well. That would not change the CI and RL algorithms. Moreover, when all control-flow variations starting from a given decision point merge together in one location/activity later in the process model without containing any further intermediate decision points, then they can be reduced to one event as well. The value of this event's attribute will vary according to which decision was made and which control-flow variant was followed earlier.

Both processes have a strong stochastic component to reflect the uncertainty accompanying real-life processes. The values of the three activities and attributes in `Process_1` are sampled from probability distributions, whereas activities in `Process_2` are governed by the given structure, with the attributes and case variables sampled from probability distributions as well. A real-life decisionmaker is not only confronted by stochasticity, but the information available to make decisions may also differ between cases. Our synthetic processes also incorporate this aspect: as long as no "A" appears in `Process_1` or `Process_2` hasn't passed through its "B" or "C" branch, it cannot be known for sure whether intervening will be beneficial or detrimental. As in many real-world processes, the outcomes of both processes will only be known at their conclusion. Including intermediary rewards or penalties, however, would not significantly alter the CI or RL algorithms.

Our experiments investigate binary actions (interventions). This simplification allows for clearer insights without loss of generalization. As direct CI is generally not suited for sequences of actions, we further simplified by opting for one-off actions (timed interventions) to permit a CI-RL comparison; the RL method, however, can be extended to sequential or continuous actions without modification. In combination with the use of synthetic data, the resulting small state space also renders calculating perfect policies practical.

3.2 Experimental Setup

Since the indirect CI approach inevitably compounds the errors of two successive prediction models, we opted for the simplicity of working with one model in the direct approach. We use one NN to predict process outcomes for CI; the intervention (Boolean) is part of its inputs, and the batch size is 1,024. RL is achieved with a standard Q-learning architecture with a first-in-first-out 1,024-transition samples memory for stabilization. A penalty of 100 is applied for intervening more than once. An NN predicts Q for both possible actions ("intervention" and "non-intervention") at every encountered state (prefix). For a balanced comparison, the same NN architecture is used for both CI and RL. Our NNs have two long short-term memory (LSTM) and two dense layers as displayed in Fig. 2.

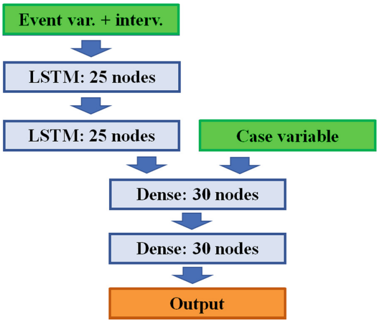


Fig. 2. The NN architectures (regression model) are almost identical for CI and RL. The first layer includes an additional "intervention" feature for CI. The last layer outputs a scalar (outcome) for CI and a 2-dimensional vector (Q for "intervention" and "non-intervention") for RL.

For the CI learning phase, an RCT dataset of 10,000 samples is generated. This largely exceeds both processes' state space size and should, therefore, offset

Table 2. Experimental settings.

Setting	Value
Input features	"activity", "attribute", and "case variable" (for `Process_2`)
Categorical features representation	one-hot-encoding
Sequence length (padding for shorter prefixes)	3 and 5 (`Process_1` and `Process_2`)
Loss function	MAE
Optimizer	ADAM
Metric	Uplift
Nr. runs per experiment	5
Batch size	1,024
Memory size (RL)	1,024
Nr. epochs	early stopping
Patience	5
Test set size	1,000 samples
Validation set size (% of training set)	20% (threshold calculation in CI)

CI's offline handicap. For RL, data are generated on the fly. The test set consists of 1,000 samples for which all counterfactuals are computed. The data generated by the synthetic processes are preprocessed as follows: The activity levels are one-hot encoded. The outcomes, attributes, and case variables (`Process_2`) are standardized. For CI, the intervention decision (1 or 0) is concatenated to the other event features. For every case sample, we build a sequence (sequence length = total process length) for every prefix, using padding to complete the sequence for ongoing process instances. We thus arrive at a two-dimensional data structure that is fed into the models' input layer. For `Process_2`, the case variable enters the models separately after the LSTM layers.

Every experiment is carried out five times and learning stopped using an early-stopping algorithm for both methods. A policy based on CI requires identifying the threshold, which we identify as the value that maximizes the ITE score on a 20% validation set. *Uplift* [7] is the metric to evaluate the results. It is the difference between the process outcomes of implementing the policy and not intervening at all, cumulated over the complete test set. The experimental settings are summarized in Table 2.

3.3 Results

We summarized our experimental results in Table 3. RL clearly outperforms CI for both processes: The mean scores are significantly higher. The standard deviations of the RL scores are much lower, making RL by far the more robust method. Most or all of this outperformance can be attributed to RL's innate superior ability to find the optimal policy (see Sect. 4). The fact that online RL permits exploring all parts of the state space plays virtually no role here, as the CI training sets in our experiments contain the complete state space as well. Were this not the case, the observed CI-RL divergence would certainly widen.

Table 3. Experimental results comparing CI, online RL, a perfect and random policy for both processes. Online RL reaches the highest uplift but requires much more computational effort than CI.

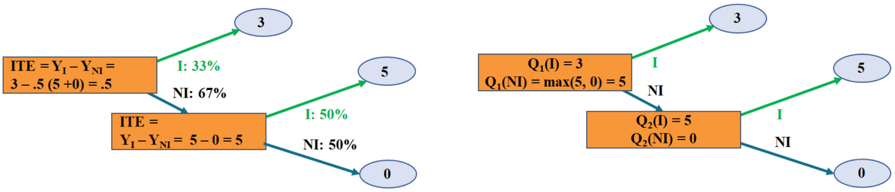
		Uplift		Computational effort		
		Mean	StDev.	Unit	Mean	StDev.
`Process_1`	CI	1,526	36.8	epochs	188	27.6
	RL	1,616	5.0	transitions	6,000	2,549
	perfect	*1,651*				
	RCT	*−515*				
`Process_2`	CI	1,682	203.1	epochs	212	51.7
	RL	1,806	34.7	transitions	26,800	10,628
	perfect	*1,845*				
	RCT	*−50,336*				

The precise knowledge of the (stochastic) synthetic generative processes enables computing perfect policies. Table 3 shows that RL comes to within 3% of the perfect policy results for both processes (some stochasticity is normal). The CI policy constitutes a substantial improvement over the RCT data-gathering policy that originally created the dataset as well, albeit to a lesser extent than RL. Having set both RL's memory and CI's batch size to 1,024, one optimization step of the NN involves the same number of samples for both methods. Since every RL transition (except for the first 1,023 ones) was followed by an NN optimization step, we can directly compare the number of RL transitions to the number of CI epochs. Table 3 shows that RL's computational requirements are one to two orders of magnitude higher than those for CI.

4 Discussion

In this section, we discuss the suitability of CI and RL for PresPM and show why RL outperformed CI in our experiments. We also address the issues of RL's online requirement, reward specification, and inefficiency.

Causal Inference. Learning counterfactuals and treatment effects is at the core of CI. The sequential aspect of processes, however, poses a problem: The decision to not treat at a certain time in a running process does not preclude treatments



(a) CI: policy requires threshold (TH), e.g., TH=0 → intervene at 1st event; TH=2 → intervene at 2nd event.

(b) RL: optimal policy follows maximum Q values: do not intervene at 1st event, intervene at 2nd event.

Fig. 3. Simple process to compare CI to RL. Both agree on the policy at the second event. At the first event, CI correctly estimates the outcome for intervention (Y_I), whereas the prediction model for the outcome for non-intervention (Y_{NI}), will observe two different outcomes (5 and 0) and **summarize** (here: average) those to 2.5. This value depends on the loss function and the samples' distribution (percentages in the graph) in the training set, which itself depends on the data-gathering policy. In contrast, RL selects the **maximum** of the two Q values in the second event.

later on in the process. For any given prefix in our experiments, direct CI relied on a predictive model to estimate the process outcomes for both intervention and non-intervention. This is problematic in the latter case: The predictive model cannot discern the optimal path from that prefix, and will instead consider the outcomes for all encountered treatments under the data-gathering policy that produced the relevant samples in the training set, as illustrated in the simplified example in Fig. 3a.

Direct CI, therefore, only operates safely on problems with one predetermined decision point and will become increasingly suboptimal when moving to processes with one flexible or with several decision points. Thresholds are sub-optimal compromises and products of optimization algorithms themselves. Dependencies between processes, e.g., when resources (space, manpower) are limited or processes interact with each other, cannot be incorporated in the CI framework. Because of these deficits, optimal policies are theoretically out of CI's reach, as confirmed in our experiments. Nevertheless, CI policy results are still better than those that the data-gathering policy yields.

Similar to all other predictive models used for prescriptive or decision-making purposes, feedback loops risk deteriorating results: Implementing the CI policy will progressively shift the real-life data distribution away from the original training data, decaying the models' predictive accuracy. Frequent updates of the CI models would help but at the same time introduce new bias in the data (new data-gathering policy). However, with a sufficient degree of randomness in the decisions taken (as in RCTs and similar to exploration in RL), this iterative, in the limit *online* CI, approach would neutralize the feedback loops.

Reinforcement Learning. RL has many theoretical advantages over CI. It does not require a process outcome prediction model and can rely on observed outcomes. RL is entirely generic: Theoretically, the same RL algorithm can deal with anything from timed intervention to next best activity prediction, which represents the ultimate complexity. RL models are very flexible: Constraints, rewards, and penalties can be added at liberty to avoid detrimental or unacceptable actions, pursue secondary goals, etc. With online RL, agents can freely interact with their environment, and dependencies between processes can be taken into account if the processes are treated concurrently by one model. Exploration in online RL theoretically visits the complete state-space (all possible prefixes). Given sufficient exploration, online RL policies will automatically adapt to a changing environment (concept drift). Proven theorems even show that online Q-learning algorithms converge given enough time. Both online and offline RL, however, are known to be inefficient, requiring many transitions to converge to the optimal policy, as demonstrated by our experiments.

The max operator over the Q-values (see Fig. 3b) explains RL outperformance versus direct CI with equal data access. For every prefix, the learned Q values represent the expected outcomes for intervention and non-intervention respectively, assuming a (calculated) *perfect* policy after that, whereas the ITEs in CI represent the difference between the expected outcomes, each of which depends on the sample distribution from the *data-gathering* policy and the loss function. Note, however, that with an online CI approach (with real-time updating after every finished process observed) and allowing exploration, this data-gathering policy would converge to the optimal policy as well, thus practically obliterating the differences between CI and RL.

Real-World Implementation. Despite its power and versatility, RL suffers from some important drawbacks. Yet, many of these are not entirely unique to RL but apply to CI and PresPM in general. The first such drawback is the risk of committing errors during real-time implementation. This implementation risk, however, can be reduced to that of the data-gathering policy (the de facto policy in place upon which the CI dataset is based) by inserting constraints into the RL algorithm that can easily deal with those. Rules mined earlier with a process discovery algorithm can frame the agent's actions. Even human intuition can be inserted by allowing the human agent to overrule the RL algorithm's proposed action. In other words, implementing RL should not be riskier either than the original, existing policy or than implementing CI. The latter two policies occasionally make or propose costly mistakes too. If necessary, a two-stage offline-online approach can further reduce the risk: Offline RL based on simulations or predictive models can serve as an initialization to an online RL that then continues to learn acting in the real world, thereby closing the reality gap.

A similar argument can be made for the related challenge of reward specification. The desired outcome for a process to be optimized will not always be one-dimensional: The primary goal may be to reduce throughput time, however, without compromising employees' well-being and product quality. Moreover, such goals may shift over time or may need adjustment in the face of concept drift. Again, this challenge is not unique to RL, and exists regardless of the solution method, if any. When possible, these goals will be consolidated into one metric for use by both CI and RL. If not, RL can be extended to include constraints on undesired actions and/or rewards/penalties that promote secondary goals. As before, the human agent can also overrule the RL's algorithms suggestions.

RL is inefficient: It is data-hungry and slow to converge. Our experiments were based on relatively short and simple processes. Longer and more complicated processes (great action width/depth) will have an exponentially larger state space, suggesting that RL will no longer be a viable option where CI could still be. Yet, in deep RL, the Q-table is replaced by an NN, which to some extent obsoletes the need to visit the complete state space as unseen state-action (prefix-action) pairs can be interpolated. Working examples of this are video games with very large, and autonomous driving with near-infinite state spaces. The more similar regions the state space contains, the better this will work. Additionally, limiting the number of actions to the most relevant ones with causal discovery techniques (first CI component in [12]) may be a worthwhile investment before starting with RL (and CI as well).

5 Conclusions and Future Work

We conducted experiments on timed process interventions with synthetic data that render genuine online RL and the comparison to CI possible and allow for an accurate evaluation of the results. We showed how the theoretical problems burdening CI can be overcome by online RL, contingent upon the strong assumption of real-time implementation of the learned policies in the real world. In our experiments, online RL produced better and more robust policies than CI. In fact, RL nearly reached the theoretically optimal solution, which can be inferred because of the use of synthetic data. The RL methods we used for timed interventions can also be applied without any modification to next best activity prediction in the limit, or problems of intermediate complexity. When computational effort and/or the real-time implementation requirement preclude online RL, CI may be a viable alternative in scenarios where the dataset covers a large and evenly distributed share of the state space and action depth is limited.

With this work, we contributed to the nascent field of PresPM. We chose a simplified setting to gain some important insights. Reaching PresPM maturity will depend on exploring other, perhaps more sophisticated approaches, in ever more realistic settings. Further extensions of this work are, therefore, plentiful. First, an initial investigation of the merits of loss attenuation [20], uncertainty [20], and future individual intervention effects [14] revealed promising insights but should be corroborated. Future work could also shed light on the conditions under which RL remains efficient enough on realistic problems with sequences of multiple possible actions (greater action width and depth). Further complications could include the introduction of outcome noise, uncertain inputs, and concept drift. Since the rewards of processes often only happen (or become known) at their conclusion, Monte-Carlo methods (as in [4]) could be a faster alternative to the classical Q-learning we used. RL does adapt to concept drift, but only very slowly. As a consequence, RL is not suited to deal with disruptions (e.g., caused by a pandemic). Digital twins for processes or organizations have been proposed as a solution [18] and are an avenue for future research. Instead of including the complete state space in the data for CI, as we did, it could be investigated to what extent CI would fall further behind online RL when the dataset only covers part of the state space (and contains selection bias caused by the data-gathering policy). For applications where online RL is not an option, more research on offline RL is recommended. Lifting the assumption of process independence would move the problem setting even closer to reality and would pose additional challenges: Process independence is a necessary assumption in CI [8]. The combinatorial explosion caused by interdependent processes is challenging for RL as well. In the domain of CI, adaptations to the standard algorithms could lead to more capabilities in terms of sequences of actions (possibly with a discounting mechanism as used in RL). Indirect CI's theoretical ability to handle sequences of actions could be weighed against the accuracy loss due to the compounding of two predictive models. Combating selection bias in processes (as in [1] for an environment without exogenous actors) beckons more research as well. We did not elaborate on how the decision points and the set of possible actions available to the agents at those points are determined.

References

1. Bica, I., Alaa, A.M., Jordon, J., van der Schaar, M.: Estimating counterfactual treatment outcomes over time through adversarially balanced representations. CoRR abs/2002.04083 (2020). https://arxiv.org/abs/2002.04083
2. Bozorgi, Z.D., Irene Teinemaa, I., Dumas, M., La Rosa, M., Polyvyanyy, A.: Prescriptive process monitoring based on causal effect estimation. Inf. Syst. **116**, 102198 (2023). https://doi.org/10.1016/j.is.2023.102198
3. Bozorgi, Z.D., Teinemaa, I., Dumas, M., Rosa, M.L., Polyvyanyy, A.: Prescriptive process monitoring for cost-aware cycle time reduction. In: 2021 3rd International Conference on Process Mining (ICPM), pp. 96–103 (2021). https://doi.org/10.1109/ICPM53251.2021.9576853
4. Branchi, S., Di Francescomarino, C., Ghidini, C., Massimo, D., Ricci, F., Ronzan, M.: Learning to act: a reinforcement approach to learn to best activities. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds.) BPM 2022. LNBIP, vol. 458, pp. 137–154. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16171-1_9
5. Bugaenko, A.A.: Application of reinforcement learning to optimize business processes in the bank. Turk. J. Comput. Math. **12**, 1638–1644 (2021)
6. de Leoni, M.d., Dees, M., Reulink, L.: Design and evaluation of a process-aware recommender system based on prescriptive analytics. In: 2020 2nd International Conference on Process Mining (ICPM), pp. 9–16 (2020). https://doi.org/10.1109/ICPM49681.2020.00013
7. Devriendt, F., Berrevoets, J., Verbeke, W.: Why you should stop predicting customer churn and start using uplift models. Inf. Sci. **548**, 497–515 (2021). https://doi.org/10.1016/j.ins.2019.12.075

8. Imbens, G.W., Rubin, D.B.: Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction. Cambridge University Press, Cambridge (2015). https://doi.org/10.1017/CBO9781139025751
9. Kubrak, K., Milani, F., Nolte, A., Dumas, M.: Prescriptive process monitoring: quo vadis? CoRR abs/2112.01769 (2021). https://arxiv.org/abs/2112.01769
10. Neal, B., Huang, C.W., Raghupathi, S.: RealCause: realistic causal inference benchmarking (2020). https://doi.org/10.48550/ARXIV.2011.15007
11. Paraschos, P.D., Koulinas, G.K., Koulouriotis, D.E.: Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. J. Manuf. Syst. **56**, 470–483 (2020). https://doi.org/10.1016/j.jmsy.2020.07.004
12. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press, Cambridge (2000)
13. Shalit, U., Johansson, F.D., Sontag, D.: Estimating individual treatment effect: generalization bounds and algorithms. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3076–3085. PMLR (2017). https://proceedings.mlr.press/v70/shalit17a.html
14. Shoush, M., Dumas, M.: When to intervene? Prescriptive process monitoring under uncertainty and resource constraints. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds.) BPM 2022. LNBIP, vol. 458, pp. 207–223. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16171-1_13
15. Spielberg, S.P., Tulsyan, A., Lawrence, N.P., Loewen, P.D., Gopaluni, R.B.: Deep reinforcement learning for process control: a primer for beginners. ArXiv abs/2004.05490 (2020)
16. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2018)
17. Tan, J., et al.: Sim-to-real: learning agile locomotion for quadruped robots. arXiv preprint arXiv:1804.10332 (2018)
18. van der Aalst, W.M., Hinz, O., Weinhardt, C.: Resilient digital twins. Bus. Inf. Syst. Eng. **63**, 615–619 (2021). https://doi.org/10.1007/s12599-021-00721-z
19. Weinzierl, S., Dunzer, S., Zilker, S., Matzner, M.: Prescriptive business process monitoring for recommending next best actions. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNBIP, vol. 392, pp. 193–209. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58638-6_12
20. Weytjens, H., De Weerdt, J.: Learning uncertainty with artificial neural networks for predictive process monitoring. Appl. Soft Comput. **125**, 109134 (2022). https://doi.org/10.1016/j.asoc.2022.109134

Towards Data-Driven Business Process Redesign Through the Lens of Process Mining Case Studies

Zeping Wang[1,2](✉), Rehan Syed[1], and Chun Ouyang[1,2]

[1] School of Information Systems, Queensland University of Technology,
Brisbane, Australia
`zeping.wang@hdr.qut.edu.au`
[2] Centre for Data Science, Queensland University of Technology, Brisbane, Australia

Abstract. Process mining is widely used for business process analysis, but rarely informs Business Process Redesign (BPR) activities. We review process mining literature and BPR framework to create thematic maps of state-of-the-art process mining analyses, techniques, outcomes and BPR best practices. We collect 156 case studies where process mining is applied and use them to validate the proposed themes. We reveal connections between the themes to explore the synergy between process mining and process redesign. Our work contributes to the development of an approach for BPR practitioners to systematically leverage the process mining capabilities, providing a solid starting point for data-driven BPR.

Keywords: Business Process Redesign · Business Process Redesign Best Practice · Process Mining · Case Study

1 Introduction

Business Process Redesign (BPR) is considered the most value-adding phase in the BPM lifecycle [10]. Reijers and Liman Mansar [31] propose a BPR framework encompassing 29 process redesign best practices to guide practitioners on their business process improvement initiatives. While this framework has been widely used across different industries, it is often difficult to replicate successful BPR projects, since most process improvement recommendations are relying on the practitioners' experience and expertise [15]. Process mining has proven its powerful capability in supporting process analytics [18], whereas process mining outcomes are rarely used to inform BPR activities [3,29]. This has motivated us to explore the synergy between process mining and process redesign.

In this paper, we aim to address the open research gap about the disconnection between process mining and BPR activities, more specifically, how process mining can be used to inform process improvement recommendations. We propose a research design based on Nickerson *et al.* [28]'s methodology to guide our study from both a theoretical aspect and a practical aspect. We review process

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 259–271, 2024.
https://doi.org/10.1007/978-3-031-50974-2_20

mining literature and BPR framework to create thematic maps of state-of-the-art process mining analyses, techniques, outcomes and BPR best practices by employing deductive thematic analysis method. We collect 153 case studies where process mining is applied and use them to validate the proposed themes. We investigate relationships between these themes to explore the synergy between process mining and process redesign.

Our work contributes towards the development of an approach for BPR practitioners to systematically leverage process mining capabilities and reduce the reliance on the practitioners' experience and expertise. As such, it provides a solid starting point for data-driven BPR.

2 Related Work

We present a few highly relevant research efforts that attempt to address data-driven process redesign. Cho *et al.* [7] develop a framework to assess the impact of a BPR best practice on process performance. The framework supports an evidence-based evaluation of BPR best practices using (process execution) event logs instead of second-hand data collected by interviews or questionnaires. The work focuses on evaluating the impact of BPR best practices rather than bridging the gap between process mining capabilities and BPR activities. Gross *et al.* [18] propose a preliminary framework to match BPR best practices and process problems that can be identified by process mining. The BPR best practices and process problems are matched subject to a two-round discussion among the authors. Systematic matching criteria for connecting process problems and BPR best practices are yet to be developed. Park and van der Aalst [29] propose an action-oriented process mining framework with an aim to connect the process mining insights and process improvement actions. The framework is designed to support process monitoring, detect violations and recommend actions to resolve the violations or mitigate their effects. The informed process improvement actions are rather specific and not guided by BPR best practices. Also, potential process deficiencies that do not trigger a violation might be overlooked.

3 Research Methodology

We adopt Nickerson *et al.* [28]'s research methodology to guide our study from a theoretical aspect and a practical aspect. Figure 1 depicts our research design.

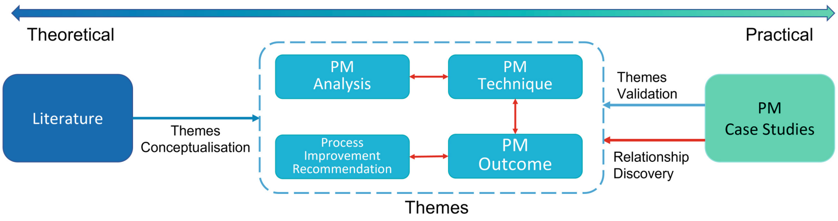


Fig. 1. A graphical overview of our research design

Conceptualise Themes. We apply thematic analysis [4] to conceptualise process mining and BPR-related topics into four themes: **process mining analysis** for activities that utilise process mining capabilities to address analytical questions [12]; **process mining technique** for techniques specifically developed for process mining analysis activities; **process mining outcome** for outputs (such as insight [3], finding [19], result [9]) produced by deploying process mining techniques; and **process improvement recommendation** for recommendations or countermeasures to improve process performance and/or compliance.

We employ purposive sampling as our literature search strategy. We search for papers that apply systematic approaches, such as systematic literature review and systematic mapping, using search string "'process mining' AND systematic". Our study excludes papers that are not domain/industry-independent. We also search for papers that discuss a process mining technique specifically. We add keywords informed by the process mining use cases from [24] to our initial search string. The BPR framework and best practices are the only process improvement methodology well-established and used across industries [27]. Thus, they are selected to inform the theme of process improvement recommendations.

Validate Themes and Discover Inter-theme Relationship. The practical aspect of our study focuses on validating the themes established from literature and analysing inter-theme relationship with real-world process mining case studies. By doing this, we are able to verify and refine the theoretical aspect with empirical evidence. It also enables us to ensure the findings are grounded in practice. In our current work, we focus on publicly available case studies to ensure truthfulness and reproducibility. We collect case studies from 3 major sources: Business Process Intelligence Challenge (BPIC)[1], the IEEE Task Force on Process Mining (TFPM)[2], and Business Process Management Cases [5,23]. By reviewing the case studies, we analyse how process mining has been used and how process improvement recommendations have been proposed when addressing real-world problems, and discover the relationship (if any) between the proposed themes.

4 Establishing Themes[3]

4.1 Process Mining Analysis

Figure 2 depicts the thematic map of process mining (PM) analysis, consisting of 12 sub-themes. **Process Discovery** builds procedural/declarative process models, or hybrid process models containing both [1,10,22,24,34]. **Process Model Enhancement** can *repair* the model to better represent the process executions [14,24], or *extend* with additional data recorded in event logs (to

---

[1] https://www.tf-pm.org/competitions-awards/bpi-challenge.

[2] https://www.tf-pm.org/resources/casestudy.

[3] Full-size thematic maps are available in a separate file on Google Drive, which can be accessed via https://tinyurl.com/bdcpw63j.

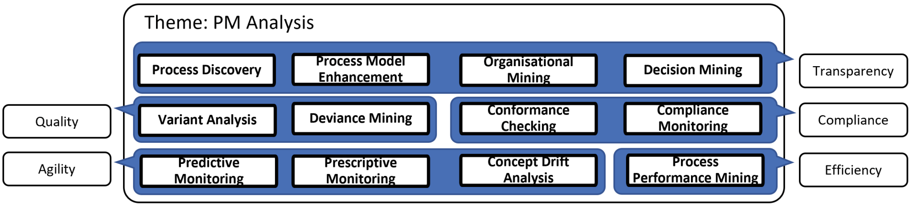


Fig. 2. Thematic map of PM Analysis

enable further analysis) [16,24,34]. **Organisational Mining** involves *Organisational Structure Mining* for discovering the resource roles and hierarchical organisational structure [16], *Social Network Mining* for discovering the performers involved in a case and their relations [1,16,22,24], and *Goal Mining* for discovering the process actor's intentions related to the execution of process activities [8,24]. **Decision Mining** (*a.k.a.* Rule Mining) examines data attributes in event logs to elicit the rules behind the choices made in the process [16,21,22,24]. **Process Performance Mining** concerns *Execution Duration* [1,10,14,16,24], *Resource Utilisation* [1,24,34], *Outcome Quality* [24], *Performance of Connected Processes* [24], and *Performance Trends Over Time* [24]. **Variant Analysis** identifies process variants in an event log [10,16,22,24,33]. **Deviance Mining** discovers the reason behind a certain variant deviating from the most frequently taken path [24]. **Conformance Checking** examines if the actual process behaviour conforms with the expected behaviour [10,11,14,24]. Similarly, **Compliance Monitoring** checks if the behaviour of active cases complies with predefined rules and constraints [24]. **Concept Drift Analysis** detects changes in the process behaviour over time [6,16,22,24,32]. **Predictive Monitoring** predicts the process outcome, risk and/or performance of active cases [16,22,24,25]. **Prescriptive Monitoring** identifies specific interventions to improve the likelihood of a favourable outcome, or when an intervention is needed [20,24].

The 12 sub-themes can be clustered into 5 groups [24]. Process Model Discovery, Process Model Enhancement, Organisational Mining and Decision Mining focus on *transparency*. Process Performance Mining focuses on *efficiency*. Variant Analysis and Deviance Mining are concerned with process *quality* in terms of how certain process traces may differ from common execution paths. Conformance Checking and Compliance Monitoring deal with process *compliance*. Predictive/prescriptive monitoring and concept drift focus on process *agility*.

4.2 Process Mining Technique

Figure 3 depicts the thematic map of process mining technique, consisting of 10 sub-themes. **Process Discovery** has two different types: *Model-based Discovery* produces a graphical process map [1,14,16,22,24], and *Constraint-based Discovery* discovers textual process descriptions, e.g., declarative mining [1,24]. **Variant Analysis** covers *Model-based Techniques* [16] and *Vector-*

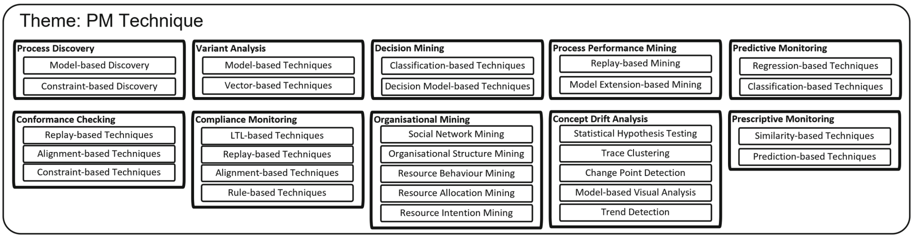

Theme: PM Technique
Process Discovery
Model-based Discovery
Constraint-based Discovery
Variant Analysis
Model-based Techniques
Vector-based Techniques
Decision Mining
Classification-based Techniques
Decision Model-based Techniques
Process Performance Mining
Replay-based Mining
Model Extension-based Mining
Predictive Monitoring
Regression-based Techniques
Classification-based Techniques
Conformance Checking
Replay-based Techniques
Alignment-based Techniques
Constraint-based Techniques
Compliance Monitoring
LTL-based Techniques
Replay-based Techniques
Alignment-based Techniques
Rule-based Techniques
Organisational Mining
Social Network Mining
Organisational Structure Mining
Resource Behaviour Mining
Resource Allocation Mining
Resource Intention Mining
Concept Drift Analysis
Statistical Hypothesis Testing
Trace Clustering
Change Point Detection
Model-based Visual Analysis
Trend Detection
Prescriptive Monitoring
Similarity-based Techniques
Prediction-based Techniques

Fig. 3. Thematic Map of PM Technique

based Techniques [16,33]. **Decision Mining** utilises *Decision-aware Techniques* [1] and *Classification-based Techniques* [21]. **Process Performance Mining** mainly covers techniques based on *Replay* to support bottleneck identification and enable temporal analysis [1], and *Model Extension-based Techniques* to incorporate performance-related information to the control-flow and enhance process analysis [30]. **Organisational Mining** includes techniques such as *Social Network Mining* to discover the relationship between resources [1], *Organisational Structure Mining* to discover the organisational hierarchical structure [1], *Resource Behaviour Mining* to discover the behaviour of resources [1,26], *Resource Allocation Mining* to discover the allocation of resources [13], and *Resource Intention Mining* (Goal Mining) to discover the intention of resources instead of the goal of the process and whether they are aligned [17]. **Conformance Checking** consists of techniques that are employed for conformance checking analysis, including *Replay-based Techniques*, *Alignment-based Techniques*, and *Constraint-based Techniques* [2]. **Compliance Monitoring** differentiates from conformance checking as it deals with active cases, though some techniques used for conformance checking can also be applied for compliance monitoring. **Predictive Monitoring** applies techniques based on *Regression* [25] or *Classification* [25]. **Prescriptive Monitoring** uses *Similarity-based Techniques* to recommend interventions based on completed cases that have yielded the same output [20], and *Prediction-based Techniques* to recommend interventions based on predicted outcome or performance for active cases [20]. **Concept Drift Analysis** encompasses six types of techniques [32]: *Statistical Hypothesis Testing* to compute the place of change and its characteristics (sudden drifts or gradual drifts), *Trace Clustering*, *Change Point Detection*, *Model-based Visual Analysis*, *Change Detection* to discover the process after change, and *Trend Detection* to identify the trend of drift.

4.3 Process Mining Outcome

Figure 4 depicts the thematic map of process mining outcomes, consisting of 8 sub-themes. **Process Model** includes both *Graphical Model* mined by traditional process discovery algorithms [1,14,16,22,24] and *Declarative Model* by declarative algorithms [1,22,24]. **Process Variants** involves detected *Variants* [16,24], *Differences & Similarities between Variants* [16,24,33], *Variant Frequency* [16,33] and *Deviant Point* [16,24]. **Process Rules** covers two types

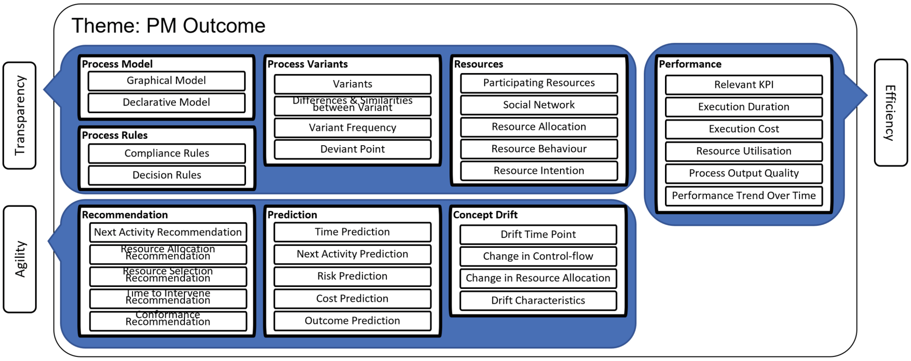


Fig. 4. The Thematic Map of PM Outcomes

of rules, which are *Compliance Rules* [16,24], and *Decision Rules* [16,21,24]. **Resources** include *Participating Resources* [16,24,34], *Social Network* [1,16, 24,34], *Resource Allocation* [16,24], *Resource Behaviour* [1,16], and *Resource Intention* [8,24]. **Prediction** is the outcome produced by predictive monitoring, including *Time Prediction* [16,22,24,25], *Next Activity Prediction* [16,24], *Risk Prediction* [24,25], *Cost Prediction* [24,25], and *Outcome Prediction* [16,24] of active cases. **Recommendation** is usually derived from the prediction, aiming for a favourable outcome or performance [14,20], including *Next Activity Recommendation* [16,20,24], *Resource Allocation Recommendation* [14,16,20], *Resource Selection Recommendation* [24], *Time to Intervene Recommendation* [24], and *Conformance Recommendation* [24]. **Concept Drift** includes *Drift Time Point* [16], *Change on Control-flow* [16,24], *Change on Resource Allocation* [16], and *Drift Characteristic* [16]. **Performance** related outcomes include *Relevant KPI* that reveal critical performance indicators for decision making [16], *Execution Duration* [1,16,24,34], *Execution Cost* [16], *Process Output Quality* [14,24], and *Performance Trend Over Time* [24].

The 10 sub-themes can be categorised into 3 groups. Process Model, Process Variants, Process Rules and Resources focus on process *transparency*. Prediction, Prescriptive Recommendation and Concept Drift focus on process *agility*. Performance focus on process *efficiency*.

4.4 Process Improvement Recommendation

The BPR framework and 29 best practices proposed in [31] are considered seminal work for process improvement, and are proven to guide the design of to-be processes with improved process performance [27]. Other papers we found about BPR are either on a too high level that only discuss methodology rather than how a process can actually be improved, or on a too specific level that is case by case. Thus, establishing the Process Improvement Recommendation theme differs from the other three themes. The existing BPR framework is transformed into a thematic map, as illustrated in Fig. 5.

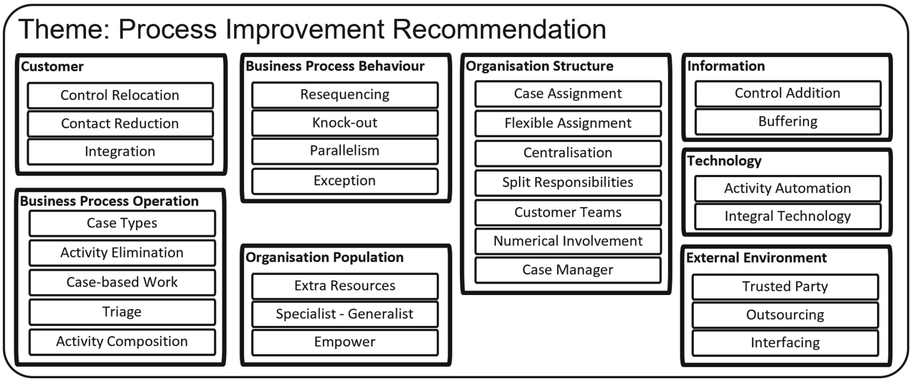


Fig. 5. Thematic Map of Process Improvement Recommendation (a representation of BPR best practices in [31])

5 Validation and Analysis

In this section, we refer to the case studies published in process mining to validate and where appropriate, further extend our findings from the literature. All case studies selected must analyse a business process using process mining techniques. Event log generation, evaluation of process discovery algorithms, and data pre-processing are not considered specific to process mining. A total of 156 case studies were selected, including 127 BPIC reports, 22 TFPM case studies, and 7 BPM case studies. The full list of selected case studies can be found on a companion file (see footnote 3). The BPIC is an annual process mining competition. Event logs are publicly available and provided to the contenders as the sole information source. Despite BPIC focuses on discovering and analysing the process, some contenders also tried to recommend process improvements, especially in the later years. The TFPM and BPM case studies represent real-world scenarios in which practitioners have access to additional information channels beyond event logs, and methods like interviews and questionnaires are also available.

In the case studies, we look for any process mining analysis, any process mining technique employed, any outcome deriving from process mining analysis, and any recommendations for process improvement. Then, we classify them according to the pertinent themes outlined in Sect. 4. If no process improvement recommendation is made based on a PM analysis, the theme is left blank.

5.1 Validation of Findings

Coverage and Awareness of Themes. We did not find any new themes from the case studies analysis. Table 1 lists the frequency of PM Analysis used in case studies. Process Discovery is the mostly used process mining analysis among the selected case studies. Almost all case studies discover a graphical process model. The declarative model has only been identified twice in (Brandão *et*

Table 1. Frequency of each PM analysis in the selected case studies

PM Analysis	Frequency	PM Analysis	Frequency	PM Analysis	Frequency
Process Model Discovery	108	Process Performance Mining	92	Compliance Monitoring	7
Process Model Enhancement	0	Variant Analysis	86	Predictive Monitoring	19
Organisational Mining	54	Deviance Mining	23	Prescriptive Monitoring	1
Decision Mining	3	Conformance Checking	13	Concept Drift Analysis	22

al., BPIC2014) and (Jalali, BPIC2016). In fact, the process analyses in many case studies were purely based on process models. They discovered the model using pre-processed event log which fits their analytical purpose to conduct their analysis for different purposes, such as (Hevia and Saint-Pierre, BPIC2013) and (Radhakrishnan and Anantha, BPIC2013). Process Model Enhancement has not been identified from the selected case studies. It enables other analyses, such as the model-extension-based performance mining techniques, rather than providing analytical value itself. Some PM Analyses and PM Techniques appear to be less frequently used in our selected case studies. This may be due to the current coverage of case studies, as the majority of the selected case studies are from BPICs, which often include specific questions to be addressed. These questions clearly influence the decision on what process mining analysis to conduct.

Few Improvement Recommendations Guided by BPR Best Practices. Through the entire validation, only 42 case studies have clearly proposed process improvement recommendations that can be used to derive the theme. Among those, only (Bautista *et al.*, BPIC2012) made recommendations that were guided by the BPR best practices. Case studies from TFPM and BPM Cases are real-world cases where the practitioners were hired to solve real business problems. However, from these reports, we do not observe any information about how and to what extent the process was improved. Consequently, we are unable to use these reports to validate the Process Improvement Recommendation theme. Based on our observations, it is evident that there is still a divergence between process mining and BPR.

5.2 Relationship Discovered

The complete set of relationships discovered from the selected case studies is available in a separate file (see footnote 5). Below, we present some findings worth of discussion.

PM Analysis and PM Technique. We observed an interesting connection between analysis and technique. Conformance checking techniques are used to identify variants (Caron et al., BPIC2011), and variant analysis techniques are used to check process conformance in (Hansen, BPIC2013). From the perspective of the techniques of variant analysis and conformance checking, they are very similar. The only difference between conformance checking and variant analysis is that conformance checking compares the variants against a reference model, but the variant analysis does not deem a variant as reference.

The work of (Paszkiewicz and Picard, BPIC2013) applied a model-based variant analysis technique to check if the cases followed the correct event order, which is a compliance rule. Thus, a connection between *Variant Analysis* techniques and *Compliance Monitoring* is identified.

BPIC 2013 requests contenders to analyse the "ping pong behaviour" in the process. The problem was interpreted differently by contenders. Some contenders, such as (Paszkiewicz and Picard, BPIC2013) and (van den Spiegel *et al.*, BPIC2013), tackle this question by importing the resource values from the event log as activity names to process mining software. The generated Directed follow graph captures the directional relationships between resources, instead of activities. This is seen as using the *Model-based Discovery* techniques to understand the problem. While others use *Social Network Mining* techniques to discover the handover of work between resources, such as (Hansen, BPIC2013).

There are also contenders who use Social Network Mining technique to discover the control-flow, such as in (Jalali, BPIC2016). The process in question captures the user behaviour of a website, which is relatively more complex in nature. The contenders set the activities as resources when importing the event log into the Social Network Miner in ProM. The chord diagram generated captures the connections between activities, which reflects how the users interact with the website. This indicates a connection between Process Model Discovery analysis and Social Network Mining technique.

PM Technique and PM Outcome. Connections between PM techniques with PM Outcomes are mostly straightforward, e.g., Model-based Discovery techniques generate process models, and Predictive Monitoring techniques generate process outcome prediction, process time prediction, risk prediction and cost prediction. One interesting finding is that some contenders apply process mining algorithms designed for discovering process control-flow to mine social networks. In (van den Spiegel *et al.*, BPIC2013) and (Teinemaa *et al.*, BPIC2015), contenders replace the activity attribute with the resource attribute when importing the log into process mining software. They claim the resulting diagram generated by process mining software captures the handover of work between resources and thus presents a social network.

PM Outcome and Process Improvement Recommendation. Suggesting process improvement recommendations is a complex task, which usually requires consideration of many factors within and beyond data recorded in event logs. In the selected case studies, practitioners made their recommendations usually based on multiple analytical findings. Table 2 lists several examples of connections between PM outcome and Process Improvement Recommendation.

Table 2. Examples of connections between PM Outcome and Process Improvement Recommendation discovered from selected case studies. Refer to footnote 5 for access to the complete list of connections discovered from selected case studies.

Source	Sub-Outcome	Sub-sub-outcome	Explanation	Recommendation made	Related BPR Best Practice
(Bautista *et al.*, BPIC2012)	Resource	Resource allocation	Resource assigned to different activities	Recast generalist as specialist	Generalist-Specialist
	Resource	Resource utilisation	Specialists may be better at handling a large number of cases		
(Bautista *et al.*, BPIC2012)	Resource	Resource allocation	Fewer resources involved in a case, the case would last too long	Early termination of cases	Knock- out
	Performance	Execution duration	Slow moving cases are more likely to be rejected		
(Berger, BPIC2017)	Variants	Variants	Contact customer more times, cases are more likely get cancelled	Reduce the contact frequency with customer	Contact reduction
(Berger, BPIC2017)	Performance	Execution duration	Manual work takes too long	Digitise the application process	Activity automation
(Dmitry *et al.*, BPIC2020)	Variants	Variants	An unnecessary step is identified in some cases	Eliminate unnecessary steps for cases of a complete cancellation	Activity elimination
(González *et al.*, BPIC2020)	Resource	Resource allocation	Only 2 resources can approve applications	Authorise other roles for approval	Empower
				Increase the number of supervisors	Extra resource
(Elena *et al.*, BPIC2020)	Variants	Variants	Rework because of wrong input	Add standard fill-in instructions	Interfacing
(Filipov *et al.*, BPIC2020)	Variants	Variants	Too many payment events, creating double payments	Only pay after trip ends	Activity composition
(Pakileva *et al.*, BPIC2020)	Performance	Execution duration	Long waiting time due to peak hour	Consider allocate resources to peak time	Flexible assignment
	Resource	Resource allocation	Non-effective resource allocation and additional assignments		

While the discovered connections between PM Analysis, PM Technique and PM Outcome are relatively clear and strong, how PM Outcome relates to Process Improvement Recommendation is relatively weak. We do identify practitioners implementing process mining for process analysis, and their process improvement recommendations are made based on their PM outcomes. However, often the PM outcomes are not the mere source to derive process improvement recommendations. Some improvement recommendations are not only based on the PM outcomes but also on the relevant context and domain knowledge as well as findings from other analysis methods. For this reason, a valid Process Improvement Recommendation should be made based on multiple inputs and the PM Outcome is one of them.

As shown in Table 2, process improvement recommendations may be proposed based on more than one PM outcome, such as (Bautista *et al.*, BPIC2012). Also, it is possible that more than one process improvement recommendation are made due to one PM outcome, such as (González *et al.*, BPIC2020). This is potentially due to the fact that not all the required information is available in the event log and as a result, the analysts do not have sufficient information to make the most suitable recommendation.

Another interesting finding is different practitioners propose different process improvement recommendations for the same scenario. In BPIC2017, the contenders are requested to answer a question about the conversion rate between the applicants with one offer and those with multiple offers. Despite the different interpretations that have been made in different reports, different recommendations are made based on similar analysis outcomes. (Povalyaeva *et al.*, BPIC2017), (Fani Sani and Sotudeh, BPIC2017) and (van der Ham, BPIC2017) all think more offers would bring a higher conversion rate, but (Povalyaeva *et al.*, BPIC2017) suggest proposing more offers to applicants, (Fani Sani and Sotudeh, BPIC2017) have not suggested any recommendation regarding this finding, and (van der Ham, BPIC2017) suggest to provide more attractive offers to applicants with a high credit score.

Connections Across All Themes. Despite the connection between PM Outcome and Process Improvement Recommendation is relatively weak, we indeed discovered connections that start from PM Analysis and end in Process Improvement Recommendation. In BPIC2015, the contenders were required to identify possible points for improvement in each municipality's organisational structure. The study of (van den Spiegel and Blevi, BPIC2015) tackled this question by analysing the relationship between resources. They found that some resources worked on the same cases, and suggested that assigning case ownership to a dedicated resource could improve the process. In such an example, the connection across all themes is clear; they conducted an *Organisational Mining* analysis (**PM Analysis**), used *Social Network Mining* technique (**PM Technique**), extracted *Social Network* (**PM Outcome**), and suggested *Case assignment* (**Process Improvement Recommendation**).

6 Conclusion

With an aim to establish the connection between process mining and BPR, we have established 4 themes — process mining analysis, process mining technique, process mining outcome, and process improvement recommendation. We have validated the proposed themes with case studies. Along the validation of the themes, we also managed to discover some connections between the themes.

The current work has a limited coverage of literature and case studies, resulting in potential threat to validity. Expanding the literature and case study coverage can mitigate this limitation. The proposed themes and discovered connections have not been tested in a real business context. We believe a field study could expose potential missing themes and dramatically increase their applicability. All these inform important agenda items for future work.

References

1. van der Aalst, W.: Process Mining: Data Science in Action. Springer (2016). https://doi.org/10.1007/978-3-662-49851-4

2. Alizadeh, M., Lu, X., Fahland, D., Zannone, N., van der Aalst, W.M.: Linking data and process perspectives for conformance analysis. Comput. Secur. **73**, 172–193 (2018)
3. Beerepoot, I., Martin, N., Koorn, J.: From insights to INTEL: evaluating process mining insights with healthcare professionals. In: HICSS (2023)
4. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. **3**, 77–101 (2006)
5. vom Brocke, J., Mendling, J.: Business Process Management Cases: Digital Innovation and Business Transformation in Practice. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-58307-5
6. Ceravolo, P., Tavares, G.M., Junior, S.B., Damiani, E.: Evaluation goals for online process mining: a concept drift perspective. IEEE Trans. Serv. Comput. **15**, 2473–2489 (2022)
7. Cho, M., Song, M., Comuzzi, M., Yoo, S.: Evaluating the effect of best practices for business process redesign: an evidence-based approach based on process mining techniques. DSS **104**, 92–103 (2017)
8. Diaz, O.E., Perez, M.G., Lascano, J.E.: Literature review about intention mining in information systems. J. Comput. Inf. Syst. **61**, 295–304 (2021)
9. Dixit, P.M., Buijs, J.C.A.M., van der Aalst, W.M.P., Hompes, B.F.A., Buurman, J.: Using domain knowledge to enhance process mining results. In: Ceravolo, P., Rinderle-Ma, S. (eds.) SIMPDA 2015. LNBIP, vol. 244, pp. 76–104. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53435-0_4
10. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, 2nd edn. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-56509-4
11. Dunzer, S., Stierle, M., Matzner, M., Baier, S.: Conformance checking: a state-of-the-art literature review, pp. 1–10. ACM (2019)
12. van Eck, M.L., Lu, X., Leemans, S.J.J., van der Aalst, W.M.P.: PM^2: a process mining project methodology. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) CAiSE 2015. LNCS, vol. 9097, pp. 297–313. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19069-3_19
13. Erasmus, J., Vanderfeesten, I., Traganos, K., Jie-A-Looi, X., Kleingeld, A., Grefen, P.: A method to enable ability-based human resource allocation in business process management systems. In: Buchmann, R.A., Karagiannis, D., Kirikova, M. (eds.) PoEM 2018. LNBIP, vol. 335, pp. 37–52. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02302-7_3
14. Erdogan, T.G., Tarhan, A.: Systematic mapping of process mining studies in healthcare. IEEE Access **6**, 24543–25567 (2018)
15. Fehrer, T., Fischer, D.A., Leemans, S.J.J., Röglinger, M., Wynn, M.T.: An assisted approach to business process redesign. DSS **156**, 113749 (2022)
16. Garcia, C.D.S., et al.: Process mining techniques and applications - a systematic mapping study. Expert Syst. Appl. **133**, 260–295 (2019)
17. Ghasemi, M.: Towards goal-oriented process mining. In: RE, pp. 484–489 (2018)
18. Gross, S., Yeshchenko, A., Djurica, D., Mendling, J.: Process mining supported process redesign: matching problems with solutions. In: PoEM, pp. 24–33 (2020)
19. Koorn, J.J., et al.: Bringing rigor to the qualitative evaluation of process mining findings: an analysis and a proposal. In: ICPM, pp. 120–127 (2021)
20. Kubrak, K., Milani, F., Nolte, A., Dumas, M.: Prescriptive process monitoring: quo vadis? PeerJ Comput. Sci. **8**, e1097 (2022)

21. Leewis, S., Smit, K., Zoet, M.: Putting decision mining into context: a literature study. In: Agrifoglio, R., Lamboglia, R., Mancini, D., Ricciardi, F. (eds.) Digital Business Transformation. LNISO, vol. 38, pp. 31–46. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-47355-6_3
22. Maita, A.R.C., et al.: A systematic mapping study of process mining. Enterp. Inf. Syst. **12**, 505–549 (2018)
23. Mendling, J., Rosemann, M., vom Brocke, J.: Business Process Management Cases. Volume 2: Digital Transformation - Strategy, Processes and Execution. Springer, Berlin (2021). https://doi.org/10.1007/978-3-662-63047-1
24. Milani, F., Lashkevich, K., Maggi, F.M., Di Francescomarino, C.: Process mining: a guide for practitioners. In: Guizzardi, R., Ralyté, J., Franch, X. (eds.) RCIS 2022. LNBIP, vol. 446, pp. 265–282. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05760-1_16
25. Márquez-Chamorro, A.E., Resinas, M., Ruiz-Cortés, A.: Predictive monitoring of business processes: a survey. IEEE Trans. Serv. Comput. **11**, 962–977 (2018)
26. Nakatumba, J., van der Aalst, W.M.P.: Analyzing resource behavior using process mining. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 69–80. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_8
27. Netjes, M., Mansar, S.L., Reijers, H.A., van der Aalst, W.M.P.: Performing business process redesign with best practices: an evolutionary approach. In: Filipe, J., Cordeiro, J., Cardoso, J. (eds.) ICEIS 2007. LNBIP, vol. 12, pp. 199–211. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88710-2_16
28. Nickerson, R.C., Varshney, U., Muntermann, J.: A method for taxonomy development and its application in information systems. Eur. J. Inf. Syst. **22**, 336–359 (2013)
29. Park, G., van der Aalst, W.M.P.: Action-oriented process mining: bridging the gap between insights and actions. Prog. Artif. Intell. 1–22 (2022)
30. Pika, A., Wynn, M.T., Fidge, C.J., ter Hofstede, A.H.M., Leyer, M., van der Aalst, W.M.P.: An extensible framework for analysing resource behaviour using event logs. In: Jarke, M., et al. (eds.) CAiSE 2014. LNCS, vol. 8484, pp. 564–579. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07881-6_38
31. Reijers, H.A., Liman Mansar, S.: Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. Omega **33**, 283–306 (2005)
32. Sato, D.M.V., Freitas, S.C.D., Barddal, J.P., Scalabrin, E.E.: A survey on concept drift in process mining. ACM Comput. Surv. Article 189 (2021)
33. Taymouri, F., Rosa, M.L., Dumas, M., Maggi, F.M.: Business process variant analysis: survey and classification. Knowl. Based Syst. **211**, 106557 (2021)
34. Zerbino, P., Stefanini, A., Aloini, D.: Process science in action: a literature review on process mining in business management. Technol. Forecast. Soc. Change **172**, 121021 (2021)

Analyzing the Devil's Quadrangle of Process Instances Through Process Mining

Ignacio Velásquez(✉) and Marcos Sepúlveda

Department of Computer Science, Pontificia Universidad Católica de Chile,
Santiago, Chile
{iavelasquez,marcos.sepulveda}@uc.cl

Abstract. The Devil's Quadrangle is a framework used in Business Process Management to describe the inherent process performance trade-offs regarding the time, costs, flexibility, and quality dimensions. In practice, improving a process through one of these dimensions might have a negative effect on the performance of the other dimensions. The dimensions considered by the Devil's Quadrangle are often used for defining indicators that illustrate the overall performance of processes. From a Process Mining perspective, analyzing these dimensions at higher granularity levels, such as for every process instance, is of interest. To achieve this, this work proposes a method for defining Process Mining filters based on metrics related to performance indicators of the four Devil's Quadrangle dimensions. The metrics are calculated for every process instance, which allows using the filters to observe differences in process behavior while considering constraints to the performance indicators and trade-offs among the four dimensions. It is expected that this visualization will be helpful during exploratory process analysis. It will facilitate the identification of process instances that conform to the filters applied to the performance indicators, as well as the dimensions where improvement is required while considering process instances that do not conform to the applied filters. A Celonis dashboard with the proposed filters has been generated to validate the method.

Keywords: Process Mining · Devil's Quadrangle · Process Analysis · Performance Analysis · Process Performance Indicators

1 Introduction

The Devil's Quadrangle (DQ) is a framework used in Business Process Management to describe the inherent process performance trade-offs regarding four main dimensions: time, costs, flexibility, and quality [7]. Ideally, process redesign involves decreasing execution time and costs, while improving quality and ability to react to variation. In practice, however, improving upon either of these

---

This work was funded by the Agencia Nacional de Investigación y Desarrollo de Chile. Grant numbers ANID FONDECYT 1230697, and ANID-Subdirección de Capital Humano/Doctorado Nacional/2021-21210022.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 272–284, 2024.
https://doi.org/10.1007/978-3-031-50974-2_21

dimensions may have a weakening effect on another [19]. The DQ is thus utilized to signify the difficult trade-offs that must be made [7,19]. The DQ framework is often considered during process redesign and optimization endeavors [8,19]. Figure 1(a) shows the trade-offs among the four DQ dimensions [19].

The dimensions considered by the DQ are often used for defining Process Performance Indicators (PPI) [8,13,17]. PPI are quantifiable metrics that allow us to evaluate the efficiency and effectiveness of business processes [10]. While PPI are usually used to measure process performance as a whole, they can also be calculated for specific process variations [12]. However, given the existence of event logs that record several process instances at higher granularity levels, it would be desirable to analyze the dimensions considered by the DQ for every process instance. This would allow identifying outlier process instances that do not align with the general process performance, thus allowing the analysis of their distinct features and execution paths.

Process Mining (PM) discipline can be utilized to achieve the above. This discipline involves discovering, monitoring, and improving processes by extracting knowledge from event logs [1]. PM can be used for discovery, allowing the construction of process models depicting the behavior observed in the event log [16]. These models can be analyzed from a performance perspective by calculating PPI for several process dimensions [1]. Process models can also be discovered after filtering outliers from the event log [20] or based on event attributes [1].

Leveraging process performance discovery and analysis, as well as the filtering capabilities of PM, this work proposes a method for the analysis and comparison of PPI between process instances through the perspective of the DQ. Specifically, by adapting the functionality of sliders commonly used by PM tools (see Fig. 1(c)) [15], the method utilizes PM filters to replicate the trade-offs depicted by the DQ. Figure 1(b) shows the conceptualization of these filters, which are generated by calculating performance metrics related to PPI of the four DQ dimensions for every process instance. Filtering based on these metrics, process models are then generated and utilized to visualize process behavior while considering constraints to the PPI and trade-offs among the four DQ dimensions.

The proposed method contributes towards understanding how the process is being performed according to expectations within the DQ dimensions. The performance metrics that are defined for every dimension can be analyzed at distinct abstraction levels: event log, process variant, and process instance. The main contribution of the method is the definition of a way for dynamically exploring the event log, and navigating through the different DQ dimensions. This facilitates the exploratory analysis of processes, allowing the identification of instances that conform to the filters applied over the PPI, as well as the dimensions where improvement is required, while considering process instances that do not conform to the applied filters.

The remainder of this work is structured as follows: Sect. 2 summarizes the background behind this work; existing research on the DQ and the definition of PPI considering the DQ dimensions are described. In Sect. 3, the conceptualization of the proposed method is provided, whereas an example utilized for

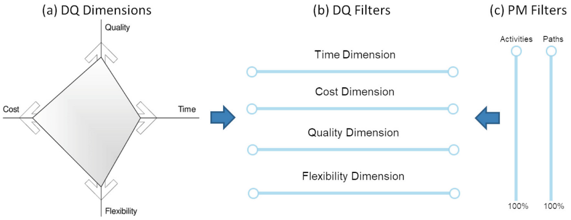


Fig. 1. Conceptualization of DQ filters based on the DQ dimensions, and PM filters.

illustrating the method is presented in Sect. 4. Section 5 focuses on the implementation of the method and its application in the illustrative example. Finally, the conclusions of this work are presented in Sect. 6.

2 Background

The DQ framework is presented in [7] as a model for signifying the difficult trade-offs that sometimes have to be made among the time, costs, flexibility and quality dimensions when applying process redesign measures. Several redesign and optimization endeavors have been performed while considering the DQ. Namely, [14] uses the DQ dimensions to analyze the impact of redesign proposals created through a simulation model for a food production process. In [3], redesign heuristics based on the DQ dimensions were utilized for the improvement of an emergency department of a public hospital. Similarly, [11] considers redesign heuristics based on the DQ dimensions within the product development process of the automotive industry, whereas [21] considers these heuristics for proposing a business process redesign capacity assessment framework.

To facilitate process redesign, DQ dimensions are often utilized for defining PPI. An approach for the calculation of metrics associated to every DQ dimension is proposed in [13]. In the context of analyzing the feasibility of service robots in the hospitality industry, [5] proposes several PPI from the DQ dimensions for assessing this feasibility. Due to the utility of using real data for PPI calculation, researchers have also considered using event logs from a PM perspective. Specifically, [8] proposes PPI for an emergency room process, whereas [17] proposes PPI for an order to cash process, both considering the DQ dimensions.

The PPI proposed by the above publications are provided at a process level, i.e., for every PPI, a single value is calculated for the whole process. [12] considers a higher granularity for the analysis of PPI; the author proposes the calculation of PPI based on the DQ dimensions for several process variants. Based on these metrics, the quadrangles depicting the performance of every variant are generated, and an interface is proposed for their side-by-side comparison. This method, however, is of a static nature, as PPI must be manually annotated in the interface. Moreover, the proposal does not allow the calculation of metrics for every process instance, which corresponds to a higher granularity level.

3 Method

The method proposed in this work addresses the observed limitations by first defining how to calculate performance metrics for every independent process instance. Second, a dynamic interface is proposed, which can be utilized for filtering process models in real-time, as well as obtaining PPI for every dimension considered by the DQ. The method facilitates process exploratory analysis, the comparison of distinct instances, and the identification of improvement opportunities based on the constraints expected by analysts.

The method proposed in this work is inspired by the sliders generally provided by PM tools for filtering process models. These sliders are used to allow dynamically filtering the number of nodes and edges that should be visualized in an attempt to reduce model complexity [15]. The objective of the proposed method aligns with the need for dynamically filtering process models. However, instead of filtering nodes or edges of the process model, the aim is to provide sliders that allow filtering process instances based on the four dimensions considered by the DQ. Every slider is double-ranged, i.e., it allows the user to define a minimum and a maximum value for filtering the process model on a particular dimension. Filtering the process model through either of these sliders would impact the other sliders, as their possible minimum and maximum values must be recalculated based on the remaining process instances. This aligns with the goal of the DQ, i.e., that affecting either process dimension may affect the others [19].

Four steps are required for the application of the proposed method: (i) to calculate time, cost, flexibility, and quality metrics for every instance in the event log, (ii) to annotate the event log with the calculated metrics, (iii) to generate the filters depicting the DQ, and (iv) to analyze the process, supported by the generated sliders. Each of these steps is described in the following subsections.

3.1 Step 1 - DQ Metric Calculation

The first step of the method is to calculate time, cost, flexibility, and quality metrics for every trace of the event log. The calculation of time and cost metrics at a process instance level has already been addressed in the literature [1,18]. However, the calculation of metrics for the flexibility and quality dimensions is generally done at the process level (a lower granularity), i.e., a single PPI is calculated for the whole event log [9]. Formulas for calculating flexibility and quality metrics at the process instance level are proposed, adapting those defined by [9] for their calculation at the process level. The four metrics presented in the current work are described below.

Time Dimension: Case Duration. The metric considered for the time dimension is case duration (lead time). Formally, lead time corresponds to the total time from the creation of the case to its completion [1]. This can be calculated for every instance by calculating the difference between the complete timestamp of the last executed activity and the start timestamp of the first executed activity.

Cost Dimension: Case Cost. Costs can be annotated for every event of an event log through the information contained in cost models obtained from management accounting [18]. Using such an annotated event log, it is possible to calculate the total cost of a case as the sum of the cost of all of its activities.

Flexibility Dimension: Optionality. In [9], the metric proposed for the calculation of a flexibility PPI is optionality. As explained in [9], optionality refers to the degree of variation that a process permits. Considering the number of optional activities within the event log, its degree of optionality (*opt*, OPT in [9]) can be quantified. *opt* can be calculated as shown in (1)-left, where A (T in [9]) is the number of activities in the event log, and A_{opt} (T_{opt} in [9]) is the number of activities in the event log that are optional. An activity is optional if there is at least one case where it does not occur [9].

Extending from [9], the current work proposes to calculate *opt* for a single case c through a straightforward adaptation of (1)-left. Given the known set of optional activities for the event log, the number of optional activities executed in c can be counted. Thus, opt_c can be calculated as shown in (1)-center, where A_{opt_c} is the number of distinct activities of the case that are optional within the event log and A_c is the number of distinct activities of the case.

Based on (1)-center, it is possible to calculate *opt* for a group of cases C as the expected value for the fraction opt_c. opt_C is shown in (1)-right. It must be noted that when C equals the event log, L, opt_C does not equal *opt*. However, *opt* acts as the upper bound for opt_C.

$$opt = \frac{A_{opt}}{A} \qquad opt_c = \frac{A_{opt_c}}{A_c} \qquad opt_C = \frac{\sum\limits_{c \in C} A_{opt_c}}{\sum\limits_{c \in C} A_c} \tag{1}$$

Quality Dimension: Repeatability. In [9], it is stated that the quality of a product created in a process is often not directly visible from event logs, but that a good indication is to check whether there are repetitions, as they typically occur when an activity has not been completed successfully. Thus, [9] proposes a repeatability (*rep*, r in [9]) metric for every activity a of the event log, based on Cycle Time (CT) and Execution Time (T), as shown in (2)-left.

CT_a is calculated as the average execution time of the activity a per case, whereas T_a is the average execution time of the activity per instantiation (every time the activity is performed). As [9] explains, $CT_a = TT_a/C_a$ and $T_a = TT_a/I_a$, where TT_a is the sum of all execution times of a, C_a is the number of cases in the event log containing a, and I_a is the number of instantiations of a in the event log. Extending from [9], the current work highlights that the fraction in (2)-left can be rewritten to only consider C_a and I_a, as shown in (2)-right.

$$rep_a = 1 - \frac{T_a}{CT_a} \qquad rep_a = 1 - \frac{TT_a/I_a}{TT_a/C_a} = 1 - \frac{C_a}{I_a} \tag{2}$$

From (2)-right, and given A the set of all activities in the event log, the current work proposes to calculate the overall *rep* of an event log as shown in (3)-left. Then, to calculate *rep* for a single case c, the numerator and denominator of (3)-left can be converted to instance granularity equivalents. In the numerator of (3)-left, as C_a is defined as the number of cases containing a, when calculated in the context of a single case, every C_a is equal to 1. Thus, the sum of C_a for a single case is equivalent to the number of distinct activities of the case, A_c. In the denominator of (3)-left, the sum of I_a equals the number of events in the event log. Thus, the number of events in the case, E_c, is used as an equivalent within the context of a single case. This is shown in (3)-center.

Based on (3)-center, it is possible to calculate *rep* for a group of cases C as the expected value for rep_c. rep_C is shown in (3)-right. When C equals the event log, $rep_C = rep$.

$$rep = 1 - \frac{\sum\limits_{a \in A} C_a}{\sum\limits_{a \in A} I_a} \qquad rep_c = 1 - \frac{A_c}{E_c} \qquad rep_C = 1 - \frac{\sum\limits_{c \in C} A_c}{\sum\limits_{c \in C} E_c} \tag{3}$$

3.2 Step 2 - Event Log Annotation

This intermediate step consists of incorporating the calculated metrics into the event log. In practice, this can be done by adding additional columns to a comma-separated event log or including the metrics as case attributes if the event log is in a specific format (e.g., XES). It is also possible that the metrics are calculated within the same environment as the next steps of the method, which would not require directly annotating the event log, but rather making a relation between the event log and the calculated metrics.

3.3 Step 3 - Filter Generation

With the annotated event log, it is possible to generate the proposed filters. Each slider associated to a filter ranges from the minimum to the maximum observed value among all process instances. For example, consider a process where the shortest instance had a duration of 240 min and the longest had a duration of 1,150 min. Similarly, the cheapest instance costs 95 USD, whereas the most expensive one costs 500 USD. Both for optionality and repeatability, the minimum obtained values were 0, whereas the maximum values were 0.5. Figure 2 shows the appearance that the sliders would have, given the values above.

Moving either of these sliders would have a filtering effect over the cases considered by the process model. For example, Fig. 3 shows how moving the slider for the cost dimension to consider only those instances with costs in the 175–225 USD range, forces the recalculation of the minimum and maximum values of all other sliders only to consider instances with costs within that particular range.

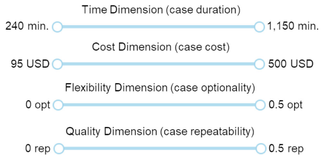

Time Dimension (case duration)
240 min.
1,150 min.
Cost Dimension (case cost)
95 USD
500 USD
Flexibility Dimension (case optionality)
0 opt
0.5 opt
Quality Dimension (case repeatability)
0 rep
0.5 rep

Fig. 2. Example of the sliders generated through the proposed method.

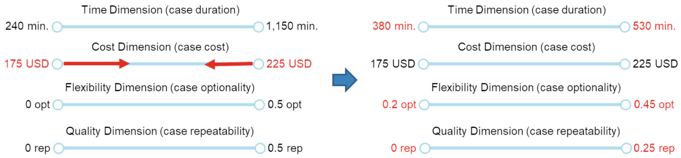

Time Dimension (case duration)
240 min.
1,150 min.
Cost Dimension (case cost)
175 USD
225 USD
Flexibility Dimension (case optionality)
0 opt
0.5 opt
Quality Dimension (case repeatability)
0 rep
0.5 rep
Time Dimension (case duration)
380 min.
530 min.
Cost Dimension (case cost)
175 USD
225 USD
Flexibility Dimension (case optionality)
0.2 opt
0.45 opt
Quality Dimension (case repeatability)
0 rep
0.25 rep

Fig. 3. Example of the effects of constraining a slider (cost) into the other sliders.

3.4 Step 4 - Process Analysis

The final step is to analyze the process model with the support of the generated sliders. For this, process analysts can define constraints of interest based on their business needs. Some examples are provided next:

- Some process instances have been failing to complete within the deadline. The analyst wants to compare the paths followed by these instances and those that are on time. They expect that there will be noticeable differences within the other dimensions between both groups.
- Due to new budgeting limitations, the analyst wants to analyze the process by considering instances within certain cost limits, identifying common features among those instances, and determining possible improvements through trade-offs between costs and the other dimensions.
- There has been an increase in complaints from customers regarding the services provided by the company. The analyst wants to explore process instances with high repeatability and low optionality. This might provide insights on how to reduce rework, or how to improve process adaptability.

In Sect. 5, through an implementation of the proposed method, further analysis examples are provided for the process described in Sect. 4.

4 Illustrative Example

An example based on the blasting process of a Chilean copper mine is used to illustrate the proposed method. A model with the activities of this process is publicly available in Appendix 4 of [2]. Blasting is one of the main methods used in the mining industry to fragment hard rock minerals [4]. It is a complex process that requires coordination of various activities between the blasting crew and other workers laboring in the mining operation [6].

The process depicted in [2] mainly consists of 3 sub-processes. First, the state of perforations required for inserting the blasting explosives and the terrain are verified, and corrections are done accordingly. Next, after loading the explosives and evacuating the blasting zone, the blasting itself takes place. Finally, depending on the results of the initial blasting, a blasting revision may be performed, and the corresponding reports are generated. Four roles are involved in the process: Turn Lead, who is in charge of coordination activities; Drilling Operator, who is required during the revision of the polygon pits; Blasting Operator, who loads the explosives and executes the blasting; and Specialist Engineer, who is tasked with activities related to planning. When a revision of the terrain is required, the blasting team must coordinate this activity with the Ground Movement and Production Support (GM&PS) team.

Since the above process does not consider rework, the repeatability metric would always be zero. To address this, two adaptations were made to the original process. The first is that multiple terrain revisions per case can be requested. The second is that, after executing a blasting revision, the blasting results are reviewed again, which allows the possibility of executing additional blasting revisions during each process instance. Figure 4 shows this adapted process model.

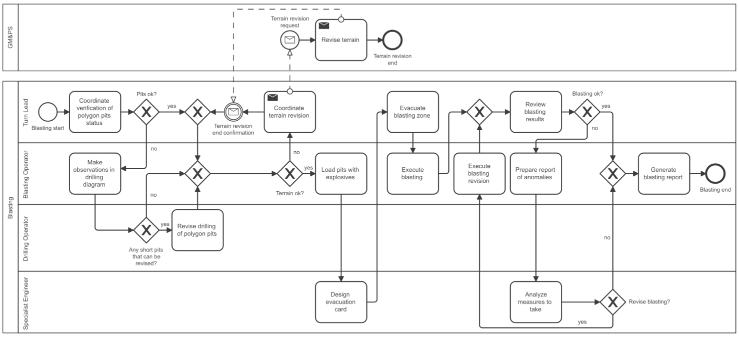


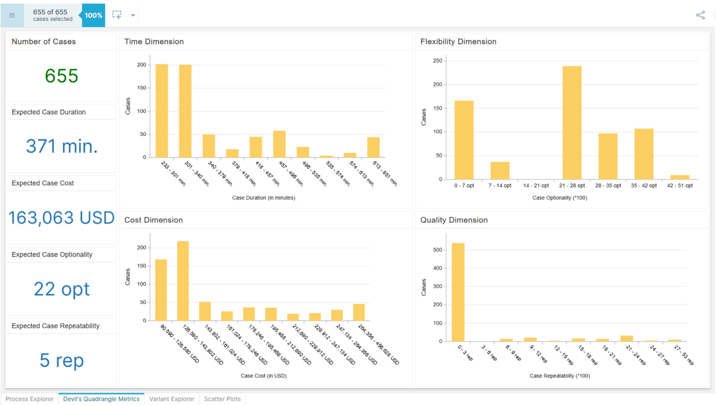
Fig. 4. Adapted process model of the blasting process of a Chilean copper mine.

Several process instances were simulated through a Python script. To achieve this, duration and waiting time ranges were defined for every activity. Resources and cost information were also considered. An occurrence probability was also given to every diverging exclusive gateway in Fig. 4. This way, an event log containing 655 process instances was generated. This event log can be obtained from https://bit.ly/blasting_with_rework_log.

5 Implementation

To utilize the method proposed in this work, the first step, calculating the case metrics, must be performed. A Python function has been generated, which receives a cost-annotated event log as input and then calculates the metrics for each case in the event log. The function has been implemented in a Google Colaboratory notebook, which contains functions to load an event log and to export the calculated metrics in a comma-separated format. The notebook can be found in https://bit.ly/dq_metric_calculation.

The implementation of the remaining steps of the method was done in Celonis. This software allows loading multiple files and defining connections between them to facilitate process analysis. Thus, both the event log and the file with the calculated metrics were uploaded and connected through the case ID. Next, a process analysis dashboard was created for the third step of the proposed method. The dashboard, applied to the illustrative example in Sect. 4, can be accessed in https://bit.ly/dq_dashboard. This dashboard contains four views. Figure 5 shows the view that has been constructed with the proposed filters. The other three views correspond to a process explorer, a variant explorer, and a view containing scatter plots that allow contrasting pairs of dimensions.



Number of Cases
655

Expected Case Duration
371 min.

Expected Case Cost
163,063 USD

Expected Case Optionality
22 opt

Expected Case Repeatability
5 rep

Fig. 5. DQ metrics view in a Celonis dashboard.

Celonis does not possess a feature for generating sliders. However, the histogram component has been used instead, as it allows a close approximation to the conceptual design. the X-axis of every histogram contains ranges calculated for every metric, based on their minimum and maximum computed values, whereas the Y-axis indicates the number of cases contained within every range. Due to the limitations of Celonis, the repeatability and optionality metrics are multiplied by 100, as they are rounded to either 0 or 1 otherwise. There are two main differences between this implementation and the conceptual design. On one hand, the histograms group the cases in buckets based on ranges defined by Celonis. This complicates filtering based on specific values. On the other hand, a histogram visualization facilitates observing how cases are spread within the metric. PPI showing the expected values of the currently filtered cases for all four metrics, as well as the number of selected cases, are also included in this view. While filtering, the duration, cost, and repeatability PPI will change color between green, blue, and red if the current average is below, within, or above the standard deviation of the unfiltered average, respectively. The optionality PPI will change color between red, blue, and green since a higher OPT value is generally the desired outcome [9].

To illustrate the exploratory analysis enabled by this method, suppose that a process analyst has noticed the occurrence of process instances exceeding an expected duration limit of 420 min. Figure 6(a) shows how the analyst can filter the process by selecting cases with a duration near or above this limit. Once the filter is applied, all histograms, PPI, and the process and variant explorers are updated to only consider the cases compliant with the filter.

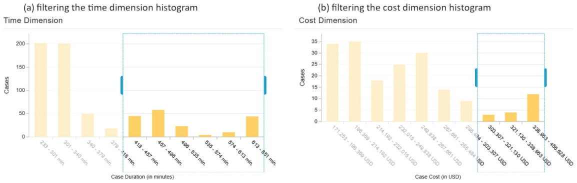
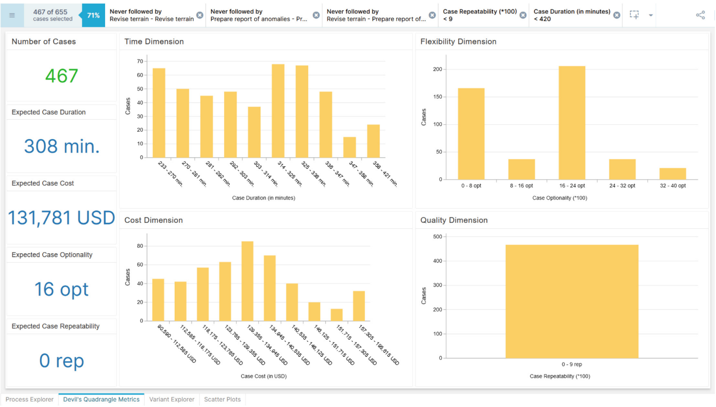


Fig. 6. Filtering on case duration and case cost through the corresponding histograms.

The first observation of the analyst is that most of the selected instances either consider a terrain revision, or anomalies are detected after reviewing the blasting results, which often results in a blasting revision. However, a longer duration is expected in these cases, as more activities must be performed during them. Figure 6(b) shows how, to further analyze the situation, the analyst selects the cases with high execution costs. This reveals the existence of cases where several terrain revisions were required and others where several blasting revisions had to be executed.

To alleviate the above situation, the analyst considers improving the process so that the terrain only needs to be revised up to once per case, anomalies can only occur once per case, and no case contains both terrain revision and detection of anomalies. After applying filters to observe the expected process behavior with these constraints, the analyst observes that some cases still take over 420 min. Moreover, the analyst notices that there are cases with a repeatability of 9 or higher, which are also undesired. Through further analysis, all these cases correspond, once more, to instances where either a terrain or a blasting revision was required. This drives the analyst to suggest improving the coordination with the GM&PS team, and to further train the blasting operators. By redesigning the process based on this exploratory analysis, the analyst expects considerable reductions in the average case duration, cost, and repeatability. However, the average optionality would also be reduced, causing a weakening of the flexibility dimension. These results align with the idea behind the DQ. Figure 7 shows the metrics view after applying the above filters, which allows the analyst to observe the expected process performance once the redesign be performed.



467 of 655 cases selected 71%

Never followed by Revise terrain - Revise terrain

Never followed by Prepare report of anomalies - Pr...

Never followed by Revise terrain - Prepare report of...

Case Repeatability (*100) < 9

Case Duration (in minutes) < 420

Number of Cases: 467

Expected Case Duration: 308 min.

Expected Case Cost: 131,781 USD

Expected Case Optionality: 16 opt

Expected Case Repeatability: 0 rep

Fig. 7. DQ metrics view of the Celonis dashboard after filter application.

6 Conclusions

This work presented a method for enhancing the exploratory analysis of processes discovered through PM. Inspired by the DQ, the method allows understanding the behavior of the time, costs, flexibility, and quality dimensions of existing process instances. By calculating metrics for PPI at a process instance granularity, the instances can be further analyzed through PM filters that define constraints over the performance metrics. It is expected that the method will be of use for process analysts when exploring process instances from a performance perspective. The method will facilitate the identification of redesign and optimization opportunities based on their effects on the dimensions considered by the method.

As future work, limitations to the proposed method should be addressed. First, the current method considers one PPI per process dimension. However, additional PPI that can be calculated at an instance level could be considered. For example, service time, that is, the sum of activity durations [1], for the time dimension; fixed and variable cost [18], for the cost dimension; the number of directly-follows relations between events [8], for the flexibility dimension; and the number of automated activities [12], for the quality dimension. With additional PPI for every dimension, it would be possible to only consider those that are of interest for every process, based on their specific particularities. Second, the method has been illustrated through an example consisting of simulated process instances. It is desirable to validate the utility of the method through its application over real data. And third, the current implementation requires two tools for using the method: metric calculation through a Python function, and filter generation and analysis through a Celonis dashboard. The implementation of all four steps of the proposed method in a single tool is desirable.

References

1. van der Aalst, W.: Process Mining: Data Science in Action. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. Acuña Maldonado, J.E.: Modelo de Gestión de Operaciones de la mina Rajo Abierto de la División Andina de CODELCO. Thesis, Pontificia Universidad Católica de Valparaíso (2008)
3. Ahmad, S., Muzaffar, S.I., Shahzad, K., Malik, K.: Using BPM frameworks for identifying customer feedback about process performance. In: Matulevičius, R., Dijkman, R. (eds.) CAiSE 2018. LNBIP, vol. 316, pp. 55–69. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92898-2_5
4. Akande, J.M., Aladejare, A.E., Lawal, A.I.: Evaluation of the environmental impacts of blasting in Okorusu fluorspar mine, Namibia. Int. J. Eng. Technol. **4**(2), 101–108 (2014)
5. Avstrievskikh, I.: Service robots in hospitality industry: processes analysis of restaurants with full table service in Helsinki. Thesis, Arcada (2021)
6. Bajpayee, T.S., Rehak, T., Mowrey, G.L., Ingram, D.: Blasting injuries in surface mining with emphasis on flyrock and blast area security. J. Safety Res. **35**(1), 47–57 (2004)
7. Brand, N., Van der Kolk, H.: Workflow analysis and design (1995)
8. Cho, M., Song, M., Park, J., Yeom, S.R., Wang, I.J., Choi, B.K.: Process mining-supported emergency room process performance indicators. Int. J. Environ. Res. Public Health **17**(17), 6290 (2020)
9. Dumas, M., La Rosa, M., Mendling, J., A Reijers, H.: Fundamentals of Business Process Management. Springer, Heidelberg (2013)
10. Estrada-Torres, B., del Río-Ortega, A., Resinas, M.: Defining process performance measures in an object-centric context. In: Cabanillas, C., Garmann-Johnsen, N.F., Koschmider, A. (eds.) BPM 2022. LNBIP, vol. 460, pp. 210–222. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-25383-6_16
11. Haberberger, N.: Data linkage of internal sensor data with external data sources and analysis of generated relationships under usage of distributed systems. Thesis, Technische Universität Wien (2019)
12. van den Ingh, L.O.: Evaluating business process performance based on process mining. Thesis, Eindhoven University of Technology (2016)
13. Kis, I., Bachhofner, S., Di Ciccio, C., Mendling, J.: Towards a data-driven framework for measuring process performance. In: Reinhartz-Berger, I., Gulden, J., Nurcan, S., Guédria, W., Bera, P. (eds.) BPMDS/EMMSAD -2017. LNBIP, vol. 287, pp. 3–18. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59466-8_1
14. de Kort, J.: Business entity lifecycle modelling in simulation-aided analysis: a case study. Thesis, Eindhoven University of Technology (2015)
15. Leemans, S.J., Poppe, E., Wynn, M.T.: Directly follows-based process mining: exploration & a case study. In: ICPM 2019, pp. 25–32. IEEE (2019)
16. Marin-Castro, H.M., Tello-Leal, E.: Event log preprocessing for process mining: a review. Appl. Sci. **11**(22), 10556 (2021)
17. Muurinen, A.: Process performance indicators for measuring order to cash process. Thesis, Arcada (2020)
18. Nauta, W.: Towards cost-awareness in process mining. Thesis, Eindhoven University of Technology (2011)
19. Reijers, H.A., Mansar, S.L.: Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. Omega **33**(4), 283–306 (2005)

20. Sani, M.F., van Zelst, S.J., van der Aalst, W.M.P.: Improving process discovery results by filtering outliers using conditional behavioural probabilities. In: Teniente, E., Weidlich, M. (eds.) BPM 2017. LNBIP, vol. 308, pp. 216–229. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74030-0_16
21. Tsakalidis, G., Vergidis, K.: A roadmap to critical redesign choices that increase the robustness of business process redesign initiatives. J. Open Innov.: Technol. Mark. Complexity **7**(3), 178 (2021)

11th International Workshop on DEClarative, DECision and Hybrid approaches to processes (DEC2H 2023)

11th International Workshop on DEClarative, DECision and Hybrid approaches to processes (DEC2H 2023)

Rules and decisions play a crucial role in defining the boundaries of behavior and influencing the attainment of process objectives. In the realm of business processes, rule-based decisions are frequently encountered, especially in knowledge-intensive processes that operate within dynamic and adaptable contexts. When depicting such processes, it becomes necessary to use languages that articulate the governing rules and decisions of flexible business processes with high variability. Imperative languages, such as BPMN or BPEL, depict predefined pathways, but they may prove inadequate when representing adaptable and diverse rules and decisions. Declarative modeling paradigms and their hybridization with imperative models have emerged as a solution to this challenge, aiming to directly encapsulate the business rules and constraints that underlie the process. The flexibility of these solutions facilitates the description of permitted and prohibited actions in a business process instead of the most common path. Several languages have been introduced for this purpose, including DECLARE, Dynamic Condition Response (DCR) Graphs, Decision Modelling and Notation (DMN), fragment-based Case Management (fCM), Case Management Model and Notation (CMMN), Guard Stage Milestone (GSM), and Declarative Process Intermediate Language (DPIL). Notably, there has been a growing interest in *hybrid* approaches that blend the strengths of both declarative and procedural modeling paradigms.

The International Workshop on DEClarative, DECision and Hybrid approaches to processes (DEC2H) serves as a platform for sharing insights into the latest challenges and innovative solutions pertaining to the modeling of rules and decisions. This encompasses all stages of the Business Process Management lifecycle, spanning from the initial stages of identification and discovery to the subsequent phases of analysis, redesign, implementation, and ongoing monitoring. Furthermore, DEC2H explores the practical applicability of these approaches in real-world scenarios.

DEC2H 2023 received nine high-quality international submissions. Each paper was single-blind reviewed by at least three members of the Program Committee. Of all the submitted manuscripts, the top four were accepted to be presented on 12 September 2023.

Chiara Di Francescomarino gave the invited talk to open the workshop with a presentation entitled “What else can we do with temporal patterns?” Di Francescomarino gave a new perspective on the use of declarative process models for supporting other functionalities in fields like Predictive and Prescriptive Process Monitoring. This presentation offered an overview of unconventional applications of temporal patterns. She explained how by providing an appropriate depiction of temporal information, temporal patterns can serve as encoding mechanisms. They offer the possibility of creating loosely constrained recommendations and concise discriminative explanations for variations between sets of temporal data. She added that temporal patterns can be employed to succinctly represent prior knowledge, which can, in turn, guide predictive models and explainers to align with predefined rules.

Anti Alman *et al.* presented a log generation approach to generate event logs based on both procedural and declarative process models as input.

Juan Camilo Rosero *et al.* proposed a solution to generate traces that are compliant with a declarative process model (in the form of a DCR graph) and Pareto optimal with respect to a potentially multi-dimensional cost function.

Edyta Brzychczy *et al.* presented an interesting application of hybrid process modeling in the industrial context.

Carl Corea *et al.* investigated errors associated with DMN models and delved into the different types of these errors.

We thank the authors for their noteworthy contributions and the members of the Program Committee for their invaluable help in the reviewing and discussion phases. We hope that the reader will benefit from reading these papers to learn more about the latest advances in research about declarative, decision, and hybrid approaches to business process management.

September 2023

Claudio Di Ciccio
María Teresa Gómez-López
Tijs Slaats
Jan Vanthienen

Organization

Workshop Chairs

Claudio Di Ciccio	Sapienza University of Rome, Italy
María Teresa Gómez-López	Universidad de Sevilla, Spain
Tijs Slaats	University of Copenhagen, Denmark
Jan Vanthienen	KU Leuven, Belgium

Program Committee

Amine Abbad Andaloussi	University of St Gallen, Switzerland
Banu Aysolmaz	Eindhoven University of Technology, The Netherlands
Christoffer Olling Back	University of Copenhagen, Denmark
Andrea Burattin	Technical University of Denmark, Denmark
Carl Corea	University of Koblenz-Landau, Germany
João Costa Seco	Universidade NOVA de Lisboa, Portugal
Johannes De Smedt	KU Leuven, Belgium
Chiara Di Francescomarino	University of Trento, Italy
Rik Eshuis	Eindhoven University of Technology, The Netherlands
Amin Jalali	Stockholm University, Sweden
Krzysztof Kluza	AGH University of Science and Technology, Poland
Hugo A. López	University of Copenhagen, Denmark
Fabrizio Maria Maggi	Free University of Bozen-Bolzano, Italy
Andrea Marrella	Sapienza University of Rome, Italy
Artem Polyvyanyy	University of Melbourne, Australia
Andrey Rivkin	Technical University of Denmark, Denmark
Flavia Santoro	UERJ, Brazil
Stefan Schönig	University of Regensburg, Germany
Mathias Weske	HPI, University of Potsdam, Germany
Han van der Aa	University of Mannheim, Germany

Generating Event Logs from Hybrid Process Models

Anti Alman[1](✉), Fabrizio Maria Maggi[2], Marco Montali[2], and Andrey Rivkin[3]

[1] Institute of Computer Science, University of Tartu, Tartu, Estonia
anti.alman@ut.ee
[2] Free University of Bozen-Bolzano, Bolzano, Italy
{maggi,montali}@inf.unibz.it
[3] Institute of Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark
ariv@dtu.dk

Abstract. To carry on controlled experiments in process mining, it is necessary to generate event logs with specific characteristics. This has led to the development of log generation techniques in which a process model is simulated to generate an event log that is both compliant with the process model and also has certain user-defined properties (e.g., a certain number of traces, traces with certain lengths, etc.). Such techniques are available for a variety of modeling languages, both procedural and declarative. However, they are limited to simulating a single (procedural or declarative) process model at a time and do not allow simulating concurrent executions of multiple separate, but interacting, processes. In this paper, we introduce a log generation approach that takes multiple (procedural and declarative) process models (i.e., a Hybrid Business Process Representation) as input and produces an event log matching the concurrent execution of these models on the same case instances. We discuss the details of our approach and evaluate its implementation.

Keywords: Log Generation · Process Simulation · Hybrid Process Model · Data Petri net · Declare · Finite State Automaton

1 Introduction

A large number of different techniques have been developed over the past few decades for analyzing business processes, with a large portion of these techniques coming from the field of process mining [1]. The core artifact of process mining techniques is an event log, which contains sequences of events representing valid executions of a process. Processes themselves can be represented using procedural or declarative process models. The procedural paradigm is very well suited for business processes where all possible executions are relatively similar to each other. A shortcoming of the procedural paradigm is the difficulty of representing

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 289–301, 2024.
https://doi.org/10.1007/978-3-031-50974-2_22

processes that have high variability among all possible executions as each variation needs to be explicitly accounted for, thus leading to high complexity in the process models [14]. An alternative, the declarative paradigm, instead, assumes that every process execution is allowed by default, and then constrains these executions by defining rules to be followed throughout the process execution. This leads to process descriptions which allow for far more flexibility [2].

However, often, business processes contain both parts that can be easily represented with the procedural paradigm and parts that are better represented using declarative constraints. For example, in the healthcare domain, some clinical guidelines are procedural by nature, but declarative medical knowledge needs to be applied on top of these guidelines to deal with the heterogeneity of the patients being treated [22]. For this reason, there has been growing interest in developing hybrid approaches to combine the declarative and the procedural paradigm in order to gain the advantages of both [5].

The development and testing of such hybrid approaches can be significantly hampered by the availability of suitable event logs, especially in cases where the developed approaches need to be tested in a controlled way. In process mining, controlled experiments are, in general, carried out by using log generation techniques, which simulate an existing model to generate a corresponding event log. Many techniques are already available for different (procedural and declarative) modeling languages [6,8–10,13,18]. However, all these techniques are limited to simulating a single (procedural or declarative) process model, thus precluding simulating any form of Hybrid Business Process Representations (HBPRs) [5].

To overcome these limitations, we introduce a data-aware hybrid log generator based on the semantics for HBPRs introduced in [3,4]. In particular, this log generator can simulate the concurrent execution of any combination of declarative and procedural process models on the same case instances to generate a corresponding event log. This enables log generation for several conceptually different use cases, such as log generation from multiple process models with shared activities, simulation of a single process model defined as a set of smaller (but still overlapping) sub-models, log generation from procedural process models in the presence of additional declarative constraints and from declarative process models in the presence of additional procedural components, or simulation of multiple procedural process models connected through declarative constraints.

In this paper, we opt for Data Petri nets to represent procedural process models and a data-aware variant of the DECLARE language to represent declarative process specifications. These languages were selected as their models can be represented in the form of finite state automata. In fact, the log generator we present in this paper (publicly available at: https://github.com/antialman/model-interplay-loggen-code) relies on this automata-based representation of the HBPR to be simulated. Any other modeling language whose models can be translated into equivalent finite state automata can be used in our approach.

The rest of this paper is structured as follows. Section 2 discusses the core background concepts. Section 3 outlines our interpretation of concurrent execution of multiple models. Section 4 describes the log generation approach. Section 5 discusses the performance of the approach. Section 6 discusses the related work. Section 7 concludes the paper and spells out directions for future work.

2 Background

In this section, we describe the relevant background concepts of our approach. More specifically, we introduce: (i) the data-aware extensions of Petri nets and DECLARE that our approach takes as inputs, (ii) finite state automata used as an intermediary data structure, and (iii) event logs produced as output.

Data Petri Nets. As a representative of the procedural paradigm, we opt for Data Petri nets (DPNs) [12,20], which extend the traditional place-transition nets by allowing the assignment of read and write conditions (called *guards*) to the transitions of the Petri net. In our setting, DPN transitions represent the possible events of the process, DPN markings (i.e., the assignments of tokens to places) represent possible states of the process, and guards represent the relationship of process data to process control flow.

More formally, we assume a finite set $\mathcal{E} = \{\epsilon | \epsilon \in \langle n, A\rangle\}$ of event signatures, where: n is the *event* name and $A = \{a_1, \ldots, a_\ell\}$ is the set of event *attributes* (ℓ being the number of attributes). With $\mathcal{N}_\mathcal{E}$, we denote the set of all event names from $\mathcal{E}$, and with $\mathcal{A}_\mathcal{E}$, the set of all attribute names occurring in $\mathcal{E}$. Then, a *Data Petri net* D over the set $\mathcal{E}$ of is a tuple $\langle P, T, F, l, V, r, w\rangle$, where: *(i)* (P, T, F) is the Petri net graph; *(ii)* $l : T \rightarrow \mathcal{N}_\mathcal{E} \cup \{\tau\}$ is a *labeling* function (here τ denotes a *silent* transition); *(iii)* $V \subseteq \mathcal{A}_\mathcal{E}$ is the set of net's *variables*; *(iv)* $r : T \rightarrow \mathcal{G}_\mathcal{E}$ (resp., $w : T \rightarrow \mathcal{G}_\mathcal{E}$) is a *read* (resp., *write*) *guard-assignment* function, mapping every transition $t \in T$ into a read (resp., write) guard – a boolean formula $\mathcal{G}_\mathcal{E}$ whose components are atomic expressions of the form $a \odot c$, where $\odot$ is a type-specific comparison predicate and c is a constant defined in the given guard.

A DPN transition is *enabled* if and only if its read and write guards are satisfied under a given "firing mode" – a function that assigns values only to variables of the guards – and all the input places of the transition contain sufficiently many tokens to consume. To check the read guard, the firing mode function picks values currently available in the net's state, while the write guard, instead, updates the net variables to values that would satisfy the guard. When a transition is enabled, it may *fire* by consuming the necessary amount of tokens from its input places and producing the necessary amount of tokens in its output places, and by updating all the values assigned to variables in the write guard using the firing mode function. Values assigned to all other variables remain untouched.

For each DPN, we assume an initial marking (i.e., marking before any transition has fired) and a final marking (i.e., the expected marking at the end of process execution). Finally, we limit ourselves to DPNs that are 1-*bounded* and well-formed over their respective set of event signatures.

Multi-perspective Declare with Local Conditions. As a representative of the declarative paradigm, we opt for an extension of the DECLARE language [7], which we refer to as multi-perspective DECLARE with local conditions

(LMP-Declare). Both Declare and LMP-Declare describe a business process as a set of *constraints*, where each constraint defines either a control flow relation between two events (e.g., if `A` occurs then `B` must also occur) or the cardinality of a single event (e.g., `A` can occur at most twice per process execution). LMP-Declare further enriches the event references with data conditions over the event attributes, allowing the modeler to specify, for example, that `A` with $x = 2$ can occur at most twice per process execution (while leaving occurrences of `A` with any other value of x unconstrained). All Declare constraints can be represented as formulas of Linear Temporal Logic over finite traces (LTL$_f$) [21], and that can be extended to LMP-Declare via data abstraction techniques.

The syntax of LMP-Declare constraints is the following:

$$\Phi := \top \mid \varphi \mid \mathbf{X}\,\Phi \mid \mathbf{F}\,\Phi \mid \mathbf{G}\,\Phi \mid \Phi_1 \mathbf{U}\,\Phi_2 \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2$$

Here, φ is a boolean combination of attribute-to-constant comparisons and event variables ranging over the set of all event names from $\mathcal{E}$. Notice that the language of boolean combinations of attribute-to-constant comparisons without event variables closely resembles that of variable-to-constant conditions in [12], thus providing a good basis for combining DPNs with LMP-Declare. As in standard LTL$_f$, $\mathbf{X}$ denotes the *strong next* operator (which requires the existence of a next state where the inner formula holds), while $\mathbf{U}$ stands for *strong until* (which requires the right-hand formula to eventually hold, forcing the left-hand formula to hold in all intermediate states). The other two LTL$_f$ operators are defined as $\mathbf{F}\,\Phi = \top\mathbf{U}\,\Phi$ (*eventually*) and $\mathbf{G}\,\Phi = \neg\mathbf{F}\,\neg\Phi$ (*globally*).

Deterministic Finite State Automata. In our approach, we transform the input process models into a deterministic finite state automaton (DFA), which we then use to simulate the concurrent execution of the input process models. Formally, a DFA is a labeled transition system $\mathcal{D} = \langle L, S, \delta, s_0, S_\mathrm{f}\rangle$ defined over states S and a set of labels L, having $\delta : S \times L \to S$ as the transition function, i.e., a function that, given a starting state and a label, returns the target state (if defined). $s_0 \in S$ is the initial state of $\mathcal{A}$, and $S_\mathrm{f} \subseteq S$ is the non-empty set of its accepting states ($S_\mathrm{f} \neq \emptyset$).

In our setting, the initial state of the DFA corresponds to the state of the process before any event has occurred, accepting states correspond to the final marking of a DPN and/or states in which the LMP-Declare constraints are satisfied, and the set of labels L corresponds to encoded event signatures from $\mathcal{E}$ (i.e., possible events of the business process, including the values of relevant event attributes, encoded as strings). We note that any sequence of labels, which, starting from s_0 and using the transition function δ, reaches any state in S_f, is considered to be accepted by the DFA. Furthermore, DFA are closed under the product operation × and the product of two DFAs accepts the sequences of labels accepted by both operands. In our approach, we rely on both these properties to generate event logs based on the input process models.

Event Logs. Event logs are the central artifacts in process mining and they can be seen as recordings of the executions of a business process [15]. Event logs

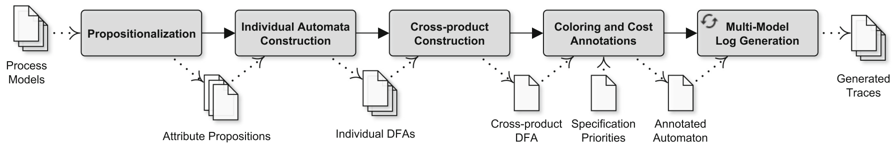


Fig. 1. Main steps and artifacts of the hybrid log generation approach.

are most commonly stored in a standardized file format called XES (eXtensible Event Stream) [16], which is structured as a set of traces, each corresponding to a single execution of a business process and consisting of an ordered set of events. Each event has a name (corresponding to the process activity executed) and may also contain an additional data payload of event attributes along with their concrete value assignments. Note that this is in line with our notion of event signatures E, which we encode into the set of DFA labels L. This allows us to use the DFA to determine sequences of events accepted by that DFA. The details of this procedure are presented in Sect. 4.

3 Interpretation of Concurrent Executions

In line with our earlier works [3,4], we interpret the concurrent execution of DPN and LMP-Declare models (for globally valid executions) as follows:

- The execution of a DPN starts from its initial marking and must reach its final marking.
- The execution of an LMP-Declare model starts with each individual constraint of that model in its initial state and must end with all of them in an accepting state.
- Events with the same name in two or more process models are considered as the same event (i.e., same-labeled events are merged for concurrent execution).
- The directly-follows relations imposed by LMP-Declare constraints (e.g., chain response) have priority over the execution of any other event that is allowed by the input models.
- Events with the same name in two or more process models can be executed whenever they are allowed by all models containing that event (unless prevented by a directly-follows relation as mentioned in the previous point).
- Each DPN variable is private to the DPN defining that variable (i.e., write guards cannot update variables of other DPNs).
- If multiple models impose conditions on the value of the same event attribute, the generated value must match the intersection of those conditions.

4 Approach

We modify and adapt specific parts of our earlier works on multi-model monitoring [3,4] to provide a data-aware hybrid log generation approach. An overview of the main steps of the resulting approach, and its intermediary artifacts, is provided in Fig. 1. These steps are further detailed in the following subsections.

4.1 Propositionalization

As the first step, we apply data abstraction techniques to handle the data perspective of the input process models. More specifically, we first find all atomic data conditions across all models. Then, for each attribute in that set of conditions, we order all constants that this attribute is compared against. Finally, these constants, and the intervals between them, are enumerated for each attribute, thus producing attribute propositions necessary for the next step of the approach. An example of propositionalization in the case of the atomic conditions $x > 0$, $x \leq 5$, $x > 5$, and $x < 8$ is given in Table 1. Note that we also encode the attribute names ($att0$ represents attribute x in the given example).

Table 1. Propositionalization for conditions $x > 0$, $x \leq 5$, $x > 5$, and $x < 8$.

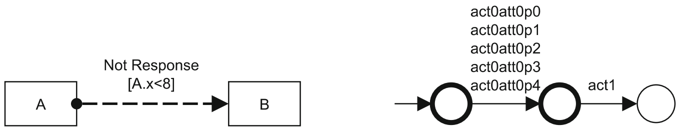
Constant		0		5		8	
Interval	$(-\infty, 0)$	$[0, 0]$	$(0, 5)$	$[5, 5]$	$(5, 8)$	$[8, 8]$	$(8, \infty)$
Interval id	p0	p1	p2	p3	p4	p5	p6
Proposition	att0p0	att0p1	att0p2	att0p3	att0p4	att0p5	att0p6

4.2 Individual DFA Construction

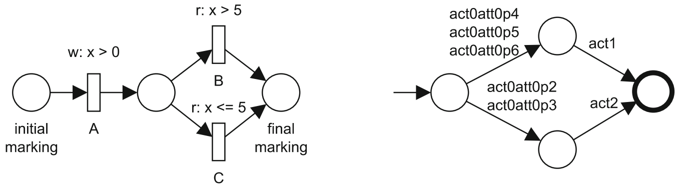
The next step after propositionalization is the construction of one DFA for each input process model using the attribute propositions.

The procedure for LMP-Declare models is fairly straightforward as it uses exactly the same set of templates as standard Declare, and equivalent LTL$_f$ formulas are defined for all of them. These formulas can be applied in our setting by using, instead of only the event name, a conjunction of the event name with all attribute propositions matching the condition on that event. For example, event `A` with condition $x < 8$ (assuming the attribute propositions in Table 1) would be replaced with $act0att0p0 \vee act0att0p1 \vee \ldots \vee act0att0p4$, where $act0$ corresponds to event `A`. An LTL$_f$ formula is created for each constraint in the LMP-Declare model, and a DFA of the entire model is constructed based on the conjunction of these formulas using existing automata construction techniques [23]. An example of an LMP-Declare constraint with the corresponding DFA is shown in Fig. 2(a) (thicker circles indicate accepting states).

The procedure for DPNs is more complex and for a full description we refer the reader to [4]. However, the basic idea is to construct a reachability graph of the DPN markings, where, analogously to LMP-Declare models, conjunctions of event name with the attribute propositions matching the corresponding DPN guard are used as the graph labels. An example of a DPN with the corresponding



(a) An LMP-Declare constraint and the corresponding DFA.



(b) A DPN and the corresponding DFA.

Fig. 2. DFAs with propositionalized events.

DFA is given in Fig. 2(b). At first glance, this example may seem counter intuitive as the DPN seems to have a different structure than the DFA. This difference is caused by the presence of the write guard on event `A`, which we handle by splitting the automaton into "regions" representing the value written by the guard. In Fig. 2(b), if `A` occurs with $x > 5$ then the next event must be `B` (represented by $act1$), otherwise the next event must be `C` (represented by $act2$).

Notice that the automata in Fig. 2(a) and Fig. 2(b) share some of the labels. This is because the same attribute propositions (and activity encodings) are used for both in order to allow calculating a cross-product where a single activity can result in a synchronous state change of multiple models (e.g., `A` activating the LMP-Declare constraint, while at the same time progressing the DPN).

4.3 Cross-Product Construction

After constructing the individual DFA of each model, we compute the cross-product of these DFAs using a standard automata cross-product algorithm. We do not minimize the cross-product as that would limit the possibilities of generating negative traces, i.e., traces that violate some input models.

The DFAs constructed from LMP-Declare models are used as-is. However, the DFAs constructed from DPNs are modified further. First, self loops are added to each state of the DFA such that the state of the DFA would not change if events from other models occur. Second, a non-accepting trap state is added such that this state is immediately entered whenever an event of that DPN would occur without the corresponding transition being enabled to fire.[1]

---

[1] Note that, for simplicity, we do not show these states in the examples provided throughout the paper.

4.4 Coloring and Cost Annotations

After computing the cross-product, we annotate it with three types of information. First, we follow the approach of colored automata [11,19] to label each cross-product state with one of four truth values for each input model, respectively indicating whether the corresponding model is *temporarily satisfied* (TS), *temporarily violated* (TV), *permanently satisfied* (PS), or *permanently violated* (PV). Second, we assume that each input model has a violation cost, which contributes to a stopping cost ($cost\_curr$) attached to each state of the cross-product (i.e., the cost of stopping the execution in the given state). Third, we calculate for each state, what is the best reachable stopping cost from that state ($cost\_best$) using the fixpoint procedure outlined in [4].

A fragment of the annotated cross-product is shown in Fig. 3. That fragment is based on the examples shown in Fig. 2. More specifically, it shows how the constraint in Fig. 2(a) affects the execution of the DPN shown in Fig. 2(b), while other states have been omitted.

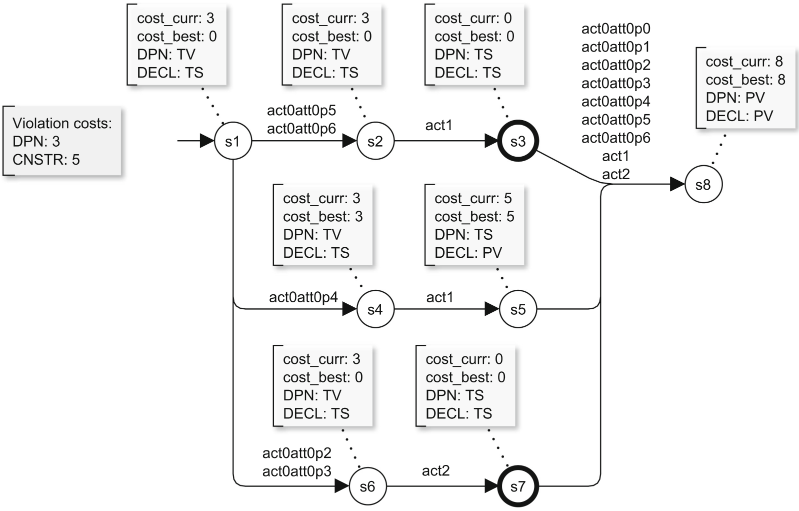
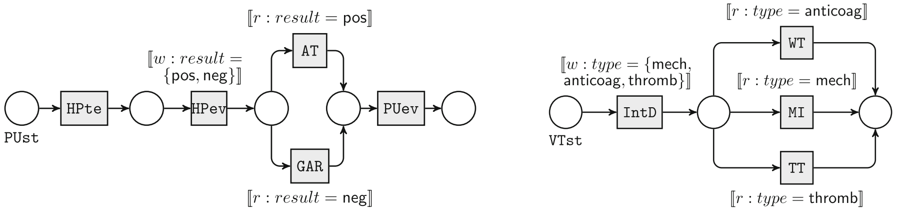


Fig. 3. Partial example of an annotated cross-product.

4.5 Multi-model Simulation

The cross-product above can be used to run multi-model simulations quite easily. The main idea is to perform semi-random walks, guided by the value of $cost\_best$ and translating the labels of the taken transitions back to concrete events.

For example, if the goal is to generate traces that satisfy all input process models, then, from $s1$ in Fig. 3, we can either take the transition to $s2$ or the transition to $s6$. Next, from state $s2$, we would need to take the transition to $s3$,



(a) DPN of peptic ulcer (PU) treatment.

(b) DPN of venous thromboembolism (VT) treatment.

Fig. 4. DPN models used in the performance experiments. Prefixes *r:* and *w:* distinguish read and write guards respectively.

while from $s6$ we would need to take the transition to $s7$. States $s3$ and $s7$ are accepting states in the cross-product, meaning that all sequences of transitions leading to these states correspond to traces that satisfy all input models.

We also need to translate the labels of the taken transitions back to concrete events. In the case of the transition from $s1$ to $s2$, both labels refer to activity $act0$, meaning that the event must be `A`. Both labels also contain the attribute identifier $att0$, meaning that the data payload of event `A` must contain attribute x. The propositions for $att0$ on that transition are $p5$ and $p6$, meaning that the value of attribute x must be 8 or greater, thus producing, for example, event `A[x=9]`. Applying the same translation to the other transitions, we can generate, for example, traces `<A[x=9], B>` and `<A[x=5], C>`, where the assignment of x is randomly selected from the interval (or possibly disjoint intervals) defined by the corresponding propositions.

Note that assigning x to a value in the range of (5, 8) in the above example would lead to either violating the DPN from Fig. 2(b) or the constraint from Fig. 2(a). Therefore, if we wanted to generate a trace that violates the constraint, but satisfies the DPN, we would first need to take the transition from $s1$ to $s4$ and then the transition from $s4$ to $s5$, which is possible due to not minimizing the cross-product. Generally speaking, it is possible to use the state annotations shown in Fig. 3 to generate negative traces with specific characteristics. In particular, the labels indicating the truth value of each input model (i.e., the colors) in a certain state can be used to understand which model(s) will be violated by a trace ending in that state. Furthermore, $cost\_curr$ and $cost\_best$ can be used to guide the generation of negative traces that violate the input HBPR with a certain violation cost.

5 Performance Experiments

In this section, we present the results of the performance experiments we conducted using our approach. All experiments are based on the two DPNs shown in Fig. 4. More specifically, we start with the models shown in the figure, and then iteratively increase both models by creating copies of them with renamed

Table 2. Experimental results. Columns $|P|$ and $|T|$ refer to the number of places and transitions in the corresponding input DPNs.

PU DPN		VT DPN		No. of Constraints	Cross-product time (s)	Log generation time (s)		
$\lvert P\rvert$	$\lvert T\rvert$	$\lvert P\rvert$	$\lvert T\rvert$			$n = 100$	$n = 1000$	$n = 10000$
5	5	3	4	1	0.047	0.230	1.394	11.453
9	10	5	8	2	0.180	0.356	2.185	19.004
13	15	7	12	3	0.906	0.449	3.073	27.628
17	20	9	16	4	3.440	0.562	4.040	36.713
21	25	11	20	5	11.837	0.652	4.612	45.016
25	30	13	24	6	45.955	0.724	5.413	55.460

activities and appending these copies to the original models via sequential composition. Furthermore, each pair of `AT` and `WT` is connected via a `not co-existence` constraint (without data conditions).

For each increase of the input models, we generated 100, 1000, and 10000 traces[2] with a 50:50 split between traces that satisfy all the input models and traces that violate at least one of them. The specific violations for each trace were determined randomly. All experiments were performed on a 6-core Intel i7 10850H machine with 2×16 GB of RAM and all input models are publicly available at: https://github.com/antialman/model-interplay-loggen-code.

The results of the experiments are shown in Table 2. For each set of input models, we report separately the cross-product construction time and the time needed for generating the event logs. Note that the cross-product needs to be rebuilt only if the input models are modified. In this case, we construct the cross-product once for each set of input models, and then reuse it for generating three event logs of different sizes. In general, the time (and the memory requirements) taken to create the cross-product increases rapidly as the size of the input models increases. However, models of realistic, though relatively small, sizes can be used to generate large logs (10000 traces) in a reasonable amount of time (55.4 s).

The log generation times for different log sizes (n) follow a roughly linear trend, which was expected as each trace is generated via a semi-random walk of the same cross-product. There is also a noticeable increase of the log generation time as the size of input models increases (from 11.4 s to 55.4 s for generating 10000 traces). This can be partially caused by the overall size of the cross-product being larger. Another plausible explanation for this is related to the fact that the way we extend the models results in longer traces being generated after each extension of the models (i.e., we need longer semi-random walks of the cross-product for generating traces that satisfy all input models).

Finally, we note that the main limiting factor of our approach, from the performance perspective, is memory consumption. The last test reported in Table 2 was also the largest test we could run with 32 GB of available memory. This

---

[2] Note that the length of the traces is constrained by the input models.

indicates that more complex models will require investigating alternatives to constructing the full cross-product, as discussed for example in [24].

6 Related Work

There are a series of approaches that use computational logic to generate log traces from declarative process models. In [9], the authors propose a tool for the synthetic generation of positive and negative event logs based on abductive logic programming for MP-Declare (an extension of Declare that allows for expressing conditions also on time and data) models. The approach in [10] relies, instead, on Answer-Set Programming for generating logs, again, from MP-Declare models. Finally, [18] proposes a log generation approach from MP-Declare models that translates the latter into Alloy programs and uses the model generation module of the Alloy Analyzer to generate traces satisfying such models. Whereas all such approaches handle trace generation from a data-aware fragment of Declare, they do not consider procedural components and neither do account for violation costs assigned to negative traces. The approach discussed in [13] relies on the traditional representation of Declare models as DFAs that, in turn, are used as string generators. Conceptually, this approach is similar to the one discussed in this paper to the point that it relies on automata-based techniques. However, it does not go beyond standard Declare specifications.

Other works propose methods for generating logs from procedural process models. One of the recent approaches in this area [8] proposes a framework for generating event logs via guided simulation of given process models. In particular, the framework is able to generate logs by taking into account specific process mining purposes such as process discovery and conformance checking. However, currently, the framework supports only procedural models (represented as BPMN or Petri nets). Similarly, the approach in [6] proposes a simulation-based log generator offering a comprehensive set of parameters for fine-tuning log generation scenarios. Finally, the CPN Tools-based approach proposed in [17] is able to generate logs from DPN models. None of the aforementioned procedural approaches has been extended to support log generation from HBPRs.

7 Conclusion

In this paper, we presented an approach to generate event logs according to the combined behavior of multiple declarative and procedural models. In its current state, we see it primarily as a tool for supporting controlled experiments in process mining. However, this instrument could also be used as a part of what-if process mining analysis pipelines where various what-if scenarios are assessed through running simulations. This use case requires additional advancements though, such as more fine grained definitions of process interplay, accounting for resource pools, and supporting durative activities. We plan to investigate the above use case as future work. Other avenues for future work include exploring memory efficient approaches for log generation and a more extensive evaluation of the approach using real-life process models.

Acknowledgements. The work of A. Alman was supported by the Estonian Research Council grant PRG1226. F.M. Maggi was supported by the UNIBZ project CAT. M. Montali was supported by the UNIBZ project ADAPTERS and the PRIN MIUR project PINPOINT Prot. 2020FNEB27.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, 2nd edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative workflows: balancing between flexibility and support. Comput. Sci. Res. Dev. **23**(2), 99–113 (2009)
3. Alman, A., Maggi, F.M., Montali, M., Patrizi, F., Rivkin, A.: Multi-model monitoring framework for hybrid process specifications. In: Franch, X., Poels, G., Gailly, F., Snoeck, M. (eds.) CAiSE 2022. LNCS, vol. 13295, pp. 319–335. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-07472-1_19
4. Alman, A., Maggi, F.M., Montali, M., Patrizi, F., Rivkin, A.: Monitoring hybrid process specifications with conflict management: an automata-theoretic approach. Artif. Intell. Med. **139**, 102512 (2023)
5. Andaloussi, A.A., Burattin, A., Slaats, T., Kindler, E., Weber, B.: On the declarative paradigm in hybrid business process representations: a conceptual framework and a systematic literature study. Inf. Syst. **91**, 101505 (2020)
6. Burattin, A.: PLG2: multiperspective process randomization with online and offline simulations. In: Azevedo, L., Cabanillas, C. (eds.) Proceedings of BPM Demo Track 2016. CEUR Workshop Proceedings, vol. 1789, pp. 1–6. CEUR-WS.org (2016)
7. Burattin, A., Maggi, F.M., Sperduti, A.: Conformance checking based on multi-perspective declarative process models. Expert Syst. Appl. **65**, 194–211 (2016)
8. Burattin, A., Re, B., Rossi, L., Tiezzi, F.: A purpose-guided log generation framework. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds.) BPM 2022. LNCS, vol. 13420, pp. 181–198. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16103-2_14
9. Chesani, F., et al.: Towards the generation of the "perfect" log using abductive logic programming. In: CILC. CEUR Workshop Proceedings, vol. 2396, pp. 179–192. CEUR-WS.org (2019)
10. Chiariello, F., Maggi, F.M., Patrizi, F.: ASP-based declarative process mining. In: AAAI, pp. 5539–5547. AAAI Press (2022)
11. De Giacomo, G., De Masellis, R., Grasso, M., Maggi, F.M., Montali, M.: Monitoring business metaconstraints based on LTL and LDL for finite traces. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) BPM 2014. LNCS, vol. 8659, pp. 1–17. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10172-9_1
12. de Leoni, M., Felli, P., Montali, M.: A holistic approach for soundness verification of decision-aware process models. In: Trujillo, J.C., et al. (eds.) ER 2018. LNCS, vol. 11157, pp. 219–235. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_17
13. Di Ciccio, C., Bernardi, M.L., Cimitile, M., Maggi, F.M.: Generating event logs through the simulation of declare models. In: Barjis, J., Pergl, R., Babkin, E. (eds.) EOMAS 2015. LNBIP, vol. 231, pp. 20–36. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24626-0_2

14. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. J. Data Semant. **4**(1), 29–57 (2015)
15. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, 2nd edn. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-56509-4
16. Gunther, C.W., Verbeek, H.M.W.: XES - standard definition, BPM reports, vol. 1409. BPMcenter.org (2014)
17. Li, G., de Carvalho, R.M., van der Aalst, W.M.P.: A model-based framework to automatically generate semi-real data for evaluating data analysis techniques. In: Filipe, J., Smialek, M., Brodsky, A., Hammoudi, S. (eds.) Proceedings of ICEIS 2019, pp. 213–220. SciTePress (2019)
18. Maggi, F.M., Marrella, A., Patrizi, F., Skydanienko, V.: Data-aware declarative process mining with SAT. ACM Trans. Intell. Syst. Technol. (2023)
19. Maggi, F.M., Montali, M., Westergaard, M., van der Aalst, W.M.P.: Monitoring business constraints with linear temporal logic: an approach based on colored automata. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 132–147. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23059-2_13
20. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Balanced multi-perspective checking of process conformance. Computing **98**(4), 407–437 (2016)
21. Montali, M., Pesic, M., van der Aalst, W.M.P., Chesani, F., Mello, P., Storari, S.: Declarative specification and verification of service choreographiess. ACM Trans. Web **4**(1), 3:1–3:62 (2010)
22. Piovesan, L., Terenziani, P., Dupré, D.T.: Temporal conformance analysis and explanation on comorbid patients. In: HEALTHINF, pp. 17–26. SciTePress (2018)
23. Westergaard, M.: Better algorithms for analyzing and enacting declarative workflow languages using LTL. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 83–98. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23059-2_10
24. Westergaard, M., Slaats, T.: Mixing paradigms for more comprehensible models. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 283–290. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_24

Exploring Hybrid Modelling of Industrial Process – Mining Use Case

Edyta Brzychczy(✉), Krzysztof Kluza, and Katarzyna Gdowska

AGH University of Krakow, Al. Mickiewicza 30, 30-059 Krakow, Poland
`{brzych3,kluza,kgdowska}@agh.edu.pl`

Abstract. Modelling industrial processes, especially in heavy industries, becomes a challenging task. The main aspects that influence modelling are related to the availability of suitable event logs and process variability, impacting the selection of modelling language and discovery algorithms. Our paper presents the results of modelling a real-life industrial process, namely longwall shearer operation. We started with motivation resulting from previous findings based on a comparison of imperative and declarative modelling of the process in question. It led us to the creation of a hybrid model and conformance checking to evaluate its ability to express sophisticated traces in the process execution. Our results show that the hybrid model reveals fewer deviations than analysed pure paradigm models (Petri nets and Declare).

Keywords: process modelling · hybrid models · process mining · mining industry · longwall shearer

1 Introduction

Industrial processes, especially in heavy branches, such as mining, oil and gas, steel production or chemical industries, are very specific due to the involvement of highly specialised machinery and uncertainty of process execution results.

In the mining and oil and gas industry, the variability related to the process is mainly a result of geology and mining conditions; in chemical or steel production industries, the variability is related to chemical and physical reactions that are part of the processes being carried out. Process variability can lead to many variants of process execution, making process modelling and analysis a difficult and challenging task.

The first challenge of modelling real-life industrial processes is related to the availability of event log data from process execution. Very often, data from highly specialised equipment are low-level grained in the form of sensor data (of binary and continuous type) requiring necessary preprocessing before the application of process mining algorithms [3].

The second challenge is related to processes' complexity in real-life scenarios, very often varying from theoretical or prescribed control flow, which impacts the selection of modelling language and discovery algorithms.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 302–313, 2024.
https://doi.org/10.1007/978-3-031-50974-2_23

As our case study, we present an example of an industrial mining process, namely the longwall shearer operation process.

Since process mining (PM) on sensor data is the wide and still open research direction with various techniques possible to apply (in our case, we used a supervised approach), we focus on the second challenge in this work. In this scope, we formulated the following research question:

(RQ) *How does the adoption of a hybrid model affect the modelling and analysis of complex traces in the longwall shearer operation compared to the use of pure paradigm models?*

The motivation for such RQ formulation arose from the results of work [8] in which authors created imperative models (in the form of Petri net) and declarative models (using Declare language) for the theoretical longwall shearer cycle. The obtained results showed the advantages of the declarative approach in modelling longwall shearer operation; however, interestingly, in some cases, the declarative model revealed many more deviations than Petri net. This finding led us to investigate hybrid models and the evaluation of a hybrid approach in the modelling and analysis of the process in question.

Our paper is structured as follows: in Sect. 2, we present a brief characteristic of the longwall shearer operation process. In Sect. 3, we present the main assumptions of our methodology. In Sect. 4, we present created hybrid model with its analysis based on real life event log. Section 5 contains related works. The final section covers the conclusion, limitations, and plans for the future.

2 Longwall Shearer Operation Process

Longwall mining is one of the primary mining technologies used in underground coal mines. In Poland, it is the dominant one. The main places in the underground mines where coal exploitation occurs are longwall panels, consisting of a maingate (headgate), a tailgate, and a longwall face. Each longwall face is equipped with machines and devices enabling the cutting of coal seams, transporting raw materials (coal and rock) and protecting excavation during mining activities.

The modern longwall complex includes a longwall shearer, an armoured face conveyor (AFC), mechanised roof supports, and a chain conveyor with a crusher (beam stage loader – BSL). The shearer is used for cutting and initial crushing of the coal, which falls onto AFC, which plays a double role, firstly as a route for the shearer and transport medium for coal, passing raw material to BSL. Safety and execution of cutting and transportation processes are ensured by mechanised roof supports, which move in the longwall face in the wake of a shearer passage.

Since the longwall shearer is the main machinery in the longwall face and the rest devices are adjusted to its activities, we focus on the longwall shearer operation process as a use case for our research.

The shearer operation can be seen as a cyclical implementation of a set of operations (activities) repeated in a specific order and time in a longwall face (Fig. 1).

The shearer starts the first part of the cycle by cutting from the beginning of the longwall face (maingate) in the along direction. After a 30-m distance, the shearer returns to the beginning and repeats the passage in the along direction, crossing the

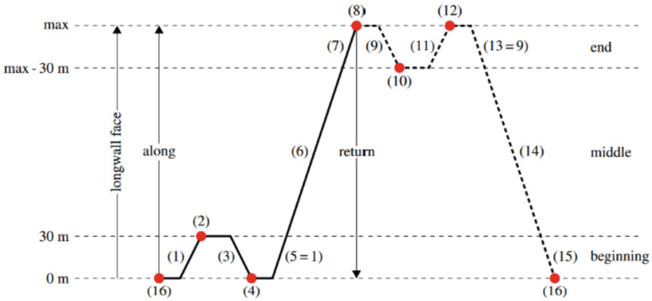


Fig. 1. Shearer operation cycle in the longwall face [15]

middle section of the longwall face and reaching the end of the longwall where the second part of the cycle starts. Thus, in the shearer cycle, the following stages can be defined [15]:

- the direction *along*: 1) Cutting beginning, 2) Stoppage beginning, 3) Return to drive, 4) Stoppage beginning, 5) Cutting beginning, 6) Cutting middle, 7) Cutting end, and 8) Stoppage end;
- the direction *return*: 9) Cutting end, 10) Stoppage end, 11) Return to drive, 12) Stoppage end, 13) Cutting end, 14) Cutting middle, 15) Cutting beginning, and 16) Stoppage beginning.

In real-life process execution, two more stages (not included in the theoretical model) can be observed: *Moving* (riding without cutting) and *Reversion* (cutting in the opposite direction than expected) in various parts of the longwall face.

Looking at the theoretical model of the longwall shearer operation process in the form of the Petri net (Fig. 2) obtained in *ProM* [18] with *Heuristic Miner*, one can state that it is relatively simple. However, in real-life execution, there are no two of the same shearer cycles (mainly due to changeable mining conditions and human operator factor), proving that the process, in reality, is very variable and complex simultaneously.

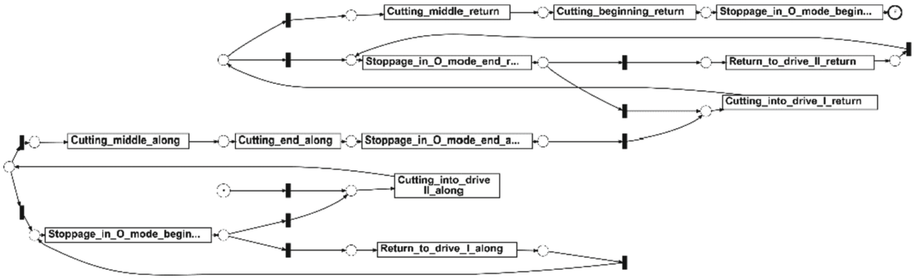

Cutting_middle_return
Cutting_beginning_return
Stoppage_in_O_mode_begin...
Stoppage_in_O_mode_end_r...
Return_to_drive_II_return
Cutting_into_drive_I_return
Cutting_middle_along
Cutting_end_along
Stoppage_in_O_mode_end_a...
Cutting_into_drive II_along
Stoppage_in_O_mode_begin...
Return_to_drive_I_along

Fig. 2. Petri net of longwall shearer operation built with Heuristic Miner.

Since the declarative approach in some cases is more restrictive than Petri net [8], we aim to analyse whether a hybrid approach with partly relaxed and restrictive parts will better deal with modelling the longwall shearer operation process. The procedure in our study is presented in the next section.

3 Procedure

The general scheme of the adopted procedure is presented in Fig. 3.

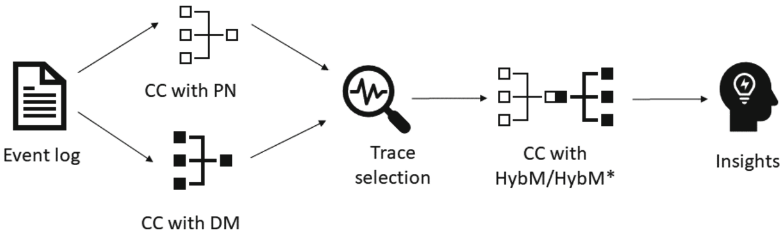


Fig. 3. Adopted procedure in our study

In our procedure, we used a real-life event log comprising 183 cycles of longwall shearer operation and the following (theoretical) process models: Petri net (PN) presented in Fig. 2, Declare model (DM) adopted from [8], expressed by constraints presented in Table 1, and hybrid models HybM and HybM*, presented in Fig. 6 and Table 1 respectively.

Providing the answer to the formulated RQ, we selected the conformance checking (CC) task to compare the number of deviations revealed by each investigated process model for event log traces. We defined deviations for PN and HybM as a sum of moves in log only and moves in the model only and for DM and HybM* as a sum of constraints' violations. We conducted CC with PN and HybM in *ProM* [16] and with DM and HybM* in *RuM* [1] software.

In the first step, we aimed to select the trace in the event log, which is characterised by the most significant difference in the number of deviations revealed by CC conducted on PN and DM. After trace selection and in-depth analysis of results, we used the selected trace in CC with a hybrid model created in an automatic manner (based on ideal process execution).

The hybrid model (HybM) used in our investigation is based on dependency nets, convertible to Petri nets and Declare. We used the implementation of the algorithm for mixed-paradigm modelling known as *Fusion Miner*, described in [9].

After CC on HybM (based on the algorithm described in [16]) and the results evaluation, we also introduced HybM*, a declarative equivalent of HybM, to conduct a more in-depth comparative analysis between PN, DM and HybM models. Based on the obtained results, we formulated insights related to the hybrid modelling of the longwall shearer operation process.

4 Results and Discussion

Our event log contains 183 variants of the process per 183 cases. The longest trace contains 172 events, while the shortest 47 events respectively.

Firstly, we compared the event log with the PN and DM. The results of conformance checking in the form of deviations' distributions are presented in Fig. 4. For the analysed event log, PN revealed 17 610 deviations and DM 14 420, respectively.

It can be noticed that conformance checking with DM for traces in the event log generally released fewer deviations than PN. However, there are some specific cases for which PN indicates the same or lesser number of deviations than DM (in total 30 cases).

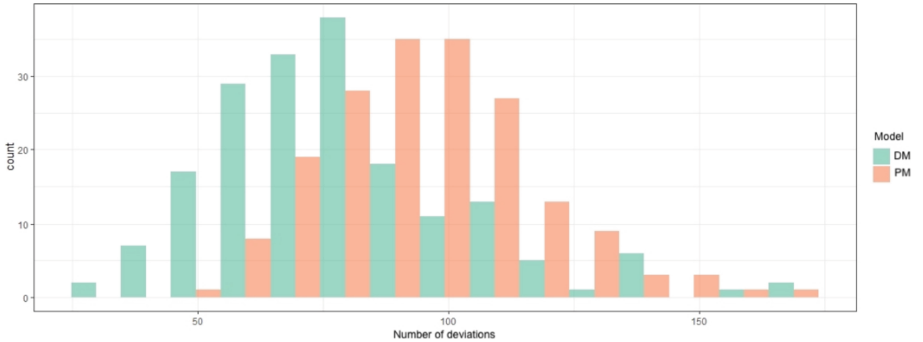


Fig. 4. Distributions of deviations (conformance checking results on PN and DM)

In Fig. 5, we present trace no. 187, for which PN reveals fewer deviations than DM (the difference equals 30). What we observed in the trace in comparison to the ideal cycle execution (Fig. 1) was that some apparent deviations occurred at the beginning of the longwall and middle of the longwall in the along direction as well as the middle of the longwall in the return direction.

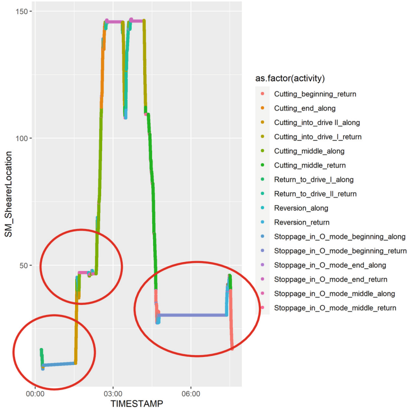


Fig. 5. Trace no 187 visualisation

Looking at the results of conformance checking on DM (Table 1), the most problematic constraints (given with bolded values) for this trace were: Chain

Table 1. Results of conformance checking trace no. 187 on PN, DM and HybM*

PN	DM	Fulfil-ments	Viola-tions	HybM*	Fulfil-ments	Viola-tions	Diff
0	absence: [Moving]	0	0		0	0	0
6	absence: [Reversion_along]	0	6		0	6	0
10	absence: [Reversion_return]	0	**10**		0	10	0
9	absence: [Stoppage_in_O_mode_middle_along]	0	9		0	9	0
14	absence: [Stoppage_in_O_mode_middle_return]	0	**14**		0	14	0
7	chain succession: [Cutting_beginning_return], [Stoppage_in_O_mode_beginning_return]	6	7		6	7	0
4	chain succession: [Cutting_end_along], [Stoppage_in_O_mode_end_along]	6	2		6	2	0
12	chain succession: [Cutting_middle_along], [Cutting_end_along]	2	**10**		2	10	0
15	chain succession: [Cutting_middle_return], [Cutting_beginning_return]	4	**15**		4	15	0
5	exactly1: [Cutting_beginning_return]	0	6		0	6	0
3	exactly1: [Cutting_end_along]	0	4		0	4	0
7	exactly1: [Cutting_middle_along]	0	8		0	8	0
12	exactly1: [Cutting_middle_return]	0	**13**		0	13	0
1	existence: [Return_to_drive_II_return]	8	0	exactly1: []	0	8	8
2	existence: [Return_to_drive_I_along]	3	0	exactly1: []	0	3	3
3	exactly1: [Stoppage_in_O_mode_end_along]	0	4		0	4	0
6	exactly1: [Stoppage_in_O_mode_beginning_return]	0	7		0	7	0
0	existence: [Stoppage_in_O_mode_beginning_along]	4	0		0	0	0
0	existence: [Cutting_into_drive II_along]	3	0		3	0	0
3	existence: [Cutting_into_drive_I_return]	6	0		6	0	0
0	existence: [Stoppage_in_O_mode_end_return]	9	0		9	0	0
1	init: [Cutting_into_drive II_along]	0	1		0	1	0
-	None	0	0	precedence: [Stoppage_in_O_mode_beginning_along], [Cutting_into_drive II_along]	3	0	0
3*	chain precedence: [Stoppage_in_O_mode_beginning_along], [Return_to_drive_I_along]	1	2	precedence: [,]	2	1	-1

(*continued*)

Table 1. (*continued*)

-	None	0	0	precedence: [Stoppage_in_O_mode_end_return], [Cutting_into_drive_I_return]	5	1	1
13**	chain precedence: [Stoppage_in_O_mode_end_return], [Return_to_drive_II_return]	6	2	precedence: [,]	8	0	-2
9**	chain response: [Return_to_drive_II_return], [Stoppage_in_O_mode_end_return]	6	2	response: [,]	8	0	-2
1*	chain response: [Return_to_drive_I_along], [Stoppage_in_O_mode_beginning_along]	3	0	response: [,]	3	0	0
-	None	0	0	response: [Stoppage_in_O_mode_beginning_along], [Cutting_into_drive II_along]	4	0	0
-	None	0	0	response: [Stoppage_in_O_mode_end_return], [Cutting_into_drive_I_return]	9	0	0
7	chain precedence: [Cutting_into_drive II_along], [Cutting_middle_along]	2	6	succession: [,]	2	0	-6
15	chain precedence: [Cutting_into_drive_I_return], [Cutting_middle_return]	2	11	succession: [,]	4	0	-11
-	None	0	0	succession: [Stoppage_in_O_mode_beginning_along], [Cutting_middle_along]	12	0	0
10	chain response: [Stoppage_in_O_mode_end_along], [Cutting_into_drive_I_return]	0	4	succession: [,]	10	0	-4
-	None	0	0	succession: [Stoppage_in_O_mode_end_along], [Stoppage_in_O_mode_end_return]	13	0	0
	None	0	0	succession: [Stoppage_in_O_mode_end_return], [Cutting_middle_return]	22	0	0
113[u]	Total:	71	**143**		141	**129**	-14

Remarks: (u) PN deviations also include 31 *moves in the model only* (1) numbers in grey denote how often activities were not in the given order on PN and are not taken into account since they are included in other moves in the log marked to *Existence*, *Exactly* or *Absence* constraints. (3) number of * denotes the same order of analysed activities on PN (4) in the HybM* column; only changes in constraints are given (5) *Diff* is calculated as the difference between violations revealed by DM and HybM*.

succession: [*Cutting_middle_return*], [*Cutting_beginning_return*], Absence: [*Stoppage_in_O_mode_middle_return*], Exactly1: [*Cutting_middle_return*], Chain precedence: [*Cutting_into_drive_I_return*], [*Cutting_middle_return*], Chain succession: [*Cutting_middle_along*], [*Cutting_end_along*], and Absence: [*Reversion_return*].

In most cases, comparison to PN results of conformance checking also revealed moves in the log in similar places of the process as DM, especially regarding behaviour modelled with *Chain* type dependencies. In PN occur also *moves in the model only* related to the construction of PN (i.e., silent transitions used to synchronise parts of the model, in this case, their number equals 31). In both models, the number of deviations excluding *Chain* type dependencies is the same and equal to 82. Thus, considering the obtained results, we decided to check whether it is possible to discover a hybrid model that better describes the complex reality of process execution without losing essential information about deviations occurring in the process.

Therefore, based on the event log containing ideal traces of process execution, we created a hybrid model (HybM) with the *FusionMINERful* plug-in [9] in *ProM* v.6.12 (Fig. 6).

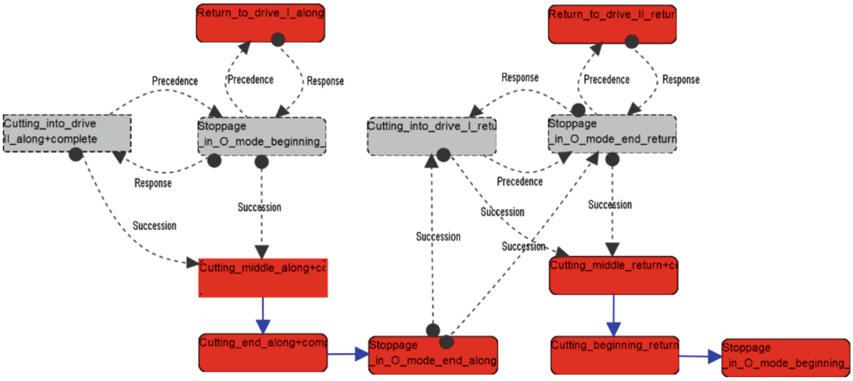


Fig. 6. Hybrid model of longwall shearer operation process (all nodes connected, prune and enhancement options included)

In the model, we can observe blue arcs representing procedural behaviour in the model and dashed arcs describing Declare constraints. Activities filled with grey denote *Existence* constraints and with red colour *Exactly1* constraints.

The conformance checking results of the analysed trace (with added artificial start and end events) using *Replay a Log on Mixed Paradigm model for Conformance Analysis* plug-in (ProM 6.7) [16] are presented in Fig. 7.

The total number of *moves in the log only* equals 84, and 1 *move on the model only*. Thus, it gave us a better total result than conformance checking on PN (113); however, in the number of *moves in the log only*, HybM has a worse performance (2 more moves). In the most problematic place in the model (marked in red), there are 54 *moves in the log only*.

Considering the results presented in Table 1, HybM in total performs better than created DM; however, to conduct a more detailed comparative analysis with PN and

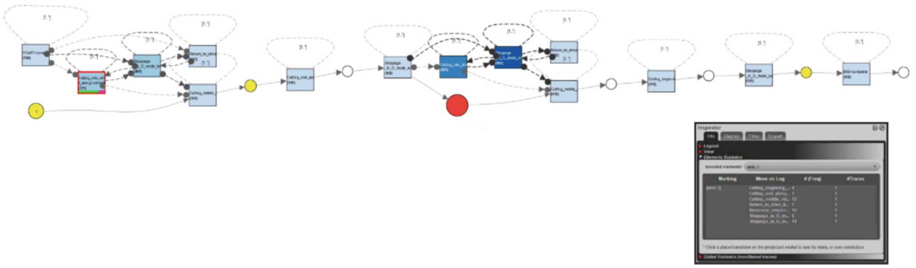


Fig. 7. Conformance checking results on the hybrid model

DM, we created a declarative equivalent of HybM (denoted as HybM*) and conducted conformance checking on HybM* in the *RuM* software.

Results of conformance checking on HybM* expressed by Declare constraints confirmed its better performance than DM for analysed trace (Table 1). However, some defined constraints are too general and do not express desired process behaviour, e.g., relaxed *Chain Precedence*, *Chain Response* or *Succession* types, which reduce the number of violations. On the other hand, it discovered *Exactly1* type constraints increasing the number of deviations.

To answer the formulated RQ, we conducted the conformance checking with HybM and HybM* for the whole event log (183 traces). HybM revealed 9 039 deviations (5 381 less than DM and 8 571 less than PN), and HybM* revealed 14 291 deviations (129 less than DM and 3 319 less than PN). Looking at the constraints fulfilment, HybM* presents a much higher number (42022) than DM (18470), which proves that the hybrid model has clear potential to describe better the reality of process execution recorded in the event log (especially when it is created in an automatic manner) than used DM.

Non-hybrid models often signal too many exceptions or errors, which can prevent the identification of key aspects in a process. On the other hand, a hybrid model typically flags fewer discrepancies, allowing for focused attention on these specific issues, thereby enabling more effective management and control over the process. Despite its complexity, selecting and refining a hybrid model provides a unique opportunity to encapsulate the nuances of real-world processes.

5 Related Work

Nowadays, process mining in the mining industry is still not fully explored. The main reasons are a lack of process mining awareness in the mining community and suitable event logs for process modelling and analysis.

Only a few publications exist in the literature related to PM implementation in this branch of industry. One of the first works [5] investigated the possibilities of PM usage in underground mining to model and analyse mechanised roof support in the coal mine with PN. In the other publications, authors investigated event log creation from sensor data sourced by the longwall monitoring system for PM purposes for the longwall shearer operation process [6, 7]. Modelling and analysis examples of other machines from the mining domain with PM, e.g., roof bolter, can be found in [4]. In this case, Directly

Follows Graphs have been applied. The mentioned examples were based on low-level data and required robust data preprocessing due to event log creation (with supervised [4, 5] or unsupervised techniques [6, 7]), making PM tasks possible to perform.

Looking at higher-level process modelling in the mining industry with PM, one can find [10] focused on the emergency rescue process of fatal gas explosion accidents in China. In the paper, process model discovery, as well as other PM tasks, were presented.

The most recent paper [17] presents the maintenance process optimisation of LHD machines in copper mine with the block-caving extraction method. The authors first used a Discrete Event Simulation to generate an event log based on available maintenance notices and order data. Process modelling and analysis were performed based on Directly Follows Graphs.

All mentioned applications relied on procedural models (PN) or process maps (DFG); only one paper [8] investigated the declarative approach for PM in the mining domain.

The choice between imperative and declarative modelling styles is not always straightforward. Imperative models, such as BPMN, are widely embraced in business process modelling due to their clear notations and capability to describe the exact behaviour of business processes in detail. They are extensively integrated into modelling tools [13].

However, in recent years, declarative models have gained significant attention, shifting the emphasis to the constraints that ought to be respected during the execution of a process. In a declarative model, behaviour that does not conform to the specified constraints is prohibited, while anything not explicitly excluded is allowed [9]. Declarative process modelling and mining techniques, including Declare Miner or Rule Mining, have become popular for creating flexible models based on event logs [1, 11].

Currently, one can observe a growing preference for hybrid process modelling notations that combine elements from both styles [2, 13, 14].

Practitioners involved in studies [12] have expressed openness to a hybrid approach that integrates both imperative and declarative techniques instead of completely transitioning from the imperative paradigm to the declarative one.

6 Conclusions, Limitations and Future Research

In this paper, we investigated whether hybrid models (combining procedural and declarative aspects) can better represent complex real-life industrial processes. Our primary focus was on the longwall shearer operation process in the mining industry in Poland. This process exhibits high variability and requires considerable sensor data processing before process mining can be applied.

Based on our findings, hybrid models demonstrated enhanced capability to handle the longwall shearer process variability, showcasing fewer deviations than purely procedural or declarative models. Despite a promising result, some limitations should be taken into account, e.g., conformance checking results exhibited instances where the hybrid model had worse performance in terms of moves in log only than PN but better than a purely DM. It is also worth noticing that the hybrid model expresses a more relaxed vision of desired process behaviour; thus, its implementation in practice for process analysis requires expert verification beforehand.

Although our research was based on a single real-life industrial case study that might not represent the full range of complexity in industrial processes, we hope it will serve as a stepping stone towards more nuanced and accurate process modelling in heavy industries and inspire further investigation into hybrid process modelling techniques.

The performed analysis shows that the creation of comparable process models describing reality using different paradigms is a challenging task; however, comparison of various paradigms' models could be helpful for domain experts to verify the desired process execution.

In our future work, we seek to expand the scope of the research by exploring other hybrid modelling techniques and algorithms that could yield more robust and reliable process models, as well as achieve better conformance checking results.

Acknowledgements. The authors thank Dr H. M. W. Eric Verbeek for supporting *ProM* plug-ins compilations.

References

1. Alman, A., Di Ciccio, C., Maggi, F., Montali, M., van der Aa, H.: RuM: declarative process mining, distilled. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 23–29. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_3
2. Andaloussi, A.A., Burattin, A., Slaats, T., Kindler, E., Weber, B.: On the declarative paradigm in hybrid business process representations: a conceptual framework and a systematic literature study. Inf. Syst. **91**, 101505 (2020)
3. Bertrand, Y., De Weerdt, J., Serral, E.: Assessing the suitability of traditional event log standards for IoT-enhanced event logs. In: Cabanillas, C., Garmann-Johnsen, N.F., Koschmider, A. (eds.) BPM 2022. LNBIP, vol. 460, pp. 63–75. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-25383-6_6
4. Brzychczy, E., Gackowiec, P., Liebetrau, M.: Data analytic approaches for mining process improvement—machinery utilisation use case. Resources **9**(2), 17 (2020)
5. Brzychczy, E., Trzcionkowska, A.: New possibilities for process analysis in an underground mine. Zeszyty Naukowe. Organ. Zarządzanie/Politech. Śląska **111**, 13–25 (2017)
6. Brzychczy, E., Trzcionkowska, A.: Creation of an event log from a low-level machinery monitoring system for process mining purposes. In: Yin, H., Camacho, D., Novais, P., Tallón-Ballesteros, A. (eds.) IDEAL 2018, Part II. LNCS, vol. 11315, pp. 54–63. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03496-2_7
7. Brzychczy, E., Trzcionkowska, A.: Process-oriented approach for analysis of sensor data from longwall monitoring system. In: Burduk, A., Chlebus, E., Nowakowski, T., Tubis, A. (eds.) ISPEM 2018. AISC, vol. 835, pp. 611–621. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-97490-3_58
8. Brzychczy, E., Szpyrka, M., Korski, J., Nalepa, G.J.: Imperative vs. declarative modeling of industrial process. The case study of the longwall shearer operation. IEEE Access **11**, 54495 (2023)
9. De Smedt, J., De Weerdt, J., Vanthienen, J.: Fusion miner: Process discovery for mixed-paradigm models. Decis. Support Syst. **77**, 123–136 (2015)
10. He, Z., Wu, Q., Wen, L., Fu, G.: A process mining approach to improve emergency rescue processes of fatal gas explosion accidents in Chinese coal mines. Saf. Sci. **111**, 154–166 (2019)

11. Maggi, F.M., Bose, R.P.J.C., van der Aalst, W.M.P.: Efficient discovery of understandable declarative process models from event logs. In: Ralyté, J., Franch, X., Brinkkemper, S., Wrycza, S. (eds.) CAiSE 2012. LNCS, vol. 7328, pp. 270–285. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31095-9_18
12. Reijers, H.A., Slaats, T., Stahl, C.: Declarative modeling–an academic dream or the future for BPM? In: Daniel, F., Wang, J., Weber, B. (eds.) Business Process Management. LNCS, vol. 8094, pp. 307–322. Springer, Cham (2013). https://doi.org/10.1007/978-3-642-40176-3_26
13. Slaats, T., Schunselaar, D.M.M., Maggi, F.M., Reijers, H.A.: The semantics of hybrid process models. In: Debruyne, C., et al. (eds.) OTM 2016. LNCS, vol. 10033, pp. 531–551. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48472-3_32
14. Slaats, T.: Declarative and hybrid process discovery: recent advances and open challenges. J. Data Semant. **9**(1), 3–20 (2020)
15. Szpyrka, M., Brzychczy, E., Napieraj, A., Korski, J., Nalepa, G.J.: Conformance checking of a longwall shearer operation based on low-level events. Energies **13**(24), 6630 (2020)
16. Van Dongen, B.F., De Smedt, J., Di Ciccio, C., Mendling, J.: Conformance checking of mixed-paradigm process models. Inf. Syst. **102**, 101685 (2021)
17. Velasquez, N., Anani, A., Munoz-Gama, J., Pascual, R.: Towards the application of process mining in the mining industry—an LHD maintenance process optimization case study. Sustainability **15**(10), 7974 (2023)
18. Verbeek, H.M.W., Buijs, J.C.A.M., Van Dongen, B.F., Van Der Aalst, W.M.P.: ProM 6: the process mining toolkit. In: CEUR Workshop Proceedings, vol. 615, pp. 34–39 (2010)

Pareto-Optimal Trace Generation from Declarative Process Models

Juan F. Diaz[1], Hugo A. López[2], Luis Quesada[3], and Juan C. Rosero[1](✉)

[1] Universidad del Valle, Cali, Colombia
{juanfco.diaz,juan.camilo.rosero}@correounivalle.edu.co
[2] Technical University of Denmark, Kongens Lyngby, Denmark
hulo@dtu.dk
[3] Insight Centre for Data Analytics, Cork, Ireland
luis.quesada@insight-centre.org

Abstract. Declarative process models (DPMs) enable the description of business process models with a high level of flexibility by being able to describe the constraints that compliant traces must abide by. In this way, a well-formed declarative specification generates a family of compliant traces. However, little is known about the difference between different compliant traces, as the only criterion used for comparison is satisfiability. In particular, we believe that not all compliant traces are alike: some might be sub-optimal in their resource usage. In this work, we would like to support users of DPMs in the selection of *compliant and optimal* traces. In particular, we use Dynamic Condition Response (DCR) graphs as our language to represent DPMs, extending it with a parametric definition of costs linked to events. Multiple types of cost imply that different traces might be optimal, each according to a different cost dimension. We encode cost-effective finite trace generation as a Constraint Optimisation Problem (COP) and showcase the feasibility of the implementation via an implementation in MiniZinc. Our initial benchmarks suggest that the implementation is capable of providing answers efficiently for processes of varying size, number of constraints, and trace length.

Keywords: Declarative Process Models · DCR graphs · Constraint Optimization Problems · Multi-Objective Optimization

1 Introduction

Starting from the seminal works of Pesic [1], declarative process models have been described as an alternative to provide flexibility in the orchestration of business processes. By describing a process in terms of its constraints, all non-compliant executions can be filtered (see Fig. 1, left-hand side). This flexibility allows knowledge workers to take discretionary decisions based on the nature of the case, knowing that any path they take will be compliant with the requirements of the process, for instance, laws or medical guidelines.

J. F. Diaz, H. A. López, L. Quesada and J. C. Rosero—Alphabetical order, equal author contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 314–325, 2024.
https://doi.org/10.1007/978-3-031-50974-2_24

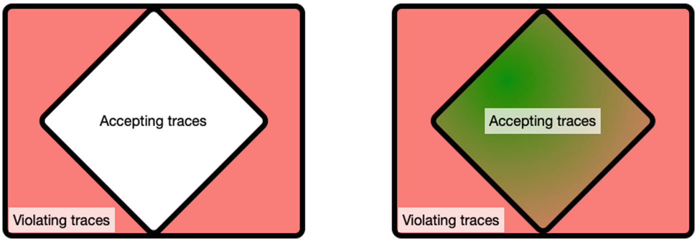


Fig. 1. Optimization of DPMs: constraints define the frame where valid executions live (left graphic), but even within the conforming space, some executions are more desirable than others (where optimality is denoted by color range in the right-most graphic).

In our previous work [2], we showcased different application cases where discerning between compliant executions is necessary. Our inspiration comes from helping consultants and case workers plan executions using declarative process models. Consider a patient-handling process as an example. A dentist needs to decide whether to apply preventive, interception, or corrective treatments depending on the impact on the patient, as well as other criteria (for instance, the associated cost). Not all the compliant treatments will be the same: some will be more cost-effective than others (depicted as the green regions in Fig. 1). Moreover, there will be cases in which different notions of costs play against each other. Consider a green-transition legislation process, where users are recommended to change to low-polluting technologies. Shifting from a gasoline car to an electric car will reduce the CO_2 expenses in the commuting process, but it will increase the economic cost for a family that already has a gasoline car.

Our Contribution. We introduce a method to identify the best traces in a declarative process model according to Pareto optimality. Our method is summarized in Fig. 2. In this case, we take a declarative process model (using the DCR graphs [3] modeling notation), extended with multiple cost annotations for each event in the model, and a maximum trace length. The algorithms presented in this paper encode the generation of cost-effective traces as a constraint optimization problem, so, in case the model has a solution within the maximum trace length that dominates one of the cost dimensions, it will be part of the output. For our optimization problem, we only consider finite traces. This means that even if the DCR models considered in the input can generate omega-regular executions, only finite traces will be considered. This restriction obeys practical user needs. In our interactions with industry users of DCR graphs in the public and private sectors, they are less interested in a universe of traces. Instead, they are interested in *value-driven* traces, those that finish a case eventually. Similar considerations have been considered for other declarative process languages [4]. Similarly, finite executions (finished cases) are an assumption commonly used when working in process discovery and conformance checking. To showcase the feasibility of our approach, we have implemented the COP in MiniZinc [5]. Our initial benchmarks suggest that the implementation can provide answers efficiently for processes of varying size, number of constraints, and trace length.

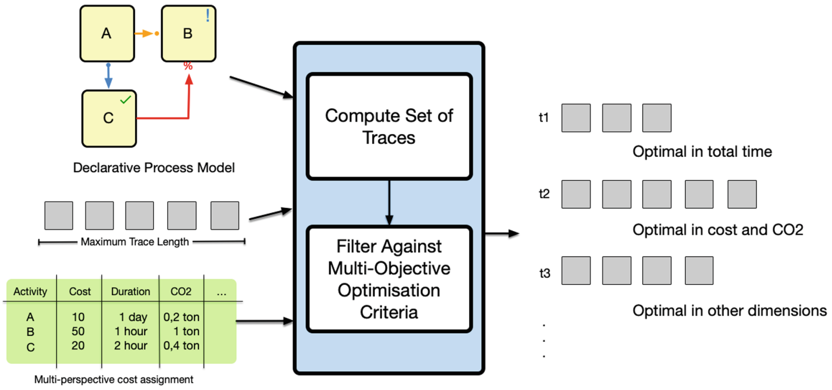

A
B
!
%
C
✓
Declarative Process Model
Maximum Trace Length
Activity | Cost | Duration | CO2 | ...
A | 10 | 1 day | 0,2 ton
B | 50 | 1 hour | 1 ton
C | 20 | 2 hour | 0,4 ton
Multi-perspective cost assignment
Compute Set of Traces
Filter Against Multi-Objective Optimisation Criteria
t1
Optimal in total time
t2
Optimal in cost and CO2
t3
Optimal in other dimensions

Fig. 2. Visual abstract of our approach. Our inputs (left-hand side) include a DCR graph, the maximum trace length, and a resource assignment. For our outputs (right-hand side) we generate a set of traces that are optimal in at least one cost dimension.

Paper Structure. We finish Sect. 1 presenting related work. In Sect. 2 we provide background to DCR graphs and multi-objective optimization. An extension of DCR graphs with multi-dimensional costs is introduced in Sect. 3. We solve the discovery of optimal traces according to costs as a combination of a Constraint Satisfaction Problem (CSP) and a Constraint Optimization Problem (COP) in Sect. 4. The benchmark of our solution is presented in Sect. 5, and we conclude in Sect. 6.

1.1 Related Work

Several approaches have been used to model multiobjective optimization in business processes. An evolutionary algorithm-based optimization framework is proposed in [6], employing the well-known Non-Dominated Sorting Genetic Algorithm II (NSGAII) [7], to generate optimized business processes by considering search space, fitness function, and optimization constraints. Evolutionary algorithms are also used in [8] to generate optimized business processes, based on predefined requirements, a task library, and input/output resources. The work in [9] enhances the approach in [8] with preprocessing phases. Djedovic et al. [10] use genetic algorithms to optimize BPMN 2.0 processes. Si et al. use genetic algorithms for the optimization of Colored Petri Nets, considering specifications for possible processes, resource types, selection, and elimination criteria [11]. Jiménez et al. introduces a framework for the automatic generation of business processes using the process modeling language SDeclare to produce optimal enactment plans by removing Pareto-dominated plans [12]. Burattin et al. propose a method to check trace conformance and compliance in business processes [13]. Additionally, López-Pintado et al. introduces an approach involving Pareto efficiency to optimize resource allocation [14]. Compared to these works, our contribution is novel in several key aspects. First, our primary focus lies in discovering optimal executions (traces) for activities that have fixed resource costs rather than allocating resources to activities for optimization purposes, as observed in [8–11, 14]. Second, we adopt a declarative approach using DCR graphs, whereas most works excepting [12, 13] work on imperative process models. Our work also diverges from [12] in the approach taken: while [12] involves assigning roles to activities and considering resource availability based on roles, our work focuses on assigning resource consumption to activities and optimizing the overall resource utilization of a process. Another aspect is that we are using a constraint-based model to generate traces and to specify the requirements needed for a trace to be valid and compliant with a given DCR graph, which differs from the conformance checking approach used in [13], which essentially looks at the compliance and informativeness of traces against a model. Finally, our pruning method to accelerate reaching the Pareto frontier is different than the ranking method used by [14].

2 Background

We will recall the definitions of DCR graphs and multi-objective process optimization. Our work is framed on the classical definitions by Hildebrandt and Mukkamala [3].

2.1 DCR Graphs

Definition 1 (DCR graphs [3]). *A Dynamic Condition Response Graph (DCR Graph) is a tuple* $G = \langle E, M, Act, \rightarrow\bullet, \bullet\rightarrow, \pm, l\rangle$*, where 1.* E *is the set of events. 2.* $M \in \mathcal{M}(G) = \mathcal{P}(E) \times \mathcal{P}(E) \times \mathcal{P}(E)$ *is a marking and* $\mathcal{M}(G)$ *is the set of all markings. 3.* Act *is the set of actions. 4.* $\rightarrow\bullet \subseteq E \times E$ *is the condition relation. 5.* $\bullet\rightarrow \subseteq E \times E$ *is the response relation. 5.* $\pm\colon E \times E \rightharpoonup \{+, \%\}$ *defines the dynamic inclusion/exclusion relations by* $e \rightarrow{+}e'$ *if* $\pm(e, e') = +$ *and* $e \rightarrow\%e'$ *if* $\pm(e, e') = \%$*. 6.* $l\colon E \rightarrow Act$ *is a labelling function.*

In DCR graphs, the condition and response relations allow for cyclic interactions. The marking $M = (Ex, Re, In) \in M(G)$ comprises three sets of events: executed events (Ex), pending responses (Re) that are yet to be executed or excluded, and currently included events (In). The dynamic exclusion and inclusion relations, denoted as $\rightarrow+$ and $\rightarrow\%$, are represented in the partial map $\pm\colon E \times E \rightharpoonup \{+, \%\}$. These relations enable events to be dynamically included or excluded from the graph. An event e *is enabled* in a marking $M = (Ex, Re, In)$ if 1) $e \in In$ and 2) if $\exists f.(f, e) \in \rightarrow\bullet \implies f \in Ex \vee f \notin In$. When evaluating constraints, only the events currently included are considered. For instance, if an event a has a response event b, but b is excluded, the occurrence of b is not necessary for the graph to be acceptable. Intuitively, the relation $e \rightarrow{+}e'$ indicates that when event e occurs, event e' is included in the graph. On the other hand, $e \rightarrow\%e'$ signifies that when event e occurs, event e' is excluded from the graph. Moreover, we will simplify our development by assuming l to be a bijective function [15], thus allowing us to compute its inverse.

The execution semantics of DCR graphs is defined in terms of a labelled transition system, where states are defined by markings, and transitions are fired events.

Definition 2 (Transitions [3]). *For a DCR graph* $G = \langle E, M, Act, \rightarrow\bullet, \bullet\rightarrow, \pm, l\rangle$*, the corresponding labelled transition system* $T(G)$ *to be the tuple* $\langle \mathcal{M}(G), M, \rightarrow \subseteq \mathcal{M}(G) \times Act \times \mathcal{M}(G)\rangle$ *where* $\mathcal{M}(G)$ *is the set of markings* G*,* $M \in \mathcal{M}(G)$ *is the initial marking, and* $\rightarrow \subseteq \mathcal{M}(G) \times (E \times Act) \times \mathcal{M}(G)$ *is the transition relation given* $M' \xrightarrow{(e,a)} M''$ *where: 1.* $M' = (Ex', Re', In')$ *is the marking before the transition. 2.* $M'' = (Ex' \cup \{e\}, Re'', In'')$ *is the marking after the transition. 3.* $e \in In'$ *and* $l(e) = a$ *4.* $\{e' \in In' \mid e' \rightarrow\bullet e\} \subseteq Ex'$ *5.* $In'' = (In' \cup \{e' \mid e \rightarrow{+}e'\}) \setminus \{e' \mid e \rightarrow\%e'\}$ *6.* $Re'' = (Re' \setminus \{e\}) \cup \{e' \mid e \bullet\rightarrow e'\}$

We define a trace $t = (e_0, a_0), (e_1, a_1), ...$ as a (possibly infinite) sequence of transitions $M_i \xrightarrow{(e_i,a_i)} M_{i+1}$ where $M_i = (Ex_i, Re_i, In_i)$ and $M_0 = M$. A trace is *accepting* if $\forall i \geq 0$, $e \in Re_i$, $\exists j \leq i$: $(e = e_j \vee e \notin In_j)$. This means that there is no event that is both included and pending at the same time, without having been executed first. Given an accepting trace $t = (e_0, a_0), (e_1, a_1), \ldots$, we also define $Actions(t) = \langle a_0, a_1, ...\rangle$.

The definition includes the following key points: (i) Markings before and after transitions, (ii) Execution requirement for an event e to be included and the condition that all currently included condition events for e must have been executed, (iii) Updates to the set of included events and pending responses during a transition.

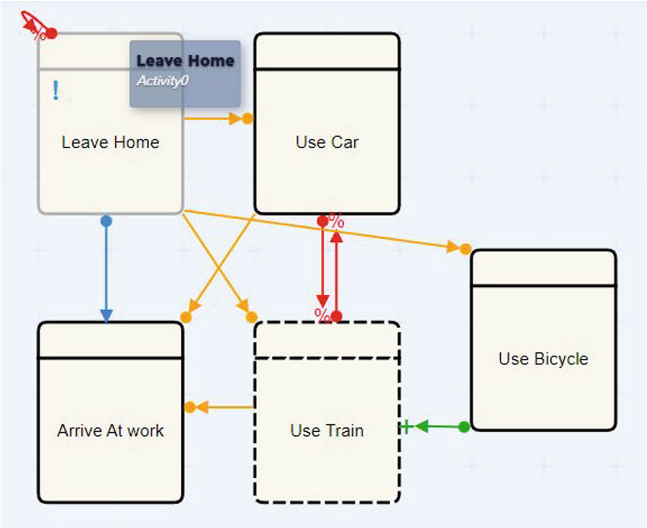


Fig. 3. A simple DCR graph showing the inclusion, exclusion, condition, and response relation

Figure 3 showcases a simple DCR graph representing the process of going to work. The graph consists of 5 events, each event has a label, e.g., $Activity0$ labeled as $Leave\,Home$. The markings indicate the status of each event, where some are included (solid borders), excluded (dashed borders), pending (decorated with a ! symbol), or executed (decorated with a ✓). The graph illustrates in the hovering label the event (Activity0) and the action (Leave Home) associated. The graph shows four types of relations: inclusions, exclusions, conditions, and responses. Executing certain events triggers changes, such as including or excluding other events. For example, executing $Use\,Bicycle$ includes $Use\,Train$ while executing $Use\,Car$ excludes it. Condition relations specify prerequisites: both $Use\,Car$ and $Use\,Train$ must be executed or excluded before $Arrive\,At\,Work$ can occur. The response relation highlights the requirement of executing or excluding $Arrive\,At\,Work$ when $Leave\,Home$ is executed. So for a trace to be valid, we must first start by leaving home ($Leave\,Home$), then we would either use the car ($Use\,Car$) and reach work ($Arrive\,At\,Work$) or use the bicycle ($Use\,Bicycle$) and then take a train ($Use\,Train$) to work ($Arrive\,At\,Work$).

2.2 Pareto Optimization

A multi-objective optimization problem requires the task of maximizing a series of $k > 1$ objective functions, usually in a conflict between them. Let S be a set of decision vectors $\vec{s_1}, \ldots, \vec{s_n}$. An objective vector is the projection of a decision vector consisting of the values of the objective functions $z = (f_1(\vec{s}), \ldots, f_n(\vec{s}))$. The optimization problem is defined as:

$Maximize\{f_1(\vec{s}), \ldots, f_n(\vec{s})\}$, such that $\vec{s} \in S, f_i \colon S \rightarrow \mathbb{R}, S \neq \emptyset$. According to [16], Pareto optimization searches for dominant decision vectors. For $\vec{s}, \vec{s'} \in S$ we say that $\vec{s} \in S$ dominates $\vec{s'}$ if:

$$\forall i \in 1, \ldots, k \colon f_i(\vec{s}) \geq f_i(\vec{s'}) \wedge \exists j \in 1, \ldots, k \colon f_j(\vec{s}) > f_j(\vec{s'})$$

Finally, we say that $\vec{s} \in S$ is a Pareto optimal vector if it does not exist $\vec{s'} \in S$ such that $\vec{s'}$ dominates $\vec{s}$. Pareto solutions can be understood as those balance points where it is impossible to better optimize one of the components of the decision vector without compromising the optimization of any other components.

3 DCR Graphs with Multi-dimensional Costs

In this section, we extend DCR graphs with notions of multi-dimensional costs and introduce the generation of cost-optimal traces as an optimization problem based on Pareto optimization.

Definition 3. *An Extended Dynamic Condition and Response Graph (EDCR graph) is a tuple* $EG = \langle G, \varPhi, \$, \oplus \rangle$*, where:*

1. *G is a DCR graph,*
2. *$\varPhi = \{\phi_1, \ldots, \phi_n\}$ is a finite set of features,*
3. *$\$ \colon \varPhi \times Act \rightarrow \mathbb{R}$. is the cost of the action associated with a feature, and*
4. *$\oplus \colon \varPhi \times (\mathbb{R} \times \mathbb{R}) \rightarrow \mathbb{R}$ is the aggregation function according to a feature.*

We assume that $\oplus$ is commutative, associative, and has an identity element, thus $\oplus(\phi, (r_1, r_2)) = \oplus(\phi, (r_2, r_1))$ and $\oplus(\phi, (r_3, \oplus(\phi, (r_2, r_1)))) = \oplus(\phi, (\oplus(\phi, (r_3, r_2), r_1)))$. Then we write $\oplus(\phi, (r_1, r_2, \ldots, r_j))$ to denote the aggregation of j results for feature ϕ. A parametric definition of aggregation allows us to have multiple ways of treating resources. For instance, some features will be aggregated with a simple sum, while others can be aggregated with a minimum/maximum among the resources.

Let EG be a $EDCR$ graph and $a \in Act$. Then we define $\alpha(\cdot) \colon Act \rightarrow \mathbb{R}^{|\varPhi|}$ to denote the multi-dimensional cost of executing an activity. For representing the cost of an action a in EG, we define $\alpha(a)$ as: $\alpha(a) = (\$(\phi_1, a), \ldots, \$(\phi_n, a))$.

Let $t = \langle (e_0, a_1), \ldots, (e_m, a_m) \rangle$ be a trace in EG. The cost of feature ϕ on t is denoted $\$(\phi, t)$ and it is defined as: $\$(\phi, t) = \oplus(\phi, \$(\phi, a_1), \ldots, \$(\phi, a_m))$ and the total cost of t is defined as the tuple: $\alpha(t) = (\$(\phi_1, t), \ldots, \$(\phi_n, t))$.

We can now define our optimization problem related to EDCR graphs.

Definition 4. *Given a $EDCR$ graph $EG = (G, \varPhi, \$, \oplus)$, and a maximum trace length k, the problem of finding optimal traces of a $EDCR$ graph (*EDCR-OPT *for short), is defined as follows:*

Input: $EG, k \in \mathbb{N}, (opt_\phi)_{\phi \in \varPhi}$

Output: $T = \{\langle (e_0, a_0), \ldots, (e_j, a_j) \rangle \mid j \leq k\}$ *such that $\alpha(t)$ is Pareto optimal for each $t \in T$ according to the optimization criteria $(opt_\phi)_{\phi \in \varPhi}$ ($opt_\phi \in \{min, max\}$).*

4 Implementing EDCR-OPT as a Constraint Optimization Problem

In this section, we present our solution to the problem specified in Definition 4. We first introduce a constraint-based model to generate a valid trace given a DCR graph. We then focus on the optimization by first presenting a Branch and Bound approach to compute an approximation of the Pareto optimal frontier and then describing a filtering process to filter out dominated solutions in the approximation.

The solution was implemented using: (i) Python version 3.9.5, (ii) MiniZinc version 2.6.2, (iii) Gecode Solver version 6.3.0, and (iv) Python-minzinc library version 0.9.0. Our implementation is available online[1].

4.1 Generating a Valid Trace

In this section we present a constraint-based model for generating a valid trace of an EDCR graph $EG = \langle G, \Phi, \$, \oplus \rangle$, where $G = \langle E, M, Act, \rightarrow\!\bullet, \bullet\!\rightarrow, \pm, l \rangle$, with a maximum trace length k and $T(G)$ is the transition system for G.

We formally introduce the model in terms of its inputs, outputs, and constraint as follows:
Inputs: EDCR graph $EG = \langle G, \Phi, \$, \oplus \rangle$, and a maximum length $k \in \mathbb{N}$.
Outputs: $t = \langle (e_0, a_0), \ldots, (e_j, a_j) \rangle$ such that $j \leq k$.
Constraints: $0 \leq i < j - 1$:

$$M_i \xrightarrow{(e,a)} M_{i+1} \implies Ex_{i+1} = Ex_i \cup \{e\} \tag{1}$$

$$M_i \xrightarrow{(e,a)} M_{i+1} \implies Re_{i+1} = (Re_i \backslash \{e\}) \cup \{f \mid (e,f) \in \bullet\!\rightarrow\} \tag{2}$$

$$M_i \xrightarrow{(e,a)} M_{i+1} \wedge (e,f) \notin \bullet\!\rightarrow \implies Re_{i+1} = Re_i \tag{3}$$

$$\forall (e,f) \in \rightarrow\!\% : M_i \xrightarrow{(e,a)} M_{i+1} \wedge (e,f) \notin \rightarrow\!+ \implies (In_{i+1} = In_i \setminus \{f\}) \tag{4}$$

$$\forall (e,f) \in \rightarrow\!+ : M_i \xrightarrow{(e,a)} M_{i+1} \implies In_{i+1} = In_i \cup \{f\} \tag{5}$$

$$M_i \xrightarrow{(e,a)} M_{i+1} \wedge (e,f) \notin \rightarrow\!+ \wedge (e,f) \notin \rightarrow\!\% \implies In_{i+1} = In_i \tag{6}$$

$$M_i \xrightarrow{(e,a)} M_{i+1} \wedge \exists (f,e) \in \rightarrow\!\bullet \implies (e \in In_i \wedge f \in Ex_i \vee f \notin In_i) \tag{7}$$

$$In_j \cap Re_j \cap Ex_j = \emptyset \tag{8}$$

Our constraint system is modelled based on the criteria for the LTS of DCR graphs (c.f. Definition 2). The intuition behind the equations of our constraint system is the following: Eq. 1 describes the effects imposed in the set of executed markings after a transition. Equations 2 and 3 define the pending markings after an event transition. The effects of transitions in the set of included events are described by Eqs. 4, 5, and 6. Equation 4 excludes the target event of an exclusion relation if the event is not included at the same time. Equation 5 handles the effects of executing an event source of an inclusion relation. Equation 6 preserve the included sets if no exclusions or inclusions have been affected. Equation 7 ensures that an executed event must be allowed, meaning it must be included at that moment, and all events that are conditions for its execution must have been executed previously or be excluded. Finally, Eq. 8 models the accepting state of the graph, indicating that for a trace to be accepted, there should not be any pending and included event that has not been executed.

---

[1] https://github.com/JuanK120/dcrGraph.

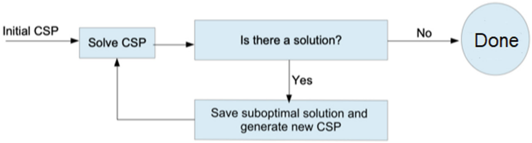


Fig. 4. An adaptation of Branch and Bound to compute a set containing the Pareto optimal frontier

4.2 Computing the Pareto Optimal Frontier

There are two phases in the computation of the frontier. We first compute a superset of traces that contains the frontier. In the second phase, we filter out the dominant solutions in the superset.

Computation of the Superset. Features, costs, and aggregations for an Extended DCR graph implementation are added to the model of Sect. 4.1 following the definitions in Sect. 3.

The process followed for the computation of the superset is presented in Fig. 4. We extend a basic constraint-based Branch and Bound approach [17] to save the solutions that are found during the execution of the process. During the iterative process, we add a constraint at each iteration to ensure that each new solution found is different, and better in at least one feature compared to the previously found solutions.

In the context of DCR graphs, we apply the Branch and Bound algorithm as follows: initially, we execute the DCR model to find a single solution. Once a possible trace, denoted as t, is found, we store this solution. Subsequently, a new CSP is created. In this CSP, we introduce an additional constraint to ensure that any subsequent solution, denoted as t', must be superior in at least one objective. This constraint is formulated as follows: $\exists(\phi \in \varPhi)(\alpha(t')\phi < \alpha(t)\phi)$. We repeat this process until it becomes infeasible to find better solutions. This approach effectively controls the size of the generated solution set by discarding solutions that are known to be sub-optimal and guarantees that Pareto optimal solutions are not overlooked.

Filtering the Superset. After applying the Branch and Bound algorithm and obtaining a set of potential solutions, the next step is to filter this set using the Pareto optimality concept. The Pareto optimal solutions represent the optimal trade-off solutions in a multi-objective optimization problem, where improving one objective comes at the expense of worsening another. By using a quadratic algorithm, we can efficiently evaluate and filter the potential solutions to identify the Pareto optimal solutions, which provide the best overall solutions considering all objectives simultaneously.

5 Scalability of the Solution

To evaluate the performance of our solutions, we developed a DCR graph generator that allowed us to vary multiple features of the graph. These features included trace length (k), the number of events, features, conditions, responses, inclusions, and exclusions. To assess the impact of each dimension on the tool's performance, we conducted separate tests for each dimension, fixing the other dimensions at a reasonable magnitude, and then gradually increasing the size of the dimension under examination. for example, for analyzing the number of events, we fixed all other dimensions at 10, as it would provide a variety of options for the model to choose from, and then we started at 15 events, then 17, then 19 and so on until reaching 45. Additionally, for each step of the analysis, we generated multiple graphs with the same dimensions, the total number of graphs generated was 3082. For each DCR graph, we collected three performance metrics: nodes, which is the number of explored nodes of the search tree of the constraint model done by the solver during the search for solutions, solveTime, which is the time spent by the MiniZinc motor running the constraint model provided by the python-minizinc tool, and totalTime, the overall solving time of the tool calculated by the python algorithm, in which we start the count at the beginning of the implemented tool until the moment just before the end of the process. We created box plots for each dimension to analyze the data, representing the measures obtained from all graphs with that specific dimension. These box plots visually represent the performance measures across different sizes. For instance, in Fig. 5, the plot compares one of the features of the graph (X-axis) to the time it took the algorithm to run (Y-axis). For example, in the first plot, with conditions, each boxplot displays the interquartile range (IQR) with a median line within the box. Whiskers extend from the box, indicating the range within a certain distance from the quartiles. Any data points beyond the whiskers are considered potential outliers, represented as diamonds in this case.

All DCR graphs, data files, and plots generated can be accessed in the code repository[2]. Figure 5 illustrates the impact of different variables on the performance measures. Conditions and responses exhibit the most significant influence, as evidenced by their larger values. This is expected since conditions and responses control the graph flow and complexity of solutions.

Regarding the variables k (trace length) and feats (features), their impact is less pronounced compared to conditions, responses, and events. The growth in k expands the search space, albeit not as significantly as the increase in the number of events, this is because even though k allows for longer traces it does not necessarily affects, the general behavior of a DCR graph, nor does it guarantees that there are traces of a length k, as depending of the specific DCR graph there could be the case where there are no traces beyond a lesser length than k. Feats contribute to the problem's complexity, but their effect is not as prominent as the other variables. This is because trace generation plays a larger role in the complexity of the problem, and it is logical, given that this is the combinatorial part of our Optimization problem, as trace generation involves choosing which events form a trace. Inclusions and exclusions have a minimal impact on performance measures. Although they may increase graph complexity, overall the performance depends more on specific constraints, restrictions, and requirements within the graph. Thus, compared to conditions and responses, inclusions and exclusions have a lesser effect on performance. This is because inclusions and exclusions do not significantly influence the flow of the DCR graph, inclusions depending on if there are any excluded events at any point might even not have any impact at all in the graph, as if an event being included is already included in the graph, then it is irrelevant. a similar case happens with exclusions, in general work to avoid certain combinations of events from happening, generally speaking, it doesn't go beyond that, as opposed to conditions and responses, which have a bigger impact, as they control, the requisites for events to be executed in the case of conditions, and also the requirements to reach an accepting state in case of responses.

Also, potential outliers can be noted due to the conformation of specific graphs, especially in variables that greatly impact the flow of a DCR graph, like conditions and responses, this is because depending on how a DCR graph is composed, a large number of events in a trace might be necessary to reach an accepting state, for example, if there are chains of relations in the generated graphs, for example, if a pending event has 2 other events as conditions, and those events have other events as conditions, these would make it so that any possible trace would take longer to execute, as the execution of multiple previous events would be required to reach an accepting state.

---

[2] https://github.com/JuanK120/dcrGraph/tree/master/Tests/Detailed.

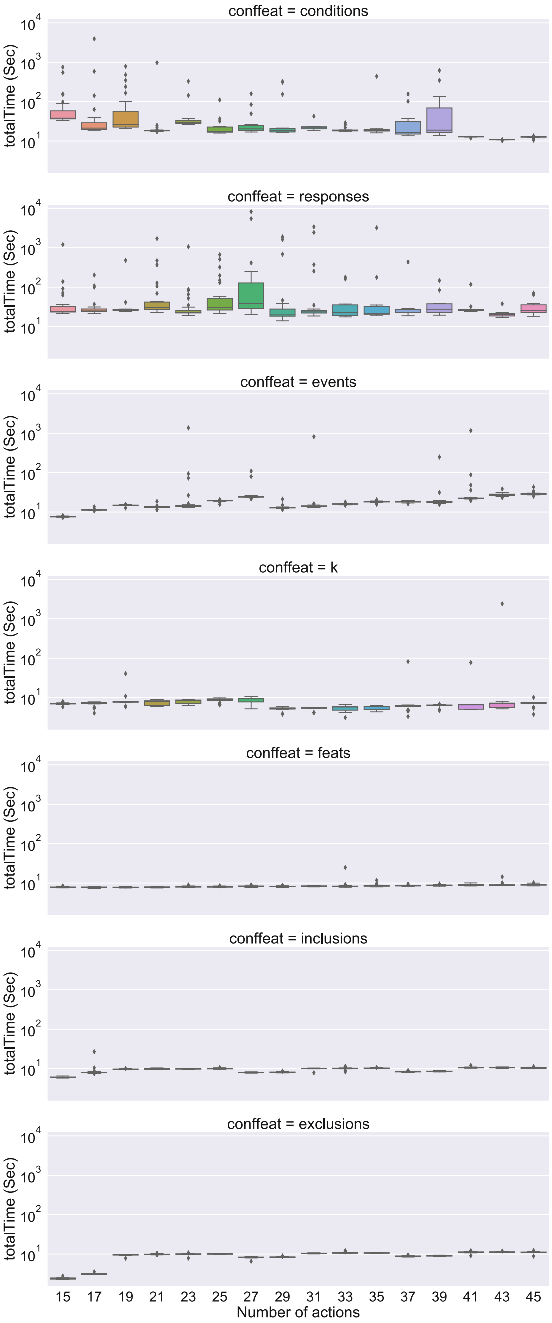


Fig. 5. Performance of the solution measuring the total time measure

6 Conclusions

In this work, we explored process optimization based on the model based on the proposed extension for DCR graphs. We introduced DCR graphs as a modeling technique and extended them to incorporate multi-objective optimization in business processes. To evaluate our implementation we conducted a performance analysis to assess the efficiency and effectiveness of our approach. By considering various factors such as trace length, events, conditions, responses, inclusions, and exclusions, we gained insights into the impact of these variables on time and complexity. This research contributes to the field of business process optimization by offering a novel application of Extended DCR graphs. Our findings highlight the importance of considering multi-objective optimization and provide a foundation for future research. Understanding how variables affect optimization solutions enables organizations to make informed decisions and enhance their competitiveness in dynamic markets.

Future work involves considering how to integrate this approach with partially executed traces, paving the way to the integration with streaming and predictive process mining approaches. Moreover, we would like to conduct a more complex empirical study to evaluate the simultaneous variation of multiple variables and explore real-world applications with diverse events, resources, and features. Additionally, our solution should be refined and integrated into a user-friendly tool capable of handling partial traces, providing businesses with tailored recommendations for process improvement.

References

1. Pesic, M.: Constraint-based workflow management systems: shifting control to users. Ph.D. Dissertation, Industrial Engineering and Innovation Sciences (2008)
2. Burattin, A., Gianola, A., López, H.A., Montali, M.: Exploring the conformance space (extended abstract). In: ITBPM@BPM, ser. CEUR Workshop Proceedings, vol. 2952, pp. 62–67. CEUR-WS.org (2021)
3. Hildebrandt, T.T., Mukkamala, R.R.: Declarative event-based workflow as distributed dynamic condition response graphs. arXiv preprint arXiv:1110.4161 (2011)
4. De Giacomo, G., De Masellis, R., Montali, M.: Reasoning on LTL on finite traces: insensitivity to infiniteness. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28, no. 1 (2014)
5. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: MiniZinc: towards a standard CP modelling language. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 529–543. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74970-7_38
6. Mahammed, N., Benslimane, S.M., Ouldkradda, A., Fahsi, M.: Evolutionary business process optimization using a multiple-criteria decision analysis method. In: International Conference on Computer, Information and Telecommunication Systems (CITS), Alsace, Colmar, France, pp. 1–5. IEEE (2018)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
8. Vergidis, K., Saxena, D., Tiwari, A.: An evolutionary multi-objective framework for business process optimisation. Appl. Soft Comput. **12**(8), 2638–2653 (2012)
9. Georgoulakos, K., Vergidis, K., Tsakalidis, G., Samaras, N.: Evolutionary multi-objective optimization of business process designs with pre-processing. In: IEEE Congress on Evolutionary Computation (CEC), Spain, pp. 897–904. IEEE (2017)

10. Djedovic, A., Zunic, E., Avdagic, Z., Karabegovic, A.: Optimization of business processes by automatic reallocation of resources using the genetic algorithm. In: XI International Symposium on Telecommunications (BIHTEL), Sarajevo, Bosnia and Herzegovina, pp. 1–7. IEEE (2016)
11. Si, Y.-W., Chan, V.-I., Dumas, M., Zhang, D.: A petri nets based generic genetic algorithm framework for resource optimization in business processes. Simul. Model. Pract. Theory **86**, 72–101 (2018)
12. Jiménez-Ramírez, A., Weber, B., Barba, I., Del Valle, C.: Generating optimized configurable business process models in scenarios subject to uncertainty. Inf. Softw. Technol. **57**, 571–594 (2015)
13. Burattin, A., Guizzardi, G., Maggi, F.M., Montali, M.: Fifty shades of green: how informative is a compliant process trace? In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 611–626. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_38
14. López-Pintado, O., Dumas, M., Yerokhin, M., Maggi, F.M.: Silhouetting the cost-time front: multi-objective resource optimization in business processes. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNBIP, vol. 427, pp. 92–108. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85440-9_6
15. López, H.A., Debois, S., Slaats, T., Hildebrandt, T.T.: Business process compliance using reference models of law. In: International Conference on Fundamental Approaches to Software Engineering, pp. 378–399 (2020)
16. Jakob, W., Blume, C.: Pareto optimization or cascaded weighted sum: a comparison of concepts. Algorithms **7**(1), 166–185 (2014)
17. van Beek, P.: Backtracking search algorithms. In: Rossi, F., van Beek, P., Walsh, T. (eds.) Handbook of Constraint Programming. Foundations of Artificial Intelligence, vol. 2, pp. 85–134. Elsevier (2006). https://doi.org/10.1016/S1574-6526(06)80008-8

Empirical Evidence of DMN Errors in the Wild - An SAP Signavio Case Study

Carl Corea[1,2](✉), Timotheus Kampik[2], and Patrick Delfmann[1]

[1] University of Koblenz, Koblenz, Germany
{ccorea,delfmann}@uni-koblenz.de
[2] SAP Signavio, Berlin, Germany
timotheus.kampik@sap.com

Abstract. While the Decision Model and Notation standard (DMN) is considered to be an increasingly popular standard, there is a broad consensus that human modelling errors can easily occur in the creation of DMN models. Yet, while this consensus is clear, there is only limited evidence of which error types exactly may occur in practice. In this work, we therefore present some empirical evidence on DMN errors in the wild. Specifically, we analyze the SAP-SAM dataset by SAP Signavio, containing over 500 000 real-world conceptual models. Our results show that modelling errors, such as missing rules, occur frequently in real-life settings (36.1% of all models contained some form of issue). Furthermore, we analyze the distribution of which error types have occurred (relative to an existing classification of DMN error types from a previous work). To the best of our knowledge, this is the largest DMN study conducted to date (N = 5 668 DMN models).

Keywords: Decision Model and Notation (DMN) · Modelling Errors · Case Study

1 Introduction

As a close counterpart to BPMN, the Decision Model and Notation (DMN) standard[1] is receiving increasing popularity for modelling organizational decision-logic. While DMN modelling can be conducted in an intuitive and graphical way, a wealth of recent research indicates that DMN modelling is prone to various types of human modelling errors, such as overlapping rules, or mismatches between DMN and BPMN [1–3].

Because the key business value that DMN provides is the direct translation of domain expert requirements to code, mitigating such errors by means of verification capabilities is crucial. Otherwise, if the correctness of the DMN model cannot be assured by the domain experts alone, the value of DMN as a bridge between business and IT does not materialize. While this consensus is clear, there is currently only

---

[1] https://www.omg.org/spec/DMN/About-DMN/.

C. Corea—Visiting Researcher.
This research was supported by SAP Signavio.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 326–336, 2024.
https://doi.org/10.1007/978-3-031-50974-2_25

limited empirical evidence for which types of errors actually occur in real-life settings (and in general, how many errors occur here). Such insights would however be needed for understanding requirements and developing better verification techniques for such settings. Therefore, in this work, we present the results of a case study conducted with SAP Signavio, where we analyzed a corpus of 5 668 DMN models.

The models have been extracted from SAP-SAM [4] – a dataset that contains hundreds of thousands of business models created for research, teaching, and learning purposes. Our results clearly show that the analyzed real-life models contain many modelling errors (36% of all models contained some form of issue), and that there are some specific error types that occur very frequently. In the following, we present the results of our study, classify the identified errors with respect to an existing classification of DMN error types from an own previous work [5], and study correlations between different error type occurrences. Finally, we also present key learnings from our case study and propose recommendations for future work.

The remainder is as follows. We present preliminaries on DMN errors in Sect. 2, and our study design in Sect. 3. Our results are presented in Sect. 4 and discussed in Sect. 5.

2 Preliminaries

DMN decision models are constituted of two different levels, namely a *decision requirements level* (which specifies the general elements of the decision making and their relations), and a *decision logic level* (which specifies the actual decision logic via decision tables). Then, the DMN model can be used to define company decision-logic in a table-like manner, where the resulting model can be used to infer decisions relative to instance-dependent input. Furthermore, it is possible to integrate DMN models with BPMN models, with the intuition that the decision-logic can be used to route the process through the BPMN model [6]. Please see [5] for a more detailled introduction.

As with BPMN, the actual contents of the DMN model are in the responsibility of the human modeller. Here, as suggested by recent research, there is a broad consensus that modelling errors can easily occur [1–3]. Figure 1 shows two examples of this: Left (a), a decision table containing overlapping rules is shown, i.e., rules 2 and 3 will fire at the same time for an *income* between [30..40] and *assets* > 100. Worse, the conclusions differ, meaning that no correct decision can be obtained in this case. Right (b), a BPMN model connected to a DMN model is shown (business rule task). As shown in [3,7], if not all outgoing branches of the BPMN XOR-gateway are included as outputs of the annotated decision table, this leads to unreachable branches for the process model. Vice versa, if a "platinum" customer is observed, the process cannot continue.

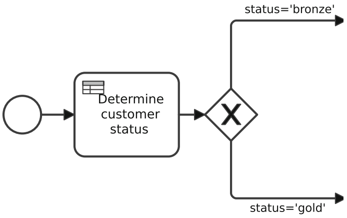
In an own previous work [5], an initial *classification* of DMN error types was distilled based on various error types found in the literature (specifically: [2,8,9]). The resulting classification is shown in Fig. 2. It consists of a total of 26 error types, and is divided into modelling errors on the level of the decision tables, the level of the decision requirements diagram, and the level of DMN+BPMN. Due to space limitations, we omit a discussion of the 26 individual error types and refer the reader to a supplementary document for detailed information[2].

---

[2] https://cloud.uni-koblenz-landau.de/s/yRbQWTiBtE9ZBFq.

	Income (n)	Assets (n)	Creditworthy?
1	≤ 20	>50	false
2	[20..50]	>50	false
3	[30..40]	>100	true

(a) Exemplary Decision Table (with overlapping rules 2 and 3)



(b) Exemplary Business Rule Task

Fig. 1. Two examples of DMN usage.

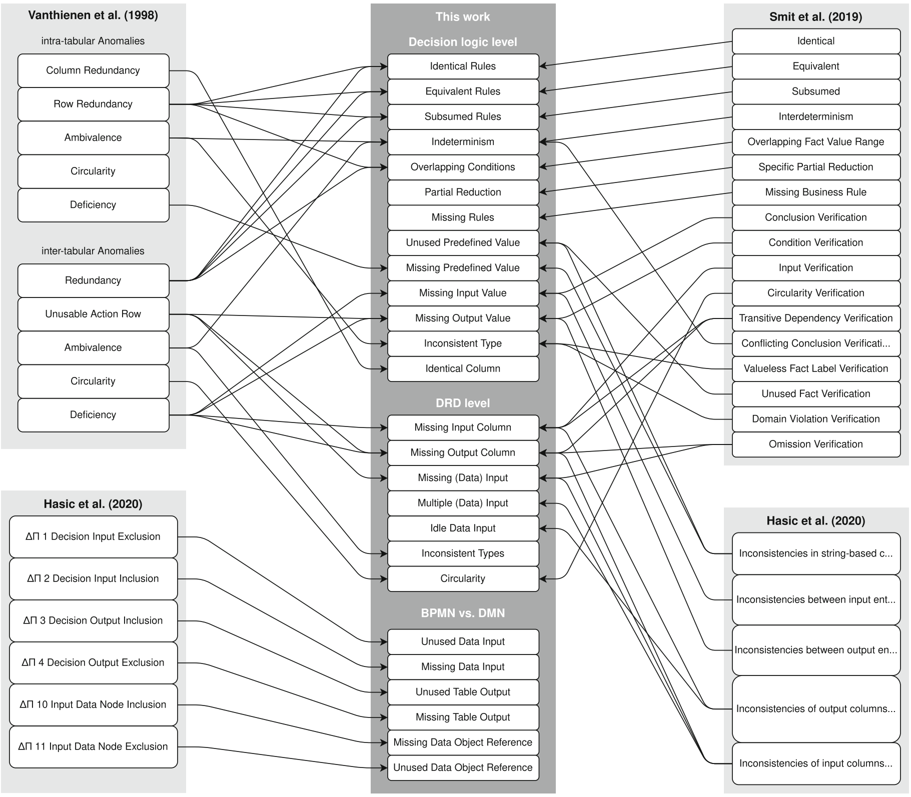


Fig. 2. Overview of the considered DMN error type classification, taken from [5].

The DMN models analyzed in this case study were all created in the software tool by (SAP) Signavio[3]. In regard to the DMN error types shown above, the Signavio modeller offers some support for finding certain types of errors (via a “verification” functional-

---

[3] https://www.signavio.com.

ity). Yet, it was shown in an own previous work that the Signavio modeller does not support all of the above error types [5]. An overview of the supported error types for the Signavio tool is shown in Table 1. Note also that the Signavio modeller supports many error types only for numerical values, and i.e., not for String values (see Table 1). In this regard, it may be expected that those error types that currently cannot be detected by the Signavio modeller will likely be contained in some of the analyzed models. We will get back to this aspect in the key learnings (Sect. 5).

Table 1. Verification capabilities supported by the Signavio modelling tool, taken from [5]. (x/o = full/partial support, n = only for numerical values.)

	Decision logic level													DRD level							BPMN/DMN					
	Identical Rules	Equivalent Rules	Subsumed Rules	Indeterminism	Overlapping Conditions	Partial Reduction	Missing Rules	Unused Predefined Value	Missing Predefined Value	Missing Input value	Missing Output value	Inconsistent Type	Identical Columns	Missing Input Column	Missing Output Column	Missing (Data) Input	Multiple (Data) Input	Idle Data Input	Inconsistent Types	Circularity	Unused Data Input	Missing Data Input	Unused Table Output	Missing Table Output	Missing Data Object Reference	Unused Data Object Reference
Signavio	X^n		X^n	X^n	X^n	X^n	X^n	X^n	X^n	o		X		X	o	X	X		X							

We now continue to present our study design and results.

3 Study Design

The aim of the study was to analyze DMN models for contained errors. The study was executed on SAP-SAM [4], a large and open dataset of process, decision, and other business models that have been created for research, teaching, and learning purposes by university researchers, teachers, and students over the course of roughly one decade (from 2011 to 2021)[4]. Importantly, while the DMN models were created by real users, they were mainly created for research and teaching purposes. Hence, one can assume that many of the models have been created by relatively inexperienced modelers, and that they do not represent "industry-scale" models. While this means the results may not reflect quality aspects of industrial datasets, it brings the advantage that our results offer novel insights regarding mistakes made by beginner to novice modellers. These insights can be used to improve modelling tools and onboarding processes. We will discuss this aspect of our study further in Sect. 5.

---

[4] The SAP-SAM is available at https://zenodo.org/record/6964945. Example code that gives a starting point for working with the dataset is available at https://github.com/signavio/sap-sam. Excluding auto-generated example models, the dataset contains around 550,000 models in various notations.

The considered dataset contains over 1 000 000 models of various notations. Some of these models are auto-generated models by Signavio (initiated for every workspace), so we excluded these for our study. The number of user-generated models remaining was around 550 000. From these models, we extraced all DMN models. There were some "empty" DMN models without any decision logic, which we excluded as well. This resulted in a total corpus of 5 668 DMN models for our study. Then, we aimed to analyze all models with respect to the error types presented in Fig. 2. For this, a previous tool which is able to identify most of the error types from Fig. 2, presented in [10,11], could be used. In turn, we could obtain results on which errors, resp. error types, were present in the analyzed models. For this study, we analyzed only DMN models (i.e., we did not verify any BPMN models which may have been linked to a DMN). In result, we only analyze error types pertaining to the decision table- and DRD level (and not the BPMN+DMN level). We aim to conduct such an analysis in future work; however, this will require further efforts in correctly aggregating the mapped models.

After having stored all DMN models locally, we ran a programm which queried all models against the mentioned verification API from [10]. The total runtime of our evaluation here was around 10 min for evaluating all models.

4 Results

Before we present the error verification results, we present some basic statistics of the analyzed dataset, as shown in Table 2.

Table 2. Some basic statistics of the analyzed dataset.

Description	Value	SD
# of analyzed models	5 668	–
Avg. # of decision tables per model	1.96	2.92
Avg. # of input data nodes per model	3.96	3.63
Avg. # of rules per model	10.66	16.30
Avg. # of rules per table (per model)	5.62	7.24
Avg. # of input columns per table (per model)	2.38	1.59

As can be seen, the DMN models contained on average only around 2 decision tables (the max. number of tables was 80), each with an average of around 2–3 input columns (the max. number of input columns was 15). Each DMN model contained on average roughly 10 rules (the max. number of rules was 324). To foreshadow our results, as we still found many errors in these models, this seemed very interesting to us, as it indicates that even for such seemingly small models, humans seem to have problems in modelling correct DMN. This could indicate that in fact, DMN may be a difficult notation to learn for beginners.

We now continue with our results. In detail, we found the following distribution of errors in the analyzed DMN models, shown in Fig. 3.

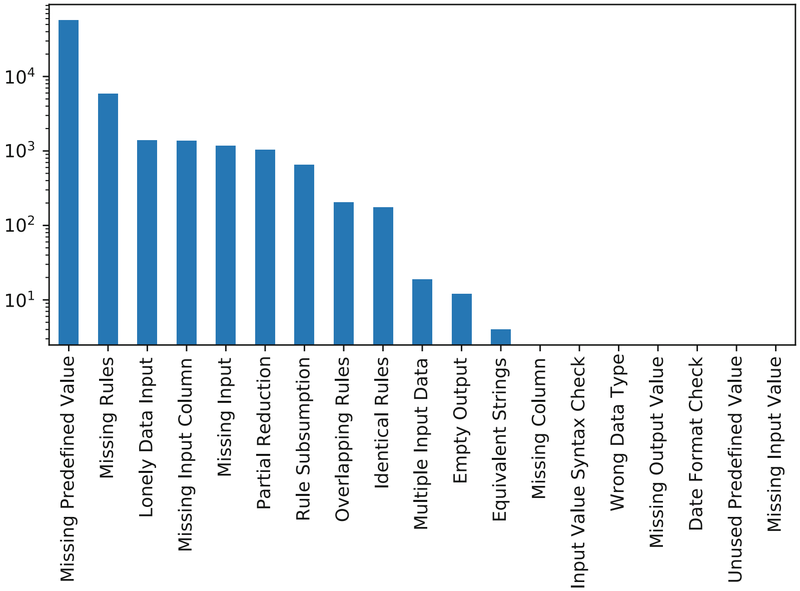
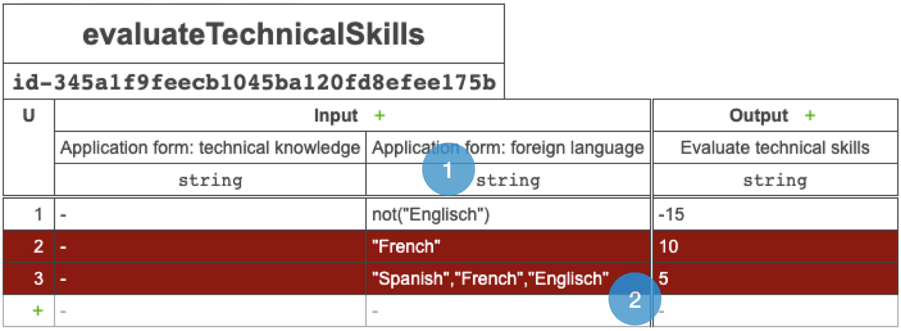


Fig. 3. Absolute occurrence of error types in the analyzed model collection.

As can be seen, the most prominent error type was that of *missing predefined values* (57 429). An example for an error of this type from the dataset is shown in Fig. 4 (1) (It refers to variables used in the rules that were not declared beforehand). Furthermore, there were many errors regarding certain problems in the decision logic, e.g., *subsumed rules* (658), or *missing rules* (5 861). An example of a subsumed rule error from the dataset is shown in Fig. 4 (2).


evaluateTechnicalSkills

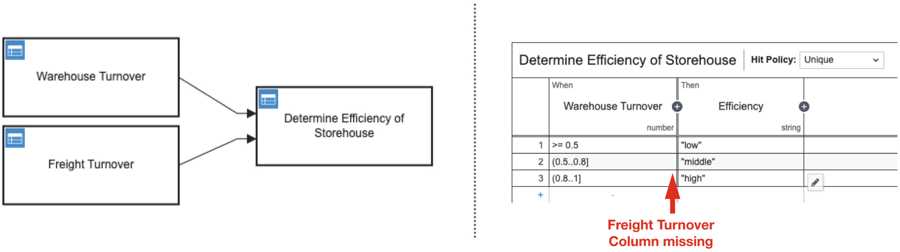
`id-345a1f9feecb1045ba120fd8efee175b`

U	Input +		Output +
	Application form: technical knowledge	Application form: foreign language	Evaluate technical skills
	string	string	string
1	-	not("Englisch")	-15
2	-	"French"	10
3	-	"Spanish","French","Englisch"	5
+	-	-	-

Fig. 4. Real error examples from the analyzed dataset: (1) No predefined item values (*English, French*, etc.), and (2) rule subsumption (rule 3 subsumes rule 2).

Interestingly, we found a total of 175 errors of type *identical rules*, which refers to finding multiple rules which are *exactly* the same (in a “String” sense), i.e., completely redundant. This was very surprising to us, as we would consider this an error that can theoretically be spotted easily, e.g., much more easily than spotting subsumed rules (where a modeller has to look exactly at the rule conditions). The fact that we found so many errors of type *identical rules* could indicate that even for smaller tables, it is cognitively hard for humans to spot two identical rules. In our opinion, this is again a motivation that human modellers need to be supported by powerful verification techniques, as it seems that even such simple errors can actually occur quite frequently.

There were also many errors on the DRD level. In particular, there were a substantial amount of *unused* data input nodes (1 393), or *missing* inputs (e.g., no data input node connected) (1 180). Also, there were many inter-table errors, e.g., an output column of one table that was not referenced by a following table (*missing input column*, 1 368). An example of this is shown in Fig. 5. This could indicate that even in such small models, it is cognitively hard for modellers to cross-check such aspects.


Warehouse Turnover
Freight Turnover
Determine Efficiency of Storehouse
Freight Turnover Column missing

Determine Efficiency of Storehouse | Hit Policy: Unique

	When Warehouse Turnover number	Then Efficiency string
1	>= 0.5	"low"
2	(0.5..0.8]	"middle"
3	(0.8..1]	"high"

Fig. 5. Real example for a *missing input column* from the dataset.

Figure 3 also includes the seven error types of which no instances were detected. In this regard, it is noteworthy that the modelling tool used for the creation of all models enforces certain quality aspects by preventing the user from saving the model as long as certain errors are present. For example, the Signavio modeller prevents the user from entering a value of a wrong data type, e.g., entering a String in an Integer column. And in fact, we could find no errors of this type (*wrong data type*). This indicates that some of these safeguards are working well, thus highlighting the potential value of implementing further safeguards. As a simple corollary, as some of the error types (such as *wrong data type*) are prevented on a software level—and thus these errors cannot occur in the database—this means that the distributions of similar models created with *other* software, e.g., without such safeguards, will likely look different. This would be an interesting aspect to investigate in future work, should such datasets become available.

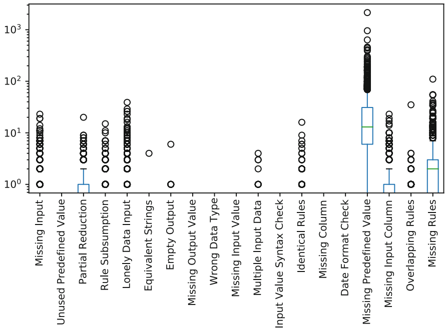
Summarizing, as a main result, 36.1% of all models contained some form of modelling issue. This clearly indicates the need for verification techniques.

As a clear limitation of this study, the used classification of DMN error types does not yet contain the dimension of a *severity* of individual error types. So in some settings, some of the “error types” we found may be seen as simple warnings, and the models could still be deployed (e.g., while identical rules are unwanted and can also be confusing the modellers, strictly speaking, they still allow for correct automated decision-making). In this sense, the value of 36.1% for the fraction of models containing some form of error should be taken with a grain of salt. To offer more granular insights, Fig. 6(a) shows which models contained at least one error with respect to all individual error types. Putting aside the mentioned problem of missing predefined values, we still see from the data that around 27% of all models contain missing rules, or some flaw in the decision logic. Furthermore, around 10% of all models contain an issue on the DRD level (e.g., missing input columns). We argue that these numbers can be seen as quite high, and should give motivation to carefully analyze DMN models designed by beginner to novice modellers.

Figure 6(b) shows the average number of errors per model (over all error types). Noteworthy, every model had on average 27 predefined values missing, and around 2–3 missing rules. Regarding the latter, given that every model contained on average a total number of 10 rules (cf. Table 2), this seems high, as it would indicate all models would require an increase of around 20–30% of rules in order to yield a sound decision logic. Also, note that on average, every model had a missing input on the DRD level, which could indicate that it is cognitively hard for human modellers to keep track of the synchronization between DRD level and the decision table. This aspect has already been mentioned in a recent work [9], and can clearly be seen in our data.

Error Type	**Models with at least one error of this type**
Missing Predefined Value	33.2%
Missing Rules	27.6%
Missing Input Column	10.6%
Partial Reduction	10.1%
Missing Input	8.9%
Rule Subsumption	7.4%
Lonely Data Input	6.0%
Overlapping Rules	2.3%
Identical Rules	1.6%
Multiple Input Data	<0.0% (11)
Empty Output	<0.0% (7)
Equivalent String	<0.0% (1)
All Others	-

(a) Error type occurrences



(b) Error type averages per model

Fig. 6. Error type occurrence and error type average per model.

In [12], the authors investigated general complexity metrics of DMN (e.g., number of tabels, number of rules, etc.) and already presented some insights on correlations – for example, an increase in the number of tables is associated with an increase in the number of rules. In regard to such complexity metrics, some correlations to DMN errors are linked inherently, e.g., an increase in the number of rules can intuitively increase the risk for overlapping rules. However, what is currently still unclear is whether there is a correlation between certain error types. We therefore computed the correlations of error types over all models using the standard Pearson correlation[5], shown in Fig. 7.

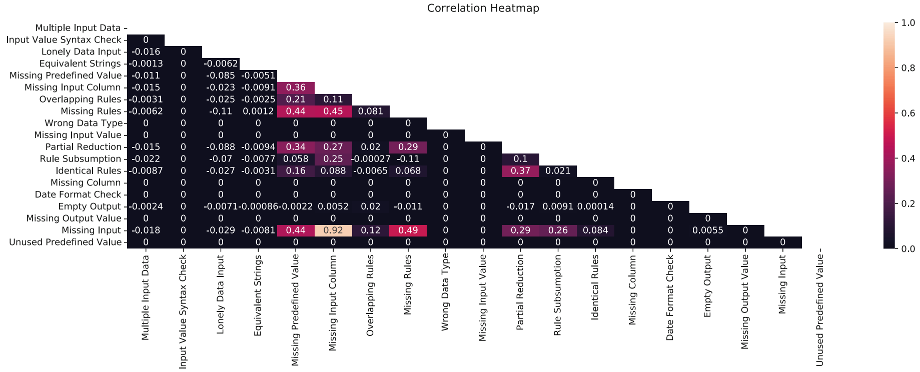


Fig. 7. Error type occurrence: correlation heatmap.

Regarding the observed error instances, there were few, yet strong correlations between the individual error types. For example, missing annotations of inputs on the DRD level were highly correlated with missing columns in the individual decision tables. Also, missing predefined values were associated with missing rules. This could indicate that some error types may have the same root cause, or leverage each other, and could give room for future work. As expected, there were no (strong) negative correlations ("the more x, the *less* y"). In future works, it could be interesting to investigate further correlations between error types and complexity metrics, e.g., the correlation between missing rules and the number of columns. Also, in future work, we aim to apply further filtering mechanisms (e.g., by size) and consequently compare error type distributions for different classes of DMN models.

5 Discussion and Conclusion

In this work, we presented empirical insights on modelling errors found in a collection of roughly 5 600 DMN models. As a clear result, our analysis shows that DMN modelling issues could be found in over a third of all models.

It is crucial to highlight that the analyzed dataset contains models created for research and learning purposes. Hence, one can assume that many of the models in the dataset have been created by beginner to novice modellers, and the results may be

---

[5] The Pearson correlation coefficient r is a metric that measures the strength of the linear relationship between two variables, defined via $r = \frac{\sum(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum(x_i-\overline{x})^2(y_i-\overline{y})^2}}$, where x_i, y_i are the individual samples of variables x, y, and $\overline{x}, \overline{y}$ are the mean values over x and y. Note that r is a value between -1 and 1, where 1 is a strong positive correlation, and vice versa for -1. For a positive number, the correlation can be read as "the more x, the more y".

different if the models were created by highly trained experts. However, this fact can also be seen as an advantage: as a consequence, the analysis results can be seen as good indicators of where to focus to make error-free DMN modeling easier for non-experts, or beginners. For example, the identified error types could be used as a basis for deciding where to focus on in DMN-related teaching and training. Even in a company context, organizations may want to let more (non-technical) domain experts work directly with the DMN models, so it can expected that support and verification functionalities will also be needed in such settings. Similar analysis of industry DMN models can be considered important future work.

From a practical perspective, the key findings of the presented research also offer relevant insights. Many of the analyzed models contain errors that could theoretically be identified with automated checks that are, to the best of our knowledge, not implemented in many industry-scale tools [5]. However, while some errors are prevalent in most or at least many models, others rarely occur, indicating a relatively low overall importance. Consequently, the findings can guide the implementation of verification functionality in industry-scale DMN modeling tools. Furthermore, there are some additional findings that may raise interesting questions for future research.

- While the tool used for modelling the DMNs (Signavio modeller) can theoretically detect some error types, still, exactly these errors occurred quite frequently. Hence, we can conclude that apparently, many users do not consider the automated verification capabilities of the tool to the fullest extent. There could be several possible reasons for this. For example: (i) the modeler does not see that the tool has identified a particular error (UX problem). Or: (ii) the modeler sees that the tool has identified a particular error but: (a) does not know how to fix the error; (b) finds fixing the error too cumbersome; (c) considers the error a false positive; (d) does not fix the error in order to keep it as an example (e.g. for teaching or research purposes). Future research could aim at identifying the reason or purpose (if any) for somewhat "fixable", yet unfixed errors. We speculate that the problem can, at least in some cases, be traced back to fixable human-computer interaction issues in the verification functionality.
- The strong correlation between some error types raises the question whether some errors have, in many cases, a joint root cause. Future research could attempt to identify such joint root causes of several errors through detailed empirical and formal analyses.
- While we classify the identified errors in terms of error types, the literature is still lacking research on the severity of different error types. For example, some of the identified error types (e.g., *Identical Rules*) do not necessarily impact the outcome of the decision (while other error types do), so it could be beneficial for companies to have some form of prioritization. Also, in some domains, it may even be desirable behavior to deploy models with some of the "errors" found in this study, in particular, some advantages of deploying incomplete models have been recently discussed[6]. Here, future research should focus on how to handle errors of different severities, and how to develop efficient safeguards in modelling tools without overly restricting the freedom of the modeler.

---

[6] https://decisioncamp2020home.files.wordpress.com/2020/06/decisioncamp2020.alanfish.pdf.

- Some error types that the tool used for modeling cannot automatically detect occur never or extremely rarely. While one may intuitively dismiss these error types as unimportant, future research on models that are in fact deployed in real-world contexts is important; e.g., the errors may occur in very complex models and their implications on decision execution in case of occurrence may be severe. In general, the study of run-time consequences of modeling errors may be a particularly interesting direction for future research.

In light of our results, we see a clear continuation of results from previous (empirical) studies [1,2] showing that human modelling errors occur frequently for the DMN standard. We therefore still see a strong need for research on error verification and modelling support, and aim to continue in this direction in future works.

References

1. Batoulis, K., Nesterenko, A., Repitsch, G., Weske, M.: Decision management in the insurance industry: standards and tools. In: Proceedings of the BPM 2017 Industry Track, Barcelona, Spain, pp. 52–63. CEUR-WS.org (2017)
2. Smit, K., Versendaal, J., Zoet, M.: Verification capabilities for business rules management in the Dutch governmental context. In: Proceedings of the 21st Pacific Conference on Information Systems (PACIS 2017), Langkawi, Malaysia, pp. 201–212. AISel (2017)
3. De Leoni, M., Felli, P., Montali, M.: Integrating BPMN and DMN: modeling and analysis. J. Data Semant. **10**(1–2), 165–188 (2021)
4. Kampik, T., et al.: SAP signavio academic models (2022)
5. Grohé, C.-C., Corea, C., Delfmann, P.: DMN 1.0 verification capabilities: an analysis of current tool support. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNBIP, vol. 427, pp. 37–53. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85440-9_3
6. Felli, P., de Leoni, M., Montali, M.: Soundness verification of decision-aware process models with variable-to-variable conditions. In: 19th International Conference on Application of Concurrency to System Design (ACSD), pp. 82–91. IEEE (2019)
7. Batoulis, K., Weske, M.: Soundness of decision-aware business processes. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNBIP, vol. 297, pp. 106–124. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65015-9_7
8. Vanthienen, J., Mues, C., Wets, G., Delaere, K.: A tool-supported approach to inter-tabular verification. Expert Syst. Appl. **15**, 277–285 (1998)
9. Hasic, F., Corea, C., Blatt, J., Delfmann, P., Serral, E.: Decision model change patterns for dynamic system evolution. Knowl. Inf. Syst. **62**, 3665–3696 (2020)
10. Blatt, J.: Prototyping a verification tool for decision model and notation. Master's thesis, University of Koblenz-Landau (2020)
11. Hasić, F., Corea, C., Blatt, J., Delfmann, P., Serral, E.: A tool for the verification of decision model and notation (DMN) models. In: Dalpiaz, F., Zdravkovic, J., Loucopoulos, P. (eds.) RCIS 2020. LNBIP, vol. 385, pp. 536–542. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50316-1_35
12. Hasić, F., Vanthienen, J.: Complexity metrics for DMN decision models. Comput. Stand. Interfaces **65**, 15–37 (2019)

2nd International Workshop on Digital Twins for Business Processes (DT4BP 2023)

2nd International Workshop on Digital Twins for Business Processes (DT4BP)

A Digital Twin of a Business Process is a virtual replica of the near-time process occurring in an organization. Thousands of sensors are placed to monitor physical processes, all collecting data from different dimensions, such as environmental conditions, behavioural characteristics of machines and their users, and work that is being performed. All this data is continuously collected, elaborated, and exchanged in such a way to guide day-to-day organization activities. Digital Twins of Business Processes, together with advancements in artificial intelligence, cloud, and edge computing, offer a great amount of business potential by predicting the future instead of analyzing the past of the processes. The representation of reality created by digital twins may allow us to evolve towards ex-ante business practices.

The main goal of this workshop is to promote the development and use of Digital Twins in the field of Business Processes. We aim to attract researchers and industry practitioners to discuss the adoption of Digital Twins to enhance business processes, up to the concept of Business Process Digital Twin. We aim to bring together practitioners and researchers from different communities such as business process management, information systems, software engineering, internet of things, artificial intelligence, process mining, and data mining who share an interest in the application of Digital Twins for Business Processes.

The 2nd edition of this workshop was held in Utrecht, the Netherlands. The workshop program was aligned with the BP-Meet-IoT workshop and kicked off with a shared keynote by Wil van der Aalst on the topic, a very well-known scientist in the BPM community for the profound impact that his research has had and is having on the field of process mining and workflow management. His keynote, entitled "Data Models in the Middle", made clear the need to cover the gap between generated application-specific data and the different techniques from AI, ML, DS, and tools and to address the differences found between these two "worlds" through the introduction of an intermediary layer based on models, specifically, Object-Centric Event Data (OCED) and Measured and Event Data (MAED).

The workshop attracted 4 international submissions. Each paper was single-blind reviewed by three members of the Program Committee. From these submissions, the top 2 were accepted as full papers for presentation at the workshop. The two regular papers presented at the workshop provided a mix of novel research ideas, as described below.

First, Wolfgang Groher and Uwe V. Riss introduced a vision of a Digital Twin of an Organization that tries to compensate shortcomings of PAIS, CRM, and other traditional systems. Process agility is a key concern as the proposal supports process redesign during execution. A prototype for the service sector was presented on the architectural bird's-eye level.

Second, the paper by Markus Rabe and Emre Kilic presented a proposal for a maturity model concerning the development stages of a digital twin for business processes together with the description of a digital business process twin referring to an automotive use case.

An open interactive session was conducted jointly with the BP-meet-IoT workshop, attracting the interest of around 20 participants who engaged in a discussion on the definition of Digital Twins and their relation with IoT and Business Processes.

We thank the authors for their contributions and the members of the Program Committee for their invaluable help in the reviewing and discussion phases. We hope that the reader will find this selection of papers useful to keep track of the latest advances of Digital Twins for Business Processes. We are looking forward to presenting new advances in future editions of the DT4BP workshop.

September 2023

Fabrizio Fornari
Pedro Valderas

Organization

DT4BP 2023 Workshop Chairs

Fabrizio Fornari	University of Camerino, Italy
Pedro Valderas	Universitat Politècnica de València, Spain

DT4BP 2023 Program Committee

Abel Armas Cervantes	University of Melbourne, Australia
Antonella Guzzo	University of Calabria, Italy
Artem Polyvyanyy	University of Melbourne, Australia
Barbara Re	University of Camerino, Italy
Bernard Rumpe	RWTH Aachen University, Germany
Daniel Calegari	Universidad de la República, Uruguay
Emanuele Laurenzi	University of Applied Sciences and Arts, Northwestern Switzerland, Switzerland
Estefanía Serral	Katholieke Universiteit Leuven, Belgium
Flavio Corradini	University of Camerino, Italy
Istvan David	University of Montréal, Canada
Lucineia Heloisa Thom	Federal University of Rio Grande do Sul, Brazil
Matteo Zavatteri	University of Padua, Italy
Oscar Fernando González Rojas	Universidad de los Andes, Colombia
Robert Andrei Buchmann	Babes-Bolyai University, Romania
Victoria Torres Bosch	Universitat Politècnica de València, Spain

Digital Twin of the Organization for Support of Customer Journeys and Business Processes

Wolfgang Groher(✉) and Uwe V. Riss

Eastern Switzerland University of Applied Sciences, Rosenbergstr. 59, 9001 St. Gallen, Switzerland
`{wolfgang.groher,uwe.riss}@ost.ch`

Abstract. We present a new approach for a Digital Twin of an Organization (DTO). Its focus is to better support weakly structured and knowledge-intensive business processes. Driving forces behind this development are the increasing demands for organizational agility, customer focus and information utilization. Existing business information systems show a number of shortcomings in this respect: Process-Aware Information Systems (PAIS) either neglect agility (e.g., workflow management systems as tightly framed PAIS) or process representation (e.g., groupware as unframed PAIS). Customer Relationship Management (CRM) systems fall behind in customer data integration and analysis for customer journey support, and Business Intelligence (BI) focuses too much on structured data. In contrast, a DTO is highly information-based, real-time enabled and visualization-oriented and thus better fits the requirements. However, organizations as complex socio-technical systems with open boundaries are challenging for digital twins. We have chosen a process-oriented representation for the virtual part of the DTO based on internal and external data. The increased availability of unstructured data, for example, from process contexts, supports process redesign during execution: this results in more process agility while maintaining a comprehensible process framing. An Insight Engine augmented with artificial intelligence processes the data. In a case study we present a DTO prototype for the service sector. More precisely, the DTO supports a marketing campaign from design to service execution associated to the respective customer journey, representing business process and customer journey in a joint representation.

Keywords: Digital Twin of an Organization · Insight Engine · Real-time Customer Journey Representation · Process-Aware Information Systems · Prototype

1 Introduction

Seizing the opportunities of digital transformation and adaptation to the digital era is a key challenge for companies across all sectors. The challenges are versatile: (1) customers expect more individualization, (2) more data is available and awaits immediate analytical evaluation, (3) fast changing digital technologies must be adopted, and this requires (4) business organizations which are adaptable and agile [32]. It also affects how services are

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 341–352, 2024.
https://doi.org/10.1007/978-3-031-50974-2_26

designed, implemented and delivered [38]. For the organizational perspective, this means a shift from efficiency towards effectiveness. Service organizations must proactively address customer needs and expectations but, at the same time, must aim at digital service automation.

This requires new approaches in business process management, which allow to include different perspectives, e.g. as in agent-based approaches [34, 35] Today companies' success is no longer achieved with predictable and efficient business processes alone. Instead, the new environment requires dynamic process design, which is adaptable and flexible to treat each customer individually. Service organizations must be highly attentive to their customers' journeys [37]. Analytical data processing and knowledge of the human expert are key. This takes a static process model to its limits.

The shift towards such processes requires the management of knowledge-intensive and weakly structured processes and the related IT-support. The requirements of such processes are already known [8, 29, 36], however, today new digital technologies are available. To address the challenges, Gartner has coined the concept of a Digital Twin of an Organization (DTO) [18]. The DTO concept is based on that of the already established "digital twin" in manufacturing. Here, it is mainly used for physical objects, such as machines or equipment. The DTO extends the idea of digital twins to processes and entire organizations [30]. Like the digital twin in manufacturing, the DTO is expected to serve as unique point of data reference for each process, providing a comprehensive real-time picture and basis for process simulation [20].

In this paper we will report on the development of a DTO for a service company to support a marketing campaign for a new service. Section 2 starts from traditional business information systems presenting the current developments of DTO approaches to supporting processes based on better information analytics. Section 3 conceptualizes the architecture of the DTO that we are using to support a service provider's marketing campaign. It is followed by Sect. 4, in which we explain how this DTO is used to enhance business process management and incorporate customer journeys. Section 5 elaborates the use case of a marketing campaign and its particular requirements for the target application in a current research project. In Sect. 6, we summarize the findings and point to future research.

2 State of Research on Business Processes and DTOs

In the last decades, information systems were mainly focused on efficiency, and the management of business processes formed the basis. The leading paradigm was the Process-Aware Information System (PAIS) [12]. With the growing focus on customers, Customer Relationship Management (CRM) systems gained a prominent position. The same holds for business intelligence (BI) systems with their analytical power used for decision support. However, the developing digital economy made distinct drawbacks in these business information systems visible.

First, the more business processes become automated, the less agility they provide (e.g., workflow management systems as tightly framed PAIS); or they allow for free collaboration but lack process representation (e.g., groupware as unframed PAIS). This is not new: knowledge-intensive and weakly structured processes have been a subject of research for the last 20 years [8, 9, 24, 36]. As modifications of traditional business process approaches, solutions such as "weakly-structured workflows" [36] or "Adaptive Case Management" [22] were suggested to address the requirements of less structured processes. Other approaches such as COREPRO [23] or ADEPTflex [27] also aimed at adaptation of process structures during runtime, see also [13].

Second, these systems often lack capabilities to process large volumes of mostly unstructured data, which are available today and are required for complex decision-making. Traditional, BI systems almost exclusively focus on structured data, whereas BI solutions for unstructured data were not available for a long time [7]. Meanwhile the processing of unstructured data, especially text mining [39], has made significant progress. Semantic analysis in the context of natural language processing (NLP) has evolved, spawning applications such as "insight engines" that have a significant impact on decision making and improving user experience [11].

Finally, CRM systems have fallen behind in customer data integration and analysis for customer journey support. For example, today's multi-channel services in the digital customer journey require significantly more diligence [3]. This includes the growing use of customer data in CRM and beyond [31]. Insight engines are capable to analyze all available data within an organization – both from internal and external data sources – and generate concrete responses, accurately aligned to an inquiry [14]. The latest technological development consists in the integration of insight engines with large language models [19], as used in the text-based dialog system ChatGPT. Such systems offer extended possibilities in the interaction with customers and generally in the handling of information.

To address the described challenges, we have resorted to the idea of a Digital Twin of an Organization (DTO) that opens up new opportunities [3, 26, 28, 30]. The fundamental idea of digital twins goes back to industrial engineering, where it supports production processes.. The transfer of this approach to organizations has led to the concept of the DTO, which increased effectiveness in process design. The power of the DTO lies in the fact that it is highly information-based, real-time enabled and visualization-oriented and thus better fits the requirements described above. However, organizations as socio-technical systems are challenging for digital twins due to their complexity and open boundaries. Becker and Pentland described baseline criteria for DTOs to address these challenges [5]:

(1) A DTO should cover dynamic, interdependent processes. For example, today's marketing processes require a high flexibility to meet the customers' demands and may be quickly adapted when new customer feedback arrives.
(2) A DTO should be tailored to organizations involving humans who learn from experience and build up individual knowledge. Experience must be kept and made available across all users of the DTO.
(3) A DTO should be designed to help humans with managing the organization: identifying new customer segments, enhancing features of existing products or services, etc. This particularly concerns decision support.
(4) A DTO should go beyond a dashboard. Dashboards are limited to visualization that shows where a process intervention might be required. DTOs aim at enabling managers to anticipate responses from customers and markets [10].

Organizations constantly interact with their environment including internal and external stakeholders. Processes and their respective interactions are mainly “informal” [15]. To this end, DTOs support organizations that are under increasing pressure to continuously learn and improve [6]. Finally, the fact that the DTO processes real-time data lays the foundation for timely decision making due to the analysis and consolidation of various information sources.

3 DTO Architecture

Having introduced the concept and described the opportunities of a DTO, we will now present the DTO architecture that we have used. This DTO architecture is mainly based on the experience in manufacturing. Thus, we used a 5-layer-architecture of digital twins that has been proposed as blueprint for industrial digital twins [4]. These 5 layers are (1) a Data Layer, which collects the data from the physical object, (2) a Data Processing Layer, which semantically enhances the data based on (3) a Models & Algorithms Layer. Further automated analysis of data is performed in (4) an Analytic Layer as support for a suitable presentation in (5) a User Interface Layer.

The primal real-word system of the digital twin significantly influences its architecture. This also holds for the processes and the customer journey touchpoints that are to be supported. Whereas a physical device can be equipped with sensors, an organization needs other types of data sources. Reflecting the situation, we decided to turn to textual information as main source. While data from information systems (e.g., the CRM system) can be easily accessed, data from customers are more difficult to collect. We also used external data sources (e.g., commercial databases for company profiles and business news). A second difference to standard digital twins consists the inclusion of human agents as information source [5]. As a source of information, they are complex to interpret and less reliable than technical sensors. In particular, it requires customers’ motivation to receive relevant information (e.g., contact feedback). Fortunately, efforts in this respect are in line with the trend to intensify customer value co-creation [1] and aim at a more active role of customers in service execution.

For the Data Processing Layer the DTO uses Squirro’s Insight Engine [33], which is designed for text mining und structuring. Main tasks consist in selecting suitable keywords and defining appropriate pipelines allows to assign information objects such as documents and news to concepts in a knowledge graph. It is to be remarked that the technologies of enterprise knowledge graphs coupled with machine learning have made significant progress [21]. This choice is related to the challenge of textual information with their heterogeneity and complexity that requires a different approach than the analytics for the multitude of sensor data.

The Model Layer consists of a knowledge graph that represents the most important entities associated with the process and their relationships. These entities represent certain types of information, for example, news related to a certain topic. Moreover, these entities are related to corresponding information objects from the Data Layer, for example, entities of the type “company challenge” are related to news texts that concern the respective challenge; these can come from different sources. The touchpoints of the customer journey are also described as such entities and refer to touchpoint-related information objects, for example, customer emails that were exchanged between the service provider and the customer during the interaction at this touchpoint. In addition, the knowledge graph can contain elements of the business model such as the value proposition or individual unique selling propositions. Details of this knowledge graph are described in the use case. However, it is to be mentioned that this knowledge graph is not fixed but can be always adapted to upcoming requirements. Its main purpose is to provide the mass of information objects with a comprehensible structure. Thus, the knowledge graph (entities and relations) gives the information objects a meaning, which goes beyond the abstract notion, and allows for more effective cognitive search [2, 16]. The definition of new entities and relations is part of the work process and might depend on the information objects that have been found in a specific inquiry. Finally, the knowledge graph serves as basic structure of the user interaction with the DTO in the User Interface Layer.

In our case, the Analytic Layer, which usually supports simulations, focuses on the communication and interaction between the employees and the customers. Its general goal in our case is to explore and anticipate customer behavior. At the current stage, simulation is quite challenging for this kind of DTO. This is due to the fact that we have complex decision-making of the agents involved and open boundaries of the socio-technical system. In consequence the relevant information is not predictable. Therefore, we replace simulation by experimentation for prediction. We will explain the experimentation later in the description of the case study.

In the present setting, the DTO is closely integrated with the CRM system, which serves as general storage for data about the interaction with customers. The CRM system also makes offline information such as telephone calls accessible that are included as notes. The same holds for the exchange of emails. The DTO extends the CRM systems by providing analytical services as well as by proving the possibility to redefine the knowledge graph and the processes.

Finally, as other digital twins, the DTO is more than just a recipient of data. As it lacks direct sensors there are also no actuators that directly execute decisions made from the DTO. The influence of human actors consists in initiating information retrieval processes or changing existing touchpoints to influence processes. It is obvious that customers cannot be controlled from the DTO, but the DTO helps to decide which information is provided to them and about the interaction with them in general.

The DTO user interface is to reflect the two major aspects: process and knowledge structure as well as their association with underlying information objects. In particular, the DTO allows to display how individual customers (or groups of customers) move between touchpoints and which information is exchanged. This also gives access to customer feedback and can be used to display individual real-time customer journeys. This is a decisive advantage of the DTO since currently real-time customer journeys are yet in their infancy and are mostly surveyed through individual customer inquiries.

4 Innovation in Business Process Management Through the DTO

Starting point for the development of the current DTO was the limitation of the traditional business process management in terms of agility. The logic of traditional BPM consists in following principles: (1) in the process design phase a business process is elaborated and the technical requirement for its implementation are specified, (2) in the implementation phase information systems are configured in a way that increases the efficiency of the process execution, (3) in the enactment phase process instances are supported along the modelled paths and data about execution qualities are collected, (4) in the diagnosis phase quality deviations and quality deficits of process in the executions are analyzed and the process model is adapted accordingly [12]. The focus of this procedure is to achieve a highly efficient process execution. However, this only works if most process executions follow the predefined model paths.

In knowledge-intensive processes such mainstream execution paths hardly exist, and rigid process models can become obstacles rather than support structures. The DTO aims at resolving this obstacle by removing the strict separation of business process design and execution. Thus, on the one hand, the DTO makes the actual process structure visible in real-time, so that additional process mining is not required. In this way, it provides a dynamic process representation of the organization and leads to more process transparency. On the other hand, the information infusion from process execution and external information sources allows for a timely check whether the current process design is effective and leads to the expected results.

If there are obstacles in the execution, for example, if the identification of customers for a given campaign does not yield results of sufficient quality (see the following example), existing tasks can be changed and/or new tasks can be introduced, that is, a change of the process design during execution.

Through the DTO's real-time information provision (e.g., from customer feedback), the efficiency and effectiveness of process adaptations can be immediately controlled. Such feedback is obtain by direct requests or analysis of other customer responses and behavior. This procedure assumes another task of digital twins, namely the simulation of possible changes. However, due to the fact that simulations may cause considerable costs, particularly for complex systems, experiments with small customer groups can also help to anticipate user reactions or other effects. If new tasks are first executed at a smaller scale (e.g., with limited number of customers), the outcome of these experiments can be used to improve the process. It is the integration of information-analytical tools that help us to interpret the results from such experiments. For example, it may turn out that an enhanced search is necessary to identify suitable addressees for a campaign and that this requires a different approach.

Summing up, we can say that the DTO represents a new type of PAIS that provides more process transparency than a project management or a case handling system. A decisive factor is the close integration with analytical tools and the possibility to immediately translate the insights from such analysis into relevant changes. In the following case description, we will show how this approach can be implemented and how it affects process execution.

5 Case Study

The partner for the implementation of the DTO is an established Swiss service company that offers services in the field of quality management. The company also provides trainings and offers certifications in this area. However, service provision makes up the major share of their sale. The recipients of the services are small and large companies in various sectors. The company employs more than 150 people in Switzerland, its turnover is around CHF 30 million, and the share of labor costs account for more than 80% of total costs. As many other service companies of that size, its digital transformation is still in an early phase.

As a first use case in this study, we have chosen the support of a marketing campaign for a new service offering. More specifically, the aim is to convince potential customers to decide for a service in the area of information security. Such a campaign is developed individually and shows a complex interplay of process enactment and information provision. The key goal of the campaign is to address the right customers for this service submitting the right information to them. The general goal is to make the proceeding from setting up the internal campaign to its entrance into a customer journey as interactive and digitally trackable as possible.

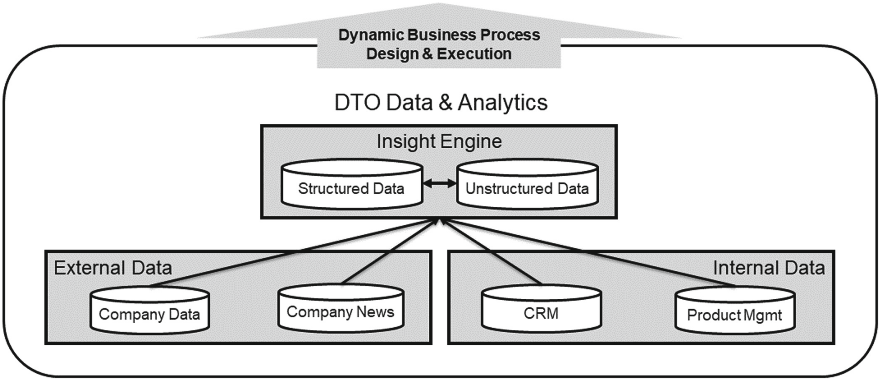


Fig. 1. DTO Data & Analytics.

To design the campaign, we started from internal data of existing customers from the CRM system and used information about existing offerings from the product management database. External data from a company database and from company news are also used to identify suitable prospects. All data, internal and external as well as structured and unstructured, are consolidated in the insight engine which allows for an AI-supported analysis that serves as the backbone of the DTO (see Fig. 1).

The first step in launching the campaign was planning and implementing the suitable activities in information research. In the first phase, we focused on a section of the entire campaign which is described in Fig. 2 (to reflect the evolution of the campaign over three development steps and flagged tasks as completed or planned, respectively). Starting the campaign requires the identification of companies with specific challenges; these challenges are to be addressed by the service provider's unique selling propositions (USPs) (see bottom process in Fig. 2). After identifying the USPs, the next activity consists in the elaboration of the corresponding challenges with the aid of the insight engine. Next, companies with the respective challenges were identified from a commercial company database. However, we faced the problem that the available company descriptions were not explicit enough to match them reliably with the challenges. To solve this problem, we assigned challenges to company features, e.g., physical data protection was associated with the operation of a network infrastructure, as an additional task (see center process in Fig. 2). With this adjustment, we were able to identify suitable companies, which met the intended quality and quantity criteria.

The DTO allows the redesign of a process while it is executed based on the latest information available. The changes become immediately effective in the respective execution. However, the aim of the DTO is not flexible process automation but human control and modification; as our case study in the service sector the processes are difficult to predict since each execution is rather specific. Therefore, the goal is better described in terms of organizational learning.

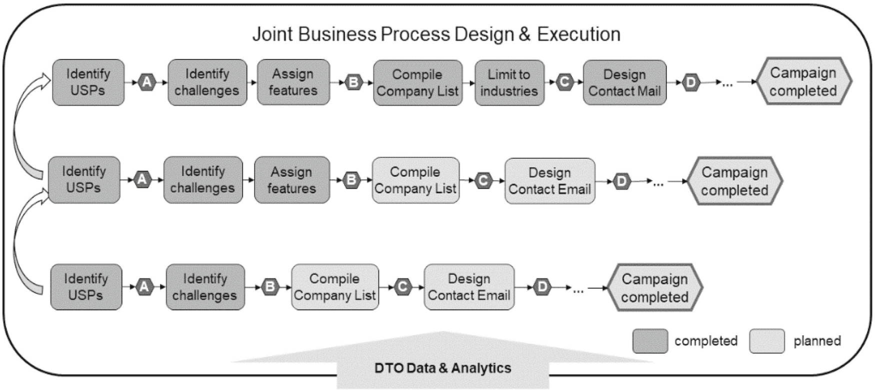


Fig. 2. Joint Business Process Design & Execution. The letters in the figure refer to quality thresholds for the activities: (A) USPs are specific enough, (B) features are clear enough, (C) the quality of the company list is sufficient, (D) emails are suitable.

In the course of setting up the marketing campaign we found that the identified companies were spread over various industries, resulting in difficulties to create an individual approach. Thus, we decided to narrow down the results to cover three focus industries: healthcare, information technology and metal and electrical industries, which reflect the service provider's key industries. This was another task we added.

The combined criteria, features and industry, allowed us to design three specifically formulated contact mails to address focus issues of individual target customers. To arrange the content, we extracted specific use cases from a company news database and added links to further in-depth information on the service provider's websites (see top process in Fig. 2).

Figure 3 provides more details how users interact with the DTO. The DTO interface model is based on the knowledge graph mentioned before. Its structure is used to represent the touchpoints in a Process & CJ (Customer Journey) View and gives users various options to select specific customer groups or touchpoint sections, for which information is displayed. This information is related to information objects that are presented in a

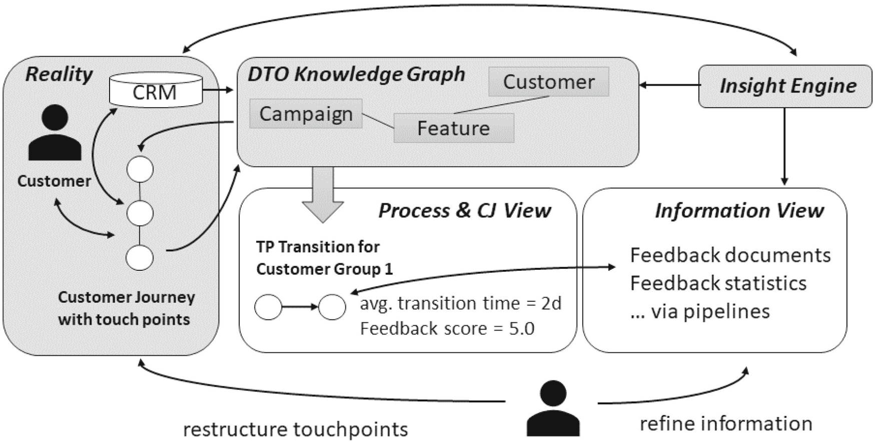


Fig. 3. Overview of the DTO use. Graphic adapted from [25].

separate Information View. This view is connected to the Insight Engine and shows on which retrieved information the displayed information is based.

The examples provided by now mark the differences between a DTO and traditional business process management. The DTO approach allows tasks to be changed within activities, to add or drop activities and to iterate within an activity. This is what we refer to as dynamic and information-based Business Process Design & Execution.

6 Discussion

The present conceptualization of the DTO, as it becomes manifest in the presented case study, represents a new type of PAIS that is adapted to the demands of the digital economy. The increasing agility of companies in all sectors requires an acceleration of the traditional lifecycle of process design, implementation, execution, and diagnosis. Highly individual customer journeys increase the pressure even more.

To address these challenges, we have chosen a process-oriented representation for the virtual part of the DTO that is based on internal as well as external data and analyzed by an insight engine augmented with artificial intelligence to process data from various available sources. This reflects the fact that artificial intelligence has been identified as one of the major future research topics in business process management [13]. The extended availability of process-related and other data allows for joint process design, monitoring, and execution (also suggested in [17]) and maintains a comprehensible process framing but with more flexibility. At the same time, the DTO places more emphasis on information retrieval activities as the predominant type of activity in the process. The insight engine allows to analyze structured as well as unstructured data from a variety of data sources. The outcome of such analysis essentially determines the process and is hard to predict.

As digital twins have been designed as information systems that deal with a plethora of unstructured data and transform them into tangible and executable representations.

The DTO is the adaptation of this paradigm to organizations with their more complex demands. In our project, the DTO equally deals with process models and process executions, which significantly increases process transparency and effectivity. In this respect it resembles dynamically adaptable data-driven approaches. To include the customer focus, they integrate the internal process view and the external customer journey view which often go hand in hand. The availability of more and more channels leads to very individual customer journeys that can no longer be grasped by traditional customer journey mapping techniques. Today, a customer journey representation requires up-to-date information about the customers' individual situation and perception. Such customer information can be taken from CRM and included in the analysis. With regard to the experimentation approach, our DTO aims at short-term process adaptations based on the latest information. This requires an intensified interaction with the customers as well as the use of customer-related information from external sources. Resulting process changes can be immediately implemented. In this way, the DTO does not only show process changes but also supports decision making to quickly react to insights from customer interaction.

Finally, one core feature of digital twins is the simulation of different scenarios. This enables testing certain configurations before implementation to avoid errors. In our DTO case, simulation is replaced by experimentation. The goal of anticipating the future is still the same. Process frames can be tested with specific, smaller customer groups before the roll-out to larger groups using the experience gained into account.

Eventually, the entwinement of data analytics and process execution has become the core characteristic of our DTO. This experience is significant for future incorporation of artificial intelligence components. In our project[1] we will further implement and evaluate the presented DTO design. For the future, we will focus on more detailed information and navigation in the user interface that help understand the evolution of the marketing campaign and how weak points can be identified and overcome.

References

1. Agrawal, A.K., Rahman, Z.: Roles and resource contributions of customers in value co-creation. Int. Strategic Manage. Rev. **3**(1–2), 144–160 (2015)
2. Alt, R., Zimmermann, H.-D.: Towards AI application marketplaces - an interview with Dorian Selz. Electron. Mark. **32**(1), 139–143 (2022)
3. Awasthi, P., Dubey, N., Sangle, P.S.: Contemporary challenges in CRM technology adoption: a multichannel view. Int. J. Electron. Customer Relat. Manage. **8**(1/2/3), 51–71 (2014)
4. Bazaz, S.M., Lohtander, M., Varis, J.: 5-dimensional definition for a manufacturing digital twin. Procedia Manuf. **38**, 1705–1712 (2019)
5. Becker, M.C., Pentland, B.T.: Digital twin of an organization: are you serious? In: Marrella, A., Weber, B. (eds.) BPM 2021. LNBIP, vol. 436, pp. 243–254. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94343-1_19
6. Berisha-Gawlowski, A., Caruso, C., Harteis, C.: The concept of a digital twin and its potential for learning organizations. In: Ifenthaler, D., Hofhues, S., Egloffstein, M., et al. (eds.) Digital Transformation of Learning Organizations, pp. 95–114. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-55878-9_6

---

[1] Innovation project 101.623 IP-SBM, by Innosuisse.

7. Blumberg, R., Atre, S.: The problem with unstructured data. DM Rev. **13**, 42–46 (2003)
8. Boissier, F., Rychkova, I., Le Grand, B.: Challenges in knowledge intensive process management. In: 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW), pp. 65–74. IEEE (2019)
9. Buck-Emden, R., Alda, S.: Systemunterstützung für wissensintensive geschäftsprozesse – konzepte und implementierungsansätze. In: Barton, T., Müller, C., Seel, C. (eds.) Geschäftsprozesse. Angewandte Wirtschaftsinformatik, pp. 101–125. Springer, Wiesbaden (2017). https://doi.org/10.1007/978-3-658-17297-8_6
10. Caporuscio, M., Edrisi, F., Hallberg, M., et al.: Architectural concerns for digital twin of the organization. In: Jansen, A., Malavolta, I., Muccini, H., et al. (eds.) ECSA 2020. LNCS, vol. 12292, pp. 265–280. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58923-3_18
11. Cleverley, P.H., Burnett, S.: Enterprise search and discovery capability: the factors and generative mechanisms for user satisfaction. J. Inf. Sci. **45**(1), 29–52 (2019)
12. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Process-Aware Information Systems. Bridging People and Software Through Process Technology. Wiley-Interscience, Hoboken (2010)
13. Dumas, M., Fournier, F., Limonad, L., et al.: AI-augmented business process management systems: a research manifesto. ACM Trans. Manag. Inf. Syst. **14**(1), 1–19 (2023)
14. Fallmann, D.: Insight engines im controlling. Control Manag. Rev. **64**(5), 58–61 (2020)
15. Groher, W.: Digitalisierung in FuE-schwachen KMU: ein IT-gestützter Ansatz zur Steigerung der Innovationsleistung, BTU Cottbus – Senftenberg (2022)
16. Hilger, J., Wahl, Z. (eds.): Making Knowledge Management Clickable. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-92385-3
17. Jansen, H.: ABB: from mining processes towards driving processes. In: Reinkemeyer, L. (ed.) Process Mining in Action, pp. 119–128. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-40172-6_15
18. Kerremans, M., Kopcho, J.: Create a digital twin of your organization to optimize your digital transformation program. Research Note G00379226, Gartner, Inc. (2019)
19. Marjanovic, O.: A novel mechanism for business analytics value creation: improvement of knowledge-intensive business processes. J. Knowl. Manage. **26**(1), 17–44 (2022)
20. Marmolejo-Saucedo, J.A., Hurtado-Hernandez, M., Suarez-Valdes, R.: Digital twins in supply chain management: a brief literature review. In: Vasant, P., Zelinka, I., Weber, G.-W. (eds.) ICO 2019. AISC, vol. 1072, pp. 653–661. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-33585-4_63
21. Miller, J.A., Mahmud, R.: Research directions in process modeling and mining using knowledge graphs and machine learning. In: Qingyang, W., Zhang, L.J. (eds.) SCC 2022. LNCS, vol. 13738, pp. 86–100. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-23515-3_7
22. Motahari-Nezhad, H.R., Swenson, K.D.: Adaptive case management: overview and research challenges. In: 15th Conference on Business Informatics, pp 264–269. IEEE (2013)
23. Müller, D., Reichert, M., Herbst, J.: A new paradigm for the enactment and dynamic adaptation of data-driven process structures. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 48–63. Springer, Cham (2008). https://doi.org/10.1007/978-3-540-69534-9_4
24. Papavassiliou, G., Ntioudis, S., Abecker, A., et al.: Supporting knowledge-intensive work in public administration processes. Knowl. Process Manage. **10**(3), 164–174 (2003)
25. Park, G., van der Aalst, W.M.P.: Realizing a digital twin of an organization using action-oriented process mining. In: 3rd International Conference on Process Mining, pp. 104–111. IEEE (2021)
26. Petänen, P., Salo, M., Hemilä, J., et al.: Identifying customer value of digital twin-enabled services process. In: Proceedings of the ISPIM Connects Athens – The Role of Innovation: Past, Present, Future, pp. 1–11. LUT Scientific and Expertise Publications (2022)

27. Reichert, M., Dadam, P.: ADEPTflex-supporting dynamic changes in workflow management systems without losing control. J. Intell. Inf. Syst. **10**(2), 93–129 (1998)
28. Riss, U.V., Groher, W.: Management instruments for digital service platforms. In: SAP Academic Community Conference 2022 DACH, pp. 209–221 (2022)
29. Riss, U.V., Rickayzen, A., Maus, H., et al.: Challenges for business process and task management. J. Univ. Knowl. Manage. **2**, 77–100 (2005)
30. Riss, U.V., Maus, H., Javaid, S., et al.: Digital twins of an organization for enterprise modeling. In: Grabis, J., Bork, D. (eds.) PoEM. LNBIP, vol. 400, pp. 25–40. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63479-7_3
31. Saarijärvi, H., Karjaluoto, H., Kuusela, H.: Customer relationship management: the evolving role of customer data. Mark. Intell. Plan. **31**(6), 584–600 (2013)
32. Sahu. N., Deng, H., Mollah, A.: Investigating the critical success factors of digital transformation for improving customer experience. In: Proceedings of CONF-IRM 2018 (2018)
33. Selz, D.: From electronic markets to data driven insights. Electron. Mark. **30**(1), 57–59 (2020). https://doi.org/10.1007/s12525-019-00393-4
34. Sulis, E., Taveter, K.: Agent-Based Business Process Simulation. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98816-6
35. van Elst, L., Aschoff, F.-R., Bernardi, A., et al.: Weakly-structured workflows for knowledge-intensive tasks: an experimental evaluation. In: 2003 Proceedings of the 12th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 340–345. IEEE (2003)
36. Voorhees, C.M., Fombelle, P.W., Gregoire, Y., et al.: Service encounters, experiences and the customer journey: defining the field and a call to expand our lens. J. Bus. Res. **79**, 269–280 (2017)
37. Yildiz, E., Møller, C., Bilberg, A.: Demonstration and evaluation of a digital twin-based virtual factory. Int. J. Adv. Manuf. Technol. **114**(1–2), 185–203 (2021)
38. Yu, E., Sangiorgi, D.: Service design as an approach to implement the value cocreation perspective in new service development. J. Serv. Res. **21**(1), 40–58 (2018)
39. Zaman, G., Mahdin, H., Hussain, K., et al.: Information extraction from semi and unstructured data sources: a systematic literature review. ICIC Express Lett. **14**(6), 593–603 (2022)

Framing the Digital Business Process Twin: From a Holistic Maturity Model to a Specific and Substantial Use Case in the Automotive Industry

Markus Rabe and Emre Kilic(✉)

IT in Production and Logistics, TU Dortmund University, Dortmund, Germany
`{markus.rabe,emre2.kilic}@tu-dortmund.de`

Abstract. The digital twin is a frequent subject of discussion in both academia and industry. High expectations are placed on the concept of the digital twin across many industries. The idea of transferring and adapting this concept to business processes reveals new potentials, perspectives, and paths for the advanced digital transformation of enterprises. The effective exploitation of the potentials of the digital twin in the business process context is of priority for many business sectors. While the idea of a business-process-related digital twin has already been discussed in academia, there is still a lack of holistic, proven, and well-defined real-world use cases. This paper aims to deliver a substantial and valuable contribution to the research regarding the integration of digital twins and business processes through understanding and describing the expected and realistic capabilities of the digital twin for business processes. This is achieved through the development of two deliverables: The thesis of this paper is positioned around the novel idea that the development of a business-process-related digital twin must follow an evolutionary course based on consecutive steps. This position is supported and reinforced by a first unique maturity model to distinguish between the development stages of a digital twin for business processes. As the second deliverable, the functionality of the depicted *digital business process twin* is applied and discussed within the perimeter of a specific use case from the automotive industry. The novelty of this paper results from the integration of the terms *digital model*, *digital shadow*, and *digital twin* into a maturity-based concept in the spotlight of business processes. The novelty is driven by applying this concept to a real-world scenario.

Keywords: Digital Twin · Digital Model · Digital Shadow · Business-process-related Digital Twin · Digital Business Process Twin · Automotive Industry · Business Processes · Digital Transformation

1 Introduction

The term digital twin has become a multi-layered and extremely adaptable constituent of digitalization efforts across all industries. The concept and understanding behind this term are designed in a versatile way and examined in different scenarios for multipurpose applications, e.g., in production [1], manufacturing [2], product development [3],

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 353–364, 2024.
https://doi.org/10.1007/978-3-031-50974-2_27

warehousing [4], or with regard to machining processes [5]. Further encouraging contemporary research practice of the digital twin is conducted in the context of business processes. This specific research field is on the rise, while appropriate real-world scenarios have yet to be developed. In addition, it is not yet profoundly examined to what extent a *digital model*, *digital shadow,* and *digital twin* differ from one another in the context of business processes. This paper is part of a wider-reaching research that aims to explore the in-depth capabilities and characteristics of a business-process-related digital twin to form a frame for a holistic and progressive concept in this field. In order to achieve this goal, this paper is divided into five sections. After this brief introduction, the current state of research regarding the digital twin in a business process context is depicted (Sect. 2). Based on this, a solution model is presented to highlight the distinct characteristics of the well-known terms *digital model*, *digital shadow,* and *digital twin* with respect to business processes (Sect. 3). A further essential part of this paper is to sketch a business scenario in which the process-related digital twin can be utilized effectively. In Sect. 4, the application of this digital twin concept to the end-to-end-control process of the model update handling of passenger cars in the automotive industry is a promising approach in solving the process-related challenges of this business field. The business process of the vehicle change management aims to control and manage the implementation of vehicle changes [6]. This process is highly complex, connected, and of high business value for automobile manufacturers. Therefore, it is very suitable to be considered and analyzed in the spotlight of the applicability of the business-process-related digital twin. Finally, in Sect. 5, the paper closes with a brief summary of the most important findings and an outlook to future developments.

2 Research Background

The description of the research background in this section will refer to the applied methodology and outline significant contributions regarding the digital twin in the context of business processes.

2.1 Methodology

The research methodology of this paper follows a straight course (Fig. 1). First, the current state of substantial research in relation to the digital twin in the process-related context is consolidated and described. Next, the research questions are formulated. A model-based solution approach is proposed in order to form a starting point to answer the research questions. Finally, the methodology finishes with the examination of the applicability of the process-related digital twin to a real-world scenario.

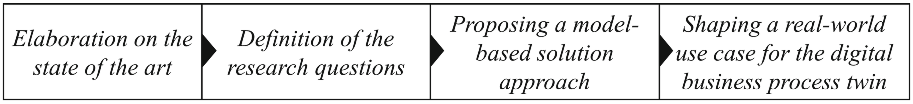

Elaboration on the state of the art
Definition of the research questions
Proposing a model-based solution approach
Shaping a real-world use case for the digital business process twin

Fig. 1. Research design and methodology of this paper based on a straight course

2.2 Related and Significant Contributions

The idea of using the digital twin concept for business processes has experienced an increase of interest and led to significant contributions. This field of research is still relatively young and both in-depth and extensive research is still required. Nevertheless, the idea of looking at processes in the spotlight of the digital twin is not completely new. For example, publications in this context already have been initially and systematically collected, consolidated, and presented in [7]. Each paper presented in that literature review introduces an own terminology for a process- or organization-related digital twin. However, these contributions do not ground their approach or idea on a holistic, uniform, and consistent concept that refers solely to business processes. Research contributions that form a foundation for the conceptual understanding of a business-process-related digital twin are in the primary scope of this paper. Research conducted by [7] provides a basic and solid concept of the business-process-related digital twin under the term *digital business process twin*. The contribution of this paper builds on this terminology, and the term is used throughout this work consequently to refer to a business-process-related digital twin.

The major aim of this paper is to evolve the characterization of the digital business process twin profoundly in order to propel the digital transformation of business processes in enterprises. Therefore, a maturity model is proposed to guide through the evolutionary growth stages of the digital business process twin. A maturity model typically consists of a sequence of maturity levels for a class of objects, describing an anticipated, desired, or typical developmental path of these objects in sequential, discrete ranks, starting with an initial stage through to full maturity [8]. An organization's current maturity level represents its capabilities towards a specific class of objects and application domains [9, 10]. Maturity models are used to assess as-is situations, to guide improvement initiatives, and to control progress [9, 11]. Maturity models of digital twins focus vastly on the original understanding of a digital twin, which is to mainly mirror and represent physical products [12–15]. But, no substantial research has been conducted for business-process-related digital twins. There is a clear lack of a well-recognized definition and characterization of the maturity levels for a digital twin with the sole aim to represent business processes. Therefore, it is challenging to distinctively portray the capabilities of a sophisticated digital twin for business processes and, thus, validate the effectiveness of such a digital representation. The applicability of the digital twin to business processes must be examined meticulously in a specific subject domain and field of research due to the wide-ranging, profound, and firm research that takes place in the field of business processes. Furthermore, it can be clearly observed that there is a gap in research regarding the development and depiction of a use-case-related characterization of the digital twin solely in the context of business processes. While there is an apparent increase regarding the use case development for the digital twin with focus on physical products, production, manufacturing processes, or healthcare applications [16–20], the use-case-centric characterization and examination of the business-process-related digital twin is still deficient. Certainly, to build a maturity model, the first crucial step is to clearly distinguish between the lifecycle stages of a digital twin for business processes and, thus, provide a starting point for appropriate and effective development of business-process-related use cases.

2.3 Research Questions

Based on the previous contributions, this paper aims to bring the conceptual development of the *digital business process twin* (DBPT) forward by answering the following research questions:

1. How can the terms *digital model*, *digital shadow,* and *digital twin* clearly be distinguished when they are applied to business processes, and how does the interaction between them look like when they are put together and examined in the same business process context?
2. In which feasible way can the DBPT be operated in a real-world business scenario?

In order to answer these research questions, two approaches will be presented in the following sections. First, a basic structure of a maturity model will be introduced to consolidate and highlight the terms *digital model*, *digital shadow*, and *digital twin* in the spotlight of business processes. Second, to clarify the stages of the maturity model and highlight the underlying concept, the practical applicability of the DBPT is discussed and examined via a real-world business scenario.

3 Maturity Model for the Digital Business Process Twin

This paper advocates the thesis that a thorough development of the DBPT must follow a cascaded progress along subsequent stages of maturity. The achievement of the highest level of maturity represented by the DBPT first requires the implementation of lower maturity levels, represented by the *digital business process model* (DBPM) and the *digital business process shadow* (DBPS) (Fig. 2). Through the differentiation between several development and maturity levels, the concept of the DBPT is not only divided into smaller and manageable development steps; the basic and essential understanding and characterization of the DBPT are also manifested and its key elements are emphasized. This novel approach shows that the implementation of a DBPT requires an evolutionary progress that accumulates the lower maturity levels of the digital twin development for business processes.

3.1 Basic Structure of the Maturity Model

On the way to the DBPT, maturity levels to be reached are proposed, which are specified in more detail using five evaluation dimensions. These dimensions are *People*, *Technology*, *Process*, *Data*, and *Controls.* The as-is situation of a digital representation is evaluated objectively based on these dimensions in order to support an exhaustive characterization and specification of the current level of maturity. With this step completed, the achievement of the next-higher maturity level can be initiated. The evaluation dimensions serve as a benchmark and provide subject domains for the derivation of quality indicators. This overall setup ensures that the outcomes produced through the successful achievement of a specific maturity level are characterized in a systematic way, providing a solid and transparent guidance for all activities regarding the development of a DBPT. This model is the foundation to assess the maturity of a digital twin that refers to a specific business process.

The maturity levels of the DBPT are evaluated and specified using the following subject domains:

1. The *People* domain evaluates the impact of the digital representation on the human factor, e.g., employees, suppliers, and customers. The organizational and cultural influence is assessed. Furthermore, it is essential to evaluate, which roles, skills, and expertise the employees have to meet in order to reach a certain level of maturity of the digital twin.
2. At each maturity level, a specific *Technology* is utilized as a tool to implement the maturity-level-specific properties and requirements of the digital twin. An important aspect is that elements of the technological solution of one level can be partly or completely reused on the next-higher level. Technology can relate to information systems, applications, and infrastructure.
3. The form and manifestation of the operational *Process* orientation need to be assessed. The essential evaluation factors are (1) the attention on the end-to-end aspect of a process and (2) the extent of formalization and standardization applied within the procedures of process execution and control.
4. Reaching a high level of maturity is essentially enabled by a consolidated, integrated, and consistent *Data* landscape. Formal and automated interfaces between distributed but essential IT systems are required for automated data exchange. Further aspects relevant to the assessment are whether the technical accessibility to the process-relevant data is possible and whether these data can be utilized for the specific applications of the corresponding digital twin of that specific maturity level.
5. Compliance with implemented process *Controls* and their appropriate design is subject to the assessment. This aspect is met by the maturity levels in a different form, quality, and effectiveness. An advanced and digitally supported execution and monitoring of controls of the business process makes a significant contribution to the structure and operation of compliant business processes.

Based on these evaluation domains, the successive development stages and, ultimately, the properties of the DBPT are characterized. The maturity model covers the scope between a purely static, model-based description of a specific business process and a detailed, formalized, and dynamically executable model at the operational level. Two dimensions span the evaluation space for the potential maturity levels. The maturity of the digital twin is evaluated along these dimensions. When determining the maturity of a DBPT, the degree of process formalization, standardization, and automation plays an important role as well as the extent to which the principles of operational excellence are considered and implemented.

3.2 Progressive and Distinct Development Stages of the Digital Business Process Twin

The digital business process *model* (DBPM) generates a coherent view of the business process as a model-based representation. The focus is on developing a valid and transparent picture of the cross-departmental end-to-end process, process interfaces, data flows, and involved IT systems that support the operational process execution. The major challenge at this maturity level is to develop an appropriate and purposeful model while

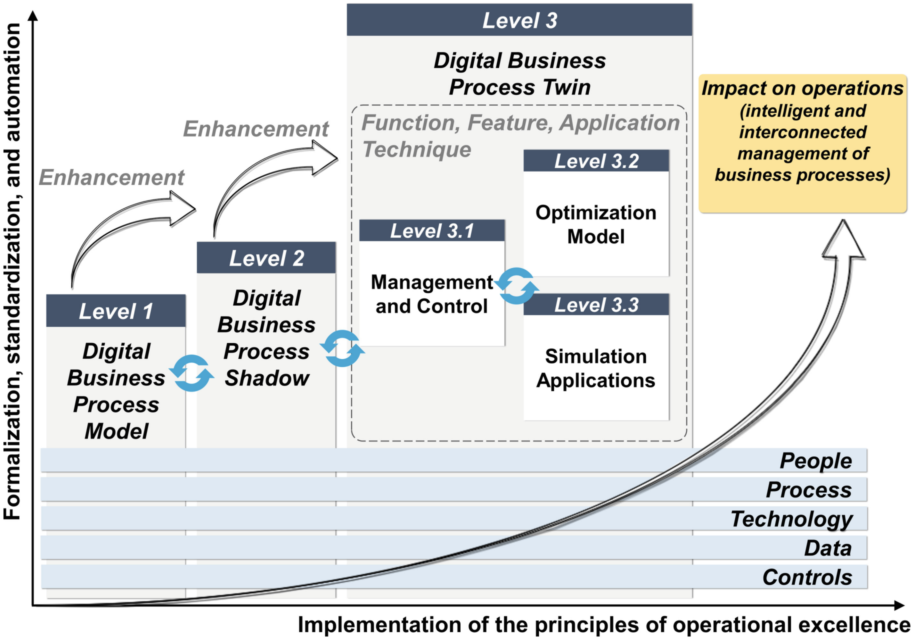

Formalization, standardization, and automation
Level 3
Digital Business Process Twin
Function, Feature, Application Technique
Enhancement
Enhancement
Level 3.2
Optimization Model
Level 3.1
Management and Control
Level 3.3
Simulation Applications
Level 2
Digital Business Process Shadow
Level 1
Digital Business Process Model
Impact on operations (intelligent and interconnected management of business processes)
People
Process
Technology
Data
Controls
Implementation of the principles of operational excellence

Fig. 2. Postulation of evolutionary stages of the digital twin development with respect to business processes

handling the model complexity sufficiently and efficient. Model quality includes, for example, the professional and valid differentiation between the relevant and irrelevant modeling aspects of the end-to-end process in the scope. The key factor for achieving this maturity level is the targeted and effective utilization of the knowledge of the involved subject matter experts in the modeling process and the applied technique to transfer this knowledge to profound models. The decisive factor to complete this task is to build a uniform and consistent understanding of the cross-departmental requirements and process dependencies. In addition, this level pursues the goal of developing and implementing an appropriate concept using a database-supported modeling tool and providing the deliverables for the next maturity level. An important specification at this point is that the modeling tool does not have to be connected to other systems in the IT landscape and can be operated proprietarily. The DBPM produces a consistent and integrated process and data model of the specific business process, which is used as a foundation for the next maturity level.

When referring to the digital business process *shadow* (DBPS), the process performance and the identification of disruptive factors in the process are at the center of the attention. These aspects are gaining importance, since the real process flow is monitored, visualized, and evaluated based on key indicators, e.g., throughput time, processing time, process variants, or the handling of certain events that represent specific disruptions in the process. The aim of the DBPS is to create a continuous virtual image (shadow) of selected real and relevant aspects of a business process, thus, fulfilling the requirement for a virtual real-time representation of that process. The object-centric view of the process is essential, since different objects (e.g., purchase order, invoice, requirements, or goods receipt) have to be linked to the end-to-end process, which operates through several departments. Harmonizing these key objects and integrating them into a common digital shadow of the real end-to-end processes is a major challenge. The DBPS combines the qualitative view of the process (delivered by the DBPM) with the quantitative data basis of the process. For this purpose, the model library of the DBPM is utilized and expanded subsequently.

After the model and data basis of the business process have been developed using the DBPM and DBPS, the functions of management and control, simulation, and optimization are addressed at the development stage of the digital business process *twin* (DBPT). Based on the insights gained from the DBPS, the DBPT triggers an action in a certain, previously examined source system of the business process. It is negligible whether this action is triggered manually by a person or automatically by the DBPT. Following the traditional understanding of a digital twin, the key factor is the automatic translation of this intention or trigger into a specific action in the source (real physical) IT system. The mapping of a target to the action is specified in the translation layer between the DBPT and the corresponding source IT system of the business process that runs and oversees the operational execution. Action triggers can also result from insights from a simulation or be based on the results of optimization or improvement models (in the context of business processes improvement factors could be, e.g., resource allocation or reduction of throughput times). Therefore, the cyclic and regular data exchange between the DBPS and DBPT plays a vital role. An advanced event handling and the utilization of progressive data analysis techniques at the maturity level of the DBPS deliver essential insights for the DBPT. These insights are processed and managed at the DBPT level in conjunction with domain-specific simulation applications and optimization models to trigger appropriate and accurate actions or events that achieve and realize the desired target state in the real business process.

4 Use-Case-Related Specification and Evaluation of the Digital Business Process Twin

In this section, an end-to-end use case for the DBPT is presented in accordance with the depicted maturity model in Sect. 3. This use case highlights not only the scope and detailed characterization of the DBPT, but also the perimeter of interaction among all digital representations. In order to shape the understanding of the maturity model and the contained successive levels, a suitable real-world application scenario from the model update process of passenger cars is demonstrated in an end-to-end business process view. The theoretical background as well as the motivation to examine this process in particular can be derived from [6]. Figure 3 shows the overarching context and the interrelation between the DBPT and the real end-to-end model update process of passenger cars. The model points out that taken as a whole the DBPT includes the functionality of the DBPS and the DBPM as well. This enclosing representation of functions results from the principle of successive and stepwise implemented maturity levels, with the next higher level being based on the results of the previous one.

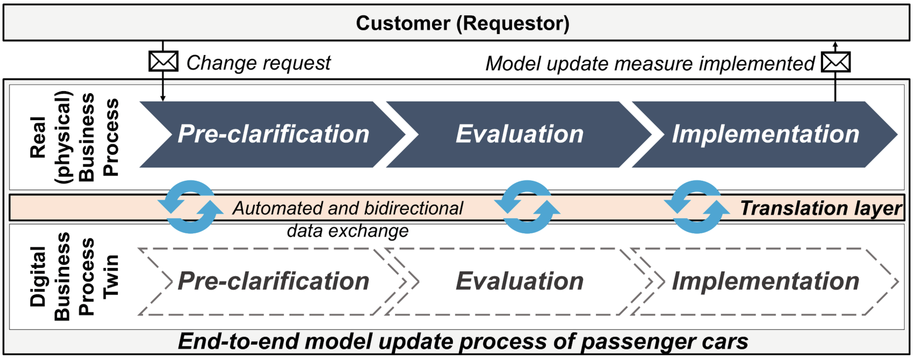

Customer (Requestor)
Change request
Model update measure implemented
Real (physical) Business Process
Pre-clarification
Evaluation
Implementation
Automated and bidirectional data exchange
Translation layer
Digital Business Process Twin
Pre-clarification
Evaluation
Implementation
End-to-end model update process of passenger cars

Fig. 3. Overarching context of the *digital business process twin* and its interrelation with the real business process of the end-to-end model update handling of passenger cars

In order to structure and explain the business use case for the DBPT properly, a process model was developed utilizing the BPMN specification. BPMN is an abbreviation of *Business Process Model and Notation* and has been developed to describe business processes for the implementation of IT processes or even to put them in distinct operation within suitably designed IT workflow systems. The end-to-end model update process is divided into the phase of *preliminary clarification*, *evaluation* (Fig. 4), and *implementation* (Fig. 5) of change requests. Some of the core elements of these phases are described at appropriate points in the following section in relation to the presented maturity model in Sect. 3. The DBPM specifies the complete design of the model update process. It defines the start and end event as well as the control and decision-making activities that occur between these two events. In addition, the process sequence is determined, and intermediate events that are relevant for the process control are specified explicitly. In Fig. 4, the DBPM shows that change requests for model update measures are checked for completeness and correctness as part of the preliminary clarification. In addition to a technical description of the model update measure, a change request must also include the affected vehicles and the desired SOP. Rules for the correct classification of the change request are attached to the corresponding task of the DBPM. Furthermore, this step uses specific operational data that are stored in a phase-specific IT application system. Before the application is handed over to the detailed assessment, a proof is performed to check whether all the required information is collected. This verification step is designed by the DBPM, while all corresponding events are handled by the DBPS. The DBPS identifies, captures, and handles real event triggers and forwards them to the DBPT for further utilization and processing. The DBPT triggers adequate activities to control the real end-to-end process and coordinates their effective execution. If the application is complete and free of errors, it is passed on to the evaluation phase. As part of the evaluation (Fig. 5), subject matter experts from involved departments must evaluate the change request as precisely as possible. If during the evaluation the DBPS records that the deadline for submitting the assessment results has been reached and certain departments have not submitted their assessment yet, a measure is initiated organizing that the missing assessments can be actively requested and collected by the DBPT. After a request has been evaluated by all involved departments, a decision is made based on the collected information whether to approve the request. If the change request is not approved, no further steps are performed. If the request is approved, the implementation of the model update measure for the envisaged SOP starts. In the implementation phase, it can happen that a certain change request has to be escalated to the management due to missing approvals or missed deadlines. The escalation activity is carried out by the DBPT as a control measure, while the DBPS follows up and reports on the escalation procedure. This event is modeled by the DBPM as the implementation task of the escalated request being halted until a decision has been made by the management regarding the further procedure. Another relevant case occurs in the implementation phase of approved model update measures, if the DBPS records that the scheduled supply of hardware or software is at risk. This is a serious risk for the scheduled start of the SOP and, thus, of high importance for the business success of a vehicle project. The DBPM depicts that the regular implementation process is not halted when this kind of event occurs. Rather, initiating and controlling the appropriate measure via the DBPT continues to be part of the range of tasks involved in implementation. In addition to this control mechanism, support for decision-making is an additional key characteristic of the DBPT. The DBPS consolidates the data basis for effective decision making and makes this data basis available in the context of the corresponding activity execution. When choosing an appropriate activity as a measure to the occurred undesired event, the potential impact of that activity on costs, deadlines, and consequently the SOP of the vehicle project must be considered. The DBPT selects the activity for which the lowest cost impact, the smallest deviation in the time line, and no SOP change is expected. A detailed procedure for the best possible decision-making is attached to the corresponding activity in the DBPT. For this complex task, the DBPT utilizes its simulation and optimization models and functions.

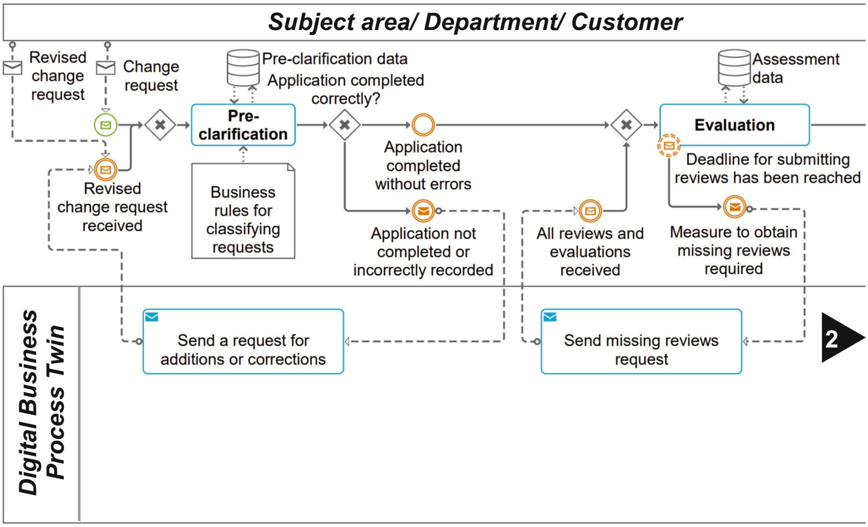

Subject area/ Department/ Customer
Revised change request
Change request
Pre-clarification data
Application completed correctly?
Pre-clarification
Revised change request received
Business rules for classifying requests
Application completed without errors
Application not completed or incorrectly recorded
All reviews and evaluations received
Assessment data
Evaluation
Deadline for submitting reviews has been reached
Measure to obtain missing reviews required
Digital Business Process Twin
Send a request for additions or corrections
Send missing reviews request
2

Fig. 4. *Pre-clarification* and *evaluation* phase of the end-to-end model update process supported by the *digital business process twin* with focus on monitoring and control

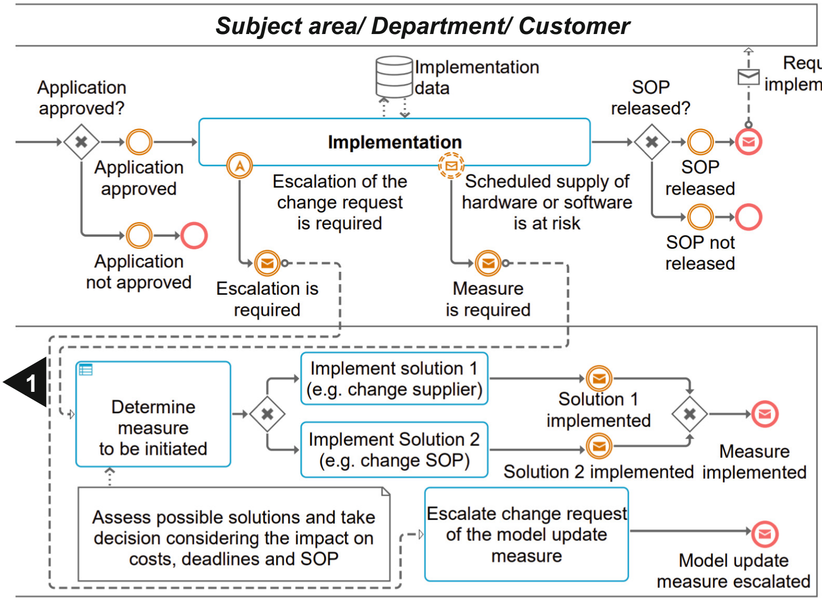

Subject area/ Department/ Customer
Application approved?
Application approved
Application not approved
Implementation data
Implementation
Escalation of the change request is required
Scheduled supply of hardware or software is at risk
Escalation is required
Measure is required
SOP released?
SOP released
SOP not released
Request implemented
1
Determine measure to be initiated
Implement solution 1 (e.g. change supplier)
Implement Solution 2 (e.g. change SOP)
Solution 1 implemented
Solution 2 implemented
Measure implemented
Assess possible solutions and take decision considering the impact on costs, deadlines and SOP
Escalate change request of the model update measure
Model update measure escalated

Fig. 5. *Implementation* phase of the end-to-end model update process supported by the *digital business process twin* with focus on decision support

5 Summary and Outlook

With this research on the business-process-related digital twin by means of applying the traditional understanding of the digital twin on business processes, a novel contribution towards framing, modelling, and testing the applicability of the digital business process twin under a real-world scenario has been made. This paper proposes a novel approach to structure a solid development path for the digital business process *twin* (DBPT). The development path along the maturity levels of the digital twin culminates ultimately in a DBPT. In conclusion, based on the presented use case it can be stated that basically the digital business process *model* specifies the design of an end-to-end business process, the digital business process *shadow* collects, archives, and checks the effectiveness of the design in relation to activities, events, controls, and decisions, and the DBPT ensures the effectiveness and success of the design through execution and prediction. Nevertheless, more research is required in the scope of digital twin and business processes. Most importantly, the digital twin of business processes has to be evaluated and tested under real-world conditions. For this purpose, first of all, it is essential to enhance and enrich the underlying concept as well as the maturity model of the digital twin for business processes thoroughly. In addition, elaborating extensive and detailed uses cases will reveal and magnify issues that hamper the direct and effective applicability of the DBPT in industry. These issues must be tackled systematically and profoundly in further research effort.

Some aspects could not be covered in this paper due to formal limitations. Emerging developments in the field of the business-process-related digital twin will play a crucial role in the profound and effective digital transformation of organizations. The authors are continuing intensive research on the digital business process twin guided by the principles of the presented maturity model with the sole aim of finding and delivering substantial outcomes to advance this research field.

References

1. Söderberg, R., Wärmefjord, K., Carlson, J.S., Lindkvist, L.: Toward a digital twin for real-time geometry assurance in individualized production. CIRP Ann. **66**, 137–140 (2017). https://doi.org/10.1016/j.cirp.2017.04.038
2. Krüger, S., dos Santos, S.B., Borsato, M.: Perceptions of a digital twin application case in the auto industry. In: Kim, K.-Y., Monplaisir, L., Rickli, J. (eds.) Flexible Automation and Intelligent Manufacturing: The Human-Data-Technology Nexus. LNME, pp. 528–536. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-17629-6_55
3. Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., Sui, F.: Digital twin-driven product design, manufacturing and service with big data. Int. J. Adv. Manuf. Technol. **94**, 3563–3576 (2018). https://doi.org/10.1007/s00170-017-0233-1
4. Bélanger, M.-J., Pellerin, R., Lamouri, S.: A literature review on digital twins in warehouses. Procedia Comput. Sci. **219**, 370–377 (2023). https://doi.org/10.1016/j.procs.2023.01.302
5. Liu, J., Zhou, H., Tian, G., Liu, X., Jing, X.: Digital twin-based process reuse and evaluation approach for smart process planning. Int. J. Adv. Manuf. Technol. **100**, 1619–1634 (2019). https://doi.org/10.1007/s00170-018-2748-5
6. Rabe, M., Kilic, E.: Towards a paradigm shift in vehicle project management: from a traditional multi-project-management to an integrated process-based digital approach. Procedia Comput. Sci. **219**, 362–369 (2023). https://doi.org/10.1016/j.procs.2023.01.301
7. Rabe, M., Kilic, E.: Concept of a business-process-related digital twin based on systems theory and operational excellence. In: 2022 IEEE 28th International Conference on Engineering, Technology and Innovation (ICE/ITMC) & 31st International Association for Management of Technology (IAMOT) Joint Conference, pp. 1–9. IEEE (2022). https://doi.org/10.1109/ICE/ITMC-IAMOT55089.2022.10033175
8. Becker, J., Knackstedt, R., Pöppelbuß, J.: Entwicklung von Reifegradmodellen für das IT-Management. Vorgehensmodell und praktische Anwendung. Wirtschaftsinformatik, Wiesbaden (2009)
9. Röglinger, M., Pöppelbuß, J., Becker, J.: Maturity models in business process management. Bus. Process. Manag. J. **18**, 328–346 (2012). https://doi.org/10.1108/14637151211225225
10. Rosemann, M., Bruin, T.: Towards a business process management maturity model. In: ECIS 2005 Proceedings of the thirteenth European Conference on Information Systems, pp. 1–12 (2005)
11. Iversen, J., Nielsen, P.A., Norbjerg, J.: Situated assessment of problems in software development. SIGMIS Database **30**, 66–81 (1999). https://doi.org/10.1145/383371.383376
12. Hu, W., Fang, J., Zhang, T., Liu, Z., Tan, J.: A new quantitative digital twin maturity model for high-end equipment. J. Manuf. Syst. **66**, 248–259 (2023). https://doi.org/10.1016/j.jmsy.2022.12.012
13. Chen, L., Xie, X., Lu, Q., Parlikad, A.K., Pitt, M., Yang, J.: Gemini principles-based digital twin maturity model for asset management. Sustainability **13**, 8224 (2021). https://doi.org/10.3390/su13158224
14. Uhlenkamp, J.-F., Hauge, J.B., Broda, E., Lutjen, M., Freitag, M., Thoben, K.-D.: Digital twins: a maturity model for their classification and evaluation. IEEE Access **10**, 69605–69635 (2022). https://doi.org/10.1109/ACCESS.2022.3186353

15. Medina, F.G., Umpierrez, A.W., Martinez, V., Fromm, H.: A maturity model for digital twin implementations in the commercial aerospace OEM industry. In: 2021 10th International Conference on Industrial Technology and Management (ICITM), pp. 149–156. IEEE (2021). https://doi.org/10.1109/ICITM52822.2021.00034
16. Patnaik, S., Das, P., Gaur, L., Jhanjhi, N.Z. (eds.): Digital Twins and Healthcare. Advances in Medical Technologies and Clinical Practice. IGI Global (2022). https://doi.org/10.4018/AMTCP
17. Shah, I.A., Sial, Q., Jhanjhi, N.Z., Gaur, L.: Use cases for digital twin. In: Patnaik, S., Das, P., Gaur, L., Jhanjhi, N.Z. (eds.) Digital Twins and Healthcare. Advances in Medical Technologies and Clinical Practice, pp. 102–118. IGI Global (2022). https://doi.org/10.4018/978-1-6684-5925-6.ch007
18. Roy, R.B., et al.: Digital twin: current scenario and a case study on a manufacturing process. Int. J. Adv. Manuf. Technol. **107**, 3691–3714 (2020). https://doi.org/10.1007/s00170-020-05306-w
19. Cimino, C., Negri, E., Fumagalli, L.: Review of digital twin applications in manufacturing. Comput. Ind. **113**, 103130 (2019). https://doi.org/10.1016/j.compind.2019.103130
20. Shiu, S.-C., Tang, K.-E., Liu, C.-W.: Digital twin-driven centering process optimization for high-precision glass lens. J. Manuf. Syst. **67**, 122–131 (2023). https://doi.org/10.1016/j.jmsy.2023.01.009

1st International Workshop on Formal Methods for Business Process Management (FM-BPM 2023)

1st International Workshop on Formal Methods for Business Process Management (FM-BPM 2023)

Contemporary business processes exhibit a high degree of inherent complexity. As a consequence, they may be affected by design mistakes that cause unwanted behavior and hence the possible violation of (some of) the predetermined objectives. Therefore, it becomes crucial that processes are correctly specified using a formal language, that their outcomes fulfill the intended requirements of the process and that they reach the planned goals. Formal Methods encompass a set of techniques and tools that exploit mathematics and formal logic for the specification, development, and analysis of complex systems, aiming at improving their reliability, trustworthiness and robustness. In Business Process Management (BPM), Formal Methods provide theoretical and practical toolkits to guarantee the correctness of processes and assess their capability to reach predetermined Key Performance Indicators. The International Workshop on Formal Methods for Business Process Management (FM-BPM) provides a platform for discussing new approaches and applications of already existing ones in the area of Formal Methods to model and analyze process-aware information systems and improve their quality.

FM-BPM 2023 was the first edition of the workshop, and attracted nine high-quality international submissions. Each paper was single-blind reviewed by at least three members of the Program Committee. Of all the submitted manuscripts, the top six were accepted for presentation – four of those being selected for publication as full papers and two as short ones. The four long manuscripts are published in these proceedings.

Fani Sani et al. presented an approach for computing approximation error bounds for efficient alignment-based conformance checking, so as to improve accuracy by analyzing the more suitable trace subsets in a log given the precalculated error bounds. Zavatteri et al. described a technique for ensuring soundness of unsound Data Petri Nets with difference logic guards. This technique was shown to work with input acyclic Data Petri Nets with sound Petri net components. Montali and Winkler studied notions of equivalences for Data Petri Nets and provided decision procedures for given subclasses of interest. Kuhlmann et al. proposed an approach for replay-based conformance checking of declarative process specifications written in Linear Temporal Logic on fixed traces using Answer Set Programming. The workshop concluded with the invited talk of Joost-Pieter Katoen on what probabilistic model checking can offer to BPM. Probabilistic model checking (PMC) is a fully automated technique to verify a broad range of properties on Markov models. In this talk, he presented preliminary results on how PMC can be used to perform stochastic conformance checking in BPM, and argued that PMC can be exploited for stochastic process mining.

We thank the authors for their noteworthy contributions, and the members of the Program Committee for their invaluable help in the reviewing and discussion phase of the manuscripts. We hope that the reader will benefit from reading papers to know more about recent applications of formal methods in business process management.

September 2023

Claudio Di Ciccio
Alessandro Gianola
Andrey Rivkin

Organization

Workshop Chairs

Claudio Di Ciccio	Sapienza University of Rome, Italy
Alessandro Gianola	Free University of Bozen-Bolzano, Italy
Andrey Rivkin	Technical University of Denmark, Denmark

Program Committee

Carl Corea	University of Koblenz-Landau, Germany
Jörg Desel	FernUniversität in Hagen, Germany
Chiara Di Francescomarino	University of Trento, Italy
Dirk Fahland	Eindhoven University of Technology, The Netherlands
Sérgio Guerreiro	INESC-ID/Instituto Superior Técnico, Portugal
Sylvain Hallé	Université du Québec à Chicoutimi, Canada
Thomas Hildebrandt	University of Copenhagen, Denmark
Cosimo Laneve	University of Bologna, Italy
Hugo A. López	Technical University of Denmark, Denmark
Andrea Marrella	Sapienza University of Rome, Italy
Fabio Patrizi	Sapienza University of Rome, Italy
Pascal Poizat	Université Paris Nanterre and LIP6, France
Jorge A. Pérez	University of Groningen, The Netherlands
Barbara Re	University of Camerino, Italy
Gwen Salaün	Grenoble Alpes University, France
Jianwen Su	University of California Santa Barbara, USA
Sergio Tessaris	Free University of Bozen-Bolzano, Italy
Emilio Tuosto	Gran Sasso Science Institute, Italy
Andrea Vandin	Sant'Anna School of Advanced Studies, Italy
Victor Vianu	University of California San Diego, USA
Matthias Weidlich	Humboldt-Universität zu Berlin, Germany
Mathias Weske	HPI - University of Potsdam, Germany
Sarah Winkler	Free University of Bozen-Bolzano, Italy
Matteo Zavatteri	University of Padua, Italy

Model-Independent Error Bound Estimation for Conformance Checking Approximation

Mohammadreza Fani Sani[1,2(✉)], Martin Kabierski[3], Sebastiaan J. van Zelst[1,4], and Wil M. P. van der Aalst[1,4]

[1] Process and Data Science Chair, RWTH Aachen University, Aachen, Germany
{s.j.v.zelst,wvdaalst}@pads.rwth-aachen.de
[2] Microsoft Development Center Copenhagen, Copenhagen, Denmark
fanisani@pads.rwth-aachen.de
[3] Department of Computer Science, Humboldt-Universität zu Berlin, Berlin, Germany
martin.kabierski@hu-berlin.de
[4] Fraunhofer FIT, Birlinghoven Castle, Sankt Augustin, Germany

Abstract. Conformance checking techniques quantify correspondence between a process's execution and a reference process model using event data. Alignments, used for conformance statistics, are computationally expensive for complex models and large datasets. Recent studies show accurate approximations can be achieved by selecting subsets of model behavior. This paper presents a novel approach deriving error bounds for conformance checking approximation based on arbitrary activity sequences. The proposed approach allows for the selection of relevant subsets for improved accuracy. Experimental evaluations validate its effectiveness, demonstrating enhanced accuracy compared to traditional alignment methods.

Keywords: Process mining · Conformance checking approximation · Alignments · Edit distance · Instance selection · Sampling

1 Introduction

Conformance checking, a sub-field of process mining, assesses the alignment between a process model and recorded event data [1]. Alignments, an established class of conformance checking artifacts [2], quantify deviations between recorded process execution and the intended behavior modeled by the process model.

Information systems generate vast amounts of event data that require efficient analysis. This *big event data*, combined with complex process models, leads to long computation times for alignments, limiting their practical application. However, in many cases, obtaining an approximate value is sufficient for meaningful

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 369–382, 2024.
https://doi.org/10.1007/978-3-031-50974-2_28

conclusions instead of exact alignment values. For instance, genetic process discovery [3], evaluating generations of *candidate process models* based on an event log requires impractical exact alignment results. Yet, determining if a newly generated process model improves alignment results is sufficient. Therefore, fast alignment approximation techniques with guaranteed error are valuable.

Various approaches for alignment approximation have been proposed recently [4–8]. In our previous work [4], we utilize subsets of the process model's behavior for alignment approximation [4]. Initially, we construct alignments for a subset of the process behavior and estimate the alignment cost for the remaining traces based on these alignments and edit distances, providing bounds for the approximated costs [4]. The quality of these subset-based approximations depends on the selected subset of model behavior [4]. Thus, quantifying the quality of an approximation based on a chosen subset aids in identifying suitable approximation subsets [4]. This paper therefore supports these approaches by introducing a novel approach for quantifying the quality of alignment approximations, which improves the error bounds introduced in our previous work [4].

Figure 1 presents a schematic overview of the proposed approach. A process model M models a process P that generates an event log L. Existing approaches compute *exact* or *approximate conformance checking results* by considering the language of the model $\mathcal{L}(M)$ (possibly infinite), or, a relevant finite subset thereof. Our method computes error bounds for alignment approximation using a proxy-set Ω. From Ω, we derive the relevant subset of process model behavior $\mathcal{L}_F(M)$ and use it to approximate alignment costs of traces in L. We also provide bounds on the introduced approximation error.

We evaluate our new error bound estimation technique using real event logs. Our experiments confirm a correlation between the maximum error bounds calculated a-priori and the eventual approximation error. The accuracy improves by using more suitable subsets of process model behavior with lower error bounds. Additionally, the computation time for error bounds is negligible compared to exact alignments.

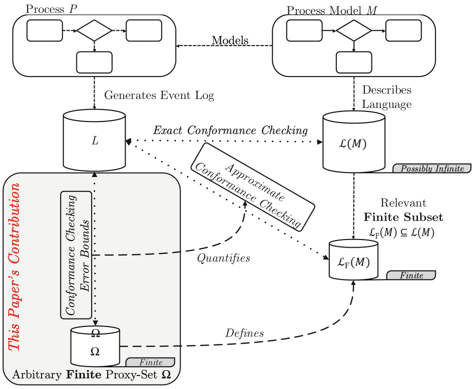

Process P
Process Model M
Models
Generates Event Log
Describes Language
L
Exact Conformance Checking
$\mathcal{L}(M)$
Possibly Infinite
Approximate Conformance Checking
Relevant
Finite Subset
$\mathcal{L}_F(M) \subseteq \mathcal{L}(M)$
$\mathcal{L}_F(M)$
Finite
This Paper's Contribution
Conformance Checking Error Bounds
Quantifies
Defines
Ω
Ω
Finite
Arbitrary Finite Proxy-Set Ω

Fig. 1. A process model M represents a process P generating an event log L. Existing approaches compute exact or approximate conformance checking results using the language of the model $\mathcal{L}(M)$. We propose quantifying error bounds for approximations by an arbitrary proxy-set Ω.

2 Related Work

Conformance checking techniques have been well-studied in the literature. In [1], different methods for conformance checking and its applications are covered.

Alignments, introduced in [9] have rapidly developed into the standard conformance checking technique. In [10], decomposition technique is proposed for improving the performance of the alignment computation. In the context of stream-based process analytics, in [5] the authors propose to incrementally compute prefix-alignments.

Few papers consider the use of sampling in process mining. In [11], the authors recommend a trace-based statistical sampling method to decrease the required time for process discovery. Moreover, in [12], we analyzed random and biased sampling methods with which we are able to adjust the size of the sampled data for process discovery.

Some research has focused on alignment approximation. General approximation schemes for alignments, i.e., the computation of near-optimal alignments, have been proposed in [13]. [6] proposes to incrementally sample the event log and check conformance on the sampled data. The approach incrementally increases the sample size until the approximated conformance value converges. The authors of [14] propose a conformance approximation method, that applies relaxation labeling methods on a partial order representation of a process model generated in a pre-processing step to produce alignments that are close to an optimal alignment. Furthermore, subset selection of model behaviors using instance selection [4] and simulation [15] have been proposed. The tool that supports these ideas is presented in [8]. In this context, in [7], the authors show, that a trie encoding of these selected subsets yields further runtime improvements.

3 Preliminaries

This section introduces conformance checking terminology and notation.

We let $\mathcal{B}(X)$ denote the set of all possible multisets over X. Given $b \in \mathcal{B}(X)$, $\overline{b} = \{x | b(x) > 0\}$. X^* denotes the set of all sequences over X. Let $X' \subseteq X$ and let $\sigma \in X^*$, $\sigma_{\downarrow_{X'}}$ returns the projected sequence of σ on set X', e.g., $\langle a, b, c, b, d \rangle_{\downarrow_{\{b,d\}}} = \langle b, b, d \rangle$. Let $X_1, X_2, \ldots, X_n$ be n arbitrary sets and let $X_1 \times X_2 \cdots \times X_n$ denote the corresponding Cartesian product. Let $\sigma \in (X_1 \times X_2 \cdots X_n)^*$ be a sequence of tuples, $\pi_i(\sigma)$ returns the sequence of elements in σ at position $1 \leq i \leq n$, e.g., $\pi_i(\langle (x_1^1, x_2^1, \ldots, x_n^1), (x_1^2, x_2^2, \ldots, x_n^2), \ldots, (x_1^{|\sigma|}, x_2^{|\sigma|}, \ldots, x_n^{|\sigma|}) \rangle) = \langle x_i^1, x_i^2, \ldots x_i^{|\sigma|} \rangle$.

Given $\sigma, \sigma' \in X^*$, $\delta(\sigma, \sigma') \in \mathbb{N}_{\geq 0}$ represents the *edit distance* (only using *insertions* and *deletions*) between σ and σ', i.e., the minimum number of edits required to transform σ into σ', e.g., $\delta(\langle w, x, y \rangle, \langle x, y, z \rangle) = 2$ (delete w and add z). Note that $\delta(\sigma, \sigma') = \delta(\sigma', \sigma)$ (δ is symmetrical) and $\delta(\sigma, \sigma'') \leq \delta(\sigma, \sigma') + \delta(\sigma', \sigma'')$ (triangle inequality applies to δ). Given a sequence $\sigma \in X^*$ and a set of sequences $S \subseteq X^*$, we define $\Delta(\sigma, S) = \min_{\sigma' \in S} \delta(\sigma, \sigma')$.

Event logs are collections of events that represent the execution of multiple process instances. They serve as the foundation for process mining algorithms. These events capture the timing of activities, denoted by their starting and finishing times, for each instance of the process identified by *Case*. In certain cases, such as alignment computation, only the control-flow information, which refers

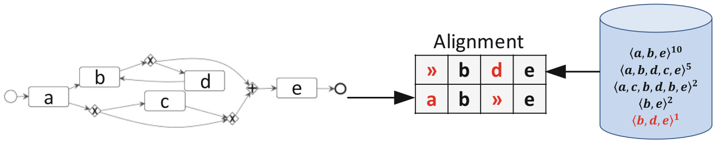

a
b
c
d
e
Alignment
» b d e
a b » e
$\langle a,b,e\rangle^{10}$
$\langle a,b,d,c,e\rangle^{5}$
$\langle a,c,b,d,b,e\rangle^{2}$
$\langle b,e\rangle^{2}$
$\langle b,d,e\rangle^{1}$

Fig. 2. A process model M_1 and an event log L_1. The optimal alignment of the last trace of L_1 and M_1 is shown in the middle of the figure.

to the sequences of executed activities within a process instance, is necessary. Thus, we utilize the aforementioned mathematical model of an event log.

Definition 1 (Event Log). *Let Σ denote the* universe of activities*. A* trace *σ is a sequence of activities ($\sigma \in \Sigma^*$). An* event log *$L \in \mathcal{B}(\Sigma^*)$ is a bag of traces.*

Process models are used to describe the behavior of a process. They can take the form of simple conceptual drawings or more complex mathematical concepts such as Petri nets and BPMN diagrams. An example of a process model is shown in Fig. 2 that uses BPMN notation. In this paper, we do not assume a specific modeling notation, but rather that process models describe activity sequences.

Definition 2 (Process Model). *Let Σ denote the universe of activities. A process model M describes the intended behavior of a process. We refer to the behavior described by model M as its* language *$\emptyset \subset \mathcal{L}(M) \subseteq \Sigma^*$, i.e., a collection of activity sequences.*

For M_1 in Fig. 2, we have $\mathcal{L}(M_1) = \{\langle a,b,e\rangle, \langle a,b,c,e\rangle, \langle a,c,b,e\rangle, \langle a,b,d,b,e\rangle, \ldots\}$. Due to the existence of loops, the language of a process model may be infinite.

To quantify whether an event log conforms to a process model, we use alignments. An alignment between a trace and a model describes which events in the trace can be "aligned with activities described by the process model". Furthermore, alignments indicate whether an event cannot be explained by the model or whether an activity as described by the model was not observed. In Fig. 2, an alignment of trace $\langle b,d,e\rangle$, and the given process model is provided. Observe that the trace does not contain activity a, which should always be present according to the model. In the alignment, this is visualized by the first column $\frac{\gg}{a}$. Similarly, after the observed d-activity, no second b-activity was observed. As such, in this alignment, the occurrence of d is rendered obsolete, i.e., visualized as $\frac{d}{\gg}$. We formally define alignments as follows.

Definition 3 (Alignment). *Let Σ denote the universe of activities, let M be a process model with corresponding language $\emptyset \subset \mathcal{L}(M) \subseteq \Sigma^*$ and let $\sigma \in \Sigma^*$ be a trace. An alignment γ of σ and M, is a sequence, characterized as $\gamma \in ((\Sigma \cup \{\gg\}) \times (\Sigma \cup \{\gg\}))^*$, s.t., $\pi_1(\gamma)_{\downarrow_\Sigma} = \sigma$ and $\pi_2(\gamma)_{\downarrow_\Sigma} \in \mathcal{L}(M)$. The set of all possible alignments of trace σ and model language $\mathcal{L}(M)$ is denoted as $\Gamma(\sigma, \mathcal{L}(M))$.*

Let $c\colon (\Sigma \cup \{\gg\}) \times (\Sigma \cup \{\gg\}) \to \mathbb{R}$ be an arbitrary cost function, assigning costs to the different type of alignments moves, then, given $\sigma \in \Sigma^*$, $M \subseteq \Sigma^*$ and $\gamma \in \Gamma(\sigma, \mathcal{L}(M))$, we let $\kappa_c(\gamma) = \sum\limits_{1 \leq i \leq |\gamma|} c(\gamma(i))$ denote the cost of alignment γ. We let $\Gamma_c^\star(\sigma, \mathcal{L}(M)) = \mathop{\arg\min}\limits_{\gamma \in \Gamma(\sigma, \mathcal{L}(M))} \kappa_c(\gamma)$ be the set of *optimal/minimal alignments*, i.e. the set of alignments, whose corresponding cost under the given cost function is minimal, and $z_c(\sigma, \mathcal{L}(M)) = \min\limits_{\gamma \in \Gamma(\sigma, \mathcal{L}(M))} \kappa_c(\gamma)$ be the optimal alignment cost for trace σ and model M (hence: $\forall \gamma \in \Gamma_c^\star(\sigma, \mathcal{L}(M)) \, (\kappa_c(\gamma) = z_c(\sigma, \mathcal{L}(M)))$). In the context of this paper, given $\gamma \in \Gamma_c^\star(\sigma, \mathcal{L}(M))$, we write $\varphi(\gamma) = \pi_2(\gamma)_{\downarrow \Sigma}$ to refer to *the "model behavior" corresponding to* σ, i.e., the projection of σ onto any of the closest possible execution sequence in M.

In the remainder, we assume that c represents the *standard cost function*, i.e., $\forall a \in \Sigma$, $c(a, \gg) = c(\gg, a) = 1$, $c(a, a) = 0$, and $c(a, a') = \infty$ if $a \neq a'$, and we omit it as a subscript.

4 Estimating Alignment Error Bounds

In this section, we derive error bounds for proxy sets Ω (Sect. 4.1). The *edit distance* between sequences provides upper and lower bounds for trace and model alignment costs. We also approximate optimal proxy sets that minimize cumulative approximation error (Sect. 4.2). Finally, we discuss enhancements to the bounds (Sect. 4.3).

4.1 Computing the Maximal Alignment Approximation Error

Here, we show that for given traces $\sigma, \sigma' \in \Sigma^*$ and a model M, the edit distance $\Delta(\sigma, \sigma')$ gives a range for the actual optimal alignment value $z(\sigma, \mathcal{L}(M))$. First, we show that under the standard cost function, we can use the edit distance for computing the cost of the optimal alignment between two arbitrary sequences.

Lemma 1 (Edit Distance Quantifies Optimal Alignment Costs). *Let Σ denote the universe of activities, let $\sigma \in \Sigma^*$ be a trace, let M be a process model and let $\gamma \in \Gamma^\star(\sigma, \mathcal{L}(M))$ be an optimal alignment of σ and M. Using the standard cost function, $\kappa(\gamma) = \delta(\sigma, \varphi(\gamma))$.*

Proof. γ only contains (a, a), $(a, \gg)$, and $(\gg, a)$ elements. Let R be the set of $(a, \gg)$ elements and I be the set of $(\gg, a)$ elements. Converting σ into $\varphi(\gamma)$ is done by removing *activities in σ and inserting activities represented by R and I, respectively. Thus, $\kappa(\gamma) = R + I$. Similarly, $\delta(\sigma, \varphi(\gamma))$ indicates the minimum number of insertions/removals to transform σ into $\varphi(\gamma)$. If $\kappa(\gamma) < \delta(\sigma, \varphi(\gamma))$, then $\delta(\sigma, \varphi(\gamma))$ does not represent the minimal number of edits. Likewise, if $\kappa(\gamma) > \delta(\sigma, \varphi(\gamma))$, then γ is not optimal.*

Corollary 1 ($\Delta(\sigma, \mathcal{L}(M))$ equals $z(\sigma, \mathcal{L}(M))$). *Let Σ denote the universe of activities, let $\sigma \in \Sigma^*$ be a trace, let M be a process model with corresponding language $\emptyset \subset \mathcal{L}(M) \subseteq \Sigma^*$. Using the standard cost function, $z(\sigma, \mathcal{L}(M)) = \Delta(\sigma, \mathcal{L}(M))$.*

Proof. Let $\gamma \in \Gamma^{\star}(\sigma, \mathcal{L}(M))$*, then,* $z(\sigma, \mathcal{L}(M)) = \kappa(\gamma) = \delta(\sigma, \varphi(\gamma)) = \Delta(\sigma, \mathcal{L}(M))$.

Again, assume the two traces to align to be $\sigma = \langle b, d, e\rangle$ and $\varphi(\sigma) = \langle a, b, e\rangle$ from Fig. 2. It is easy to see, that the edit distance of the two traces is 2 (insertion of a and deletion of b in σ), which is equivalent to the alignment cost.

Now, we show that, given an arbitrary sequence with known alignment cost, we can derive bounds for the possible alignment cost of another activity sequence. This allows the approximation of said cost without relying on the construction of alignments.

Theorem 1 (Edit Distance Provides Approximation Bounds). *Let* $\sigma, \sigma' \in \Sigma^*$ *be two traces and let* M *be a process model with corresponding language* $\emptyset \subset \mathcal{L}(M) \subseteq \Sigma^*$*. The optimal alignment value* $z(\sigma, \mathcal{L}(M))$*, is within* $\delta(\sigma, \sigma')$ *of* $z(\sigma', \mathcal{L}(M))$*, i.e.,* $z(\sigma', \mathcal{L}(M)) - \delta(\sigma, \sigma') \leq z(\sigma, \mathcal{L}(M)) \leq z(\sigma', \mathcal{L}(M)) + \delta(\sigma, \sigma')$.

Proof. Let $\gamma \in \Gamma^{\star}(\sigma, \mathcal{L}(M))$ *and let* $\gamma' \in \Gamma^{\star}(\sigma', \mathcal{L}(M))$*. Triangle inequality of edit distance yields* $\delta(\sigma, \varphi(\gamma')) \leq \delta(\sigma, \sigma') + \delta(\sigma', \varphi(\gamma'))$*, which we can rewrite (Lemma 1) to* $\delta(\sigma, \varphi(\gamma')) \leq \delta(\sigma, \sigma') + z(\sigma', \mathcal{L}(M))$*. Since* $z(\sigma, \mathcal{L}(M)) \leq \delta(\sigma, \varphi(\gamma'))$*, we have:* $z(\sigma, \mathcal{L}(M)) \leq z(\sigma', \mathcal{L}(M)) + \delta(\sigma, \sigma')$.

Similarly, $\delta(\sigma', \varphi(\gamma)) \leq \delta(\sigma, \sigma') + \delta(\sigma, \varphi(\gamma))$*. We deduce* $\delta(\sigma', \varphi(\gamma)) \leq \delta(\sigma, \sigma') + z(\sigma, \mathcal{L}(M))$*. As* $z(\sigma', \mathcal{L}(M)) \leq \delta(\sigma', \varphi(\gamma))$*, we deduce* $z(\sigma', \mathcal{L}(M)) - \delta(\sigma, \sigma') \leq z(\sigma, \mathcal{L}(M))$*. Hence, we obtain* $z(\sigma', \mathcal{L}(M)) - \delta(\sigma, \sigma') \leq z(\sigma, \mathcal{L}(M)) \leq z(\sigma', \mathcal{L}(M)) + \delta(\sigma, \sigma')$.

In Fig. 2, $z(\langle a, c, c, b, d, e\rangle, \mathcal{L}(M_1)) = 2$ and $\delta(\langle a, c, c, b, d, e\rangle, \langle a, c, b, d, e\rangle) = 1$. We deduce $1 \leq z(\langle a, c, b, d, e\rangle, \mathcal{L}(M_1)) \leq 3$. If $z(\langle a, c, c, b, d, e\rangle, \mathcal{L}(M_1))$ is unknown, $\delta(\langle a, c, c, b, d, e\rangle, \langle a, c, b, d, e\rangle) = 1$ implies that using it for approximating $z(\langle a, c, b, d, e\rangle, \mathcal{L}(M_1))$ *yields a maximal absolute approximation error of* 1.

4.2 Generating Proxy-Sets

Theorem 1 implies that, given a process model M and traces $\sigma, \sigma' \in \Sigma^*$, when using $z(\sigma', \mathcal{L}(M))$ for approximating $z(\sigma, \mathcal{L}(M))$, we obtain an approximation error $\epsilon \leq \delta(\sigma, \sigma')$, i.e. the maximum approximation error is $\delta(\sigma, \sigma')$. Interestingly, the error bounds on ϵ is determined independently of the model. Furthermore, σ' is allowed to be an arbitrary sequence, i.e., it is perfectly fine if $\sigma' \notin \mathcal{L}(M)$, and, given some $L \in \mathcal{B}(\Sigma^*)$ s.t. $\sigma \in \overline{L}$, $\sigma' \notin \overline{L}$. Hence, given an arbitrary set of sequences $\Omega \subseteq \Sigma^*$, $\underset{\sigma' \in \Omega}{\arg\min}\, \delta(\sigma, \sigma')$ represents the members of Ω that minimize the expected maximum error when using $z(\sigma', \mathcal{L}(M))$ for approximating (i.e., for $\sigma' \in \underset{\sigma' \in \Omega}{\arg\min}\, \delta(\sigma, \sigma')$).

For an event log $L \in \mathcal{B}(\Sigma^*)$ and proxy-set $\Omega \subseteq \Sigma^*$, $\forall \sigma \in \overline{L} \left(\min\limits_{\sigma' \in \Omega} \delta(\sigma, \sigma') = 0 \right) \Leftrightarrow \Omega \supseteq \overline{L}$, i.e., if every member of the log has an edit distance of 0 w.r.t. the proxy-set, then every member of the event log is a member of the proxy-set. Clearly, in such a case, using proxy-set Ω yields optimal alignments, yet, at the same (or even worse) time and memory complexity as computing conventional optimal alignments.

In the remainder, given an event log $L \in \mathcal{B}(\Sigma^*)$ and proxy-set $\Omega \subseteq \Sigma^*$, let $\epsilon_\Omega(L) = \sum_{\sigma \in \overline{L}} L(\sigma) \cdot \min_{\sigma' \in \Omega} \delta(\sigma, \sigma')$ be the accumulative approximation error of L using Ω. Given two proxy-sets $\Omega, \Omega' \subseteq \Sigma^*$, Ω *dominates* Ω' for event log L if and only if $\epsilon_\Omega(L) \leq \epsilon_{\Omega'}(L)$ and $|\Omega| < |\Omega|'$ and we refer to Ω' as a *redundant* proxy-set. A proxy-set Ω is k*-optimal* for event log L if and only if $\forall \Omega' \in \Sigma^* \left(|\Omega'| = k \implies \epsilon_\Omega(L) \leq \epsilon_{\Omega'}(L)\right)$. A k*-optimal* proxy-set Ω is k*-primal* if $|\Omega| = k$. For example, $\Omega = \overline{L}$ is $|\overline{L}|$-primal, 1-optimal, 2-optimal, ..., $|\overline{L}|$-optimal. Furthermore, it is easy to see that any (k-primal) proxy-set Ω with $|\Omega| > L$ is dominated by L and hence redundant. More interestingly, primal proxy-sets that are smaller than the event log are never redundant.

Theorem 2 (Primal Proxy-Sets are Non-Redundant). *Let $L \in \mathcal{B}(\Sigma^*)$ be an event log, $\Omega \subseteq \Sigma^*$ be a proxy-set such that $|\Omega| < |\overline{L}|$, and Ω is* k-primal. *Ω is* non-redundant.

Proof. Assume that Ω is redundant. Hence, $\exists \Omega' \subseteq \Sigma^* \left(|\Omega'| < |\Omega| \wedge \epsilon_{\Omega'}(L) \leq \epsilon_\Omega(L)\right)$. However, observe that, we are able to create $\Omega'' = \Omega' \cup L''$ with $|L''| = |\Omega| - |\Omega'|$ and $\sigma \in L'' \implies \sigma \in \overline{L} \wedge \sigma \notin \Omega'$ (note that $|\Omega| = |\Omega''|$). Observe that $\epsilon_{\Omega''}(L) < \epsilon_{\Omega'}(L)$ and as a consequence $\epsilon_{\Omega''}(L) < \epsilon_\Omega(L)$, contradicting the fact that Ω is k-primal.

Note, Theorem 2 implies the existence of a k-primal proxy-set Ω for any event log $L \in \mathcal{B}(\Sigma^*)$ and $k \in 1, 2, \ldots, |L|$. This k-primal proxy-set minimizes the accumulative approximation error $\epsilon_\Omega(L)$ for size k and can be considered the optimal proxy-set for that size. However, finding such proxy-sets is an NP-Hard problem and goes beyond the scope of this paper. Instead, here we focus on proxy-sets where $\Omega \subseteq L$. In the following paragraphs, we introduce different methods for generating proxy-sets and their relation to the optimal primal proxy-sets.

Sampling. Proxy-sets can be generated using sampling methods, either directly from the event log, the given process model, or a mixture thereof. In previous work, we investigated the sampling of model behavior using uniform distributions [4] and event-log-guided process model simulation [15].

Strictly sampling the behavior from the process model, i.e., $\Omega \subseteq \mathcal{L}(M)$, particularly when using event log-guided simulation yields (under standard cost function) $z(\sigma', \mathcal{L}(M)) = 0$, $\forall \sigma' \in \Omega$, and thus $0 \leq z(\sigma, \mathcal{L}(M)) \leq \Delta(\sigma, \sigma'), \forall \sigma \in L$. While it is very unlikely that such a proxy-set is k-primal due to it being closer to the log behaviour, $z(\sigma', \mathcal{L}(M)) = 0$, $\forall \sigma' \in \Omega$, can be exploited.

Sampling Ω from the event log is likely to result in a proxy-set that is closer to a k-primal solution, especially when prioritizing $\sigma \in \overline{L}$, for which $L(\sigma)$ is high. Hence, using log-based sampling is more likely to minimize ϵ_Ω. However, since the actual $z(\sigma', \mathcal{L}(M))$ for $\sigma' \in \Omega$ is unknown, we cannot tighten the estimator.

Centroid-Based Clustering. For a given target size k, the optimal proxy-set is *k-primal.* As an alternative to sampling, *clustering algorithms* are suitable for proxy-set selection. These algorithms group objects into clusters based on their similarity or distance, often using the edit distance as a metric. *Centroid-based clustering algorithms*, such as *K-Medoids* [16], are particularly relevant as they assign objects to the centroid with the minimal distance. While clustering algorithms can be applied to any set of activity sequences, applying them to the input event log produces proxy-sets close to the *k-primal* solution. Due to the time-consuming nature of providing optimal clustering solutions, several faster approximation techniques have been proposed.

4.3 Improving the Alignment Approximation Bounds

We showed that Ω and proxy-sequence $\sigma' \in \Omega$ can quantify the approximation error ϵ as $\epsilon \leq \delta(\sigma, \sigma')$ when approximating $z(\sigma, \mathcal{L}(M))$ with $z(\sigma', \mathcal{L}(M))$. Now, we show how using proxy-sets can improve alignment approximation bounds.

When approximating alignments using Ω, we first compute the alignments of Ω traces. We derive the bounds of the alignment cost of $z(\sigma, \mathcal{L}(M))$ by simply adding/subtracting $\delta(\sigma, \sigma')$ to $z(\sigma', \mathcal{L}(M))$. Note, when using the standard cost function, the lower bound of any alignment cannot be lower than 0. In certain cases, we can derive a tighter lower bound. Let $\Sigma_M = \{a \in \Sigma | \exists \sigma \in \mathcal{L}(M)(a \in \sigma)\}$, then, for any $\sigma \in \Sigma^*$, $z(\sigma, \mathcal{L}(M)) \geq |\sigma_{\downarrow_{\Sigma \setminus \Sigma_M}}|$, i.e., the elements of $\sigma_{\downarrow_{\Sigma \setminus \Sigma_M}}$ are always moves of the form $\frac{a}{\gg}$. Furthermore, in case $|\sigma| < \min\limits_{\sigma' \in \mathcal{L}(M)} |\sigma'|$, we need at least $|\sigma'| - |\sigma|$ (where $\sigma' \in \arg\min\limits_{\sigma' \in \mathcal{L}(M)} |\sigma'|$) moves of the form $\frac{\gg}{a}$. Hence, the theoretical lower-bound of any $\sigma \in \Sigma^*$ is equal to $\max(0, \min\limits_{\sigma' \in \mathcal{L}(M)} (|\sigma'|) - |\sigma|) + |\sigma_{\downarrow_{\Sigma \setminus \Sigma_M}}|$. We correspondingly define the Ω-driven lower and upper bound as follows.

Table 1. Statistics regarding the real event logs that are used in the experiment.

Event Log	Activities	Traces	Variants
BPIC-2012	23	13087	4336
BPIC-2018-*Inspection*	15	5485	3190
BPIC-2019	42	251734	11973
Hospital-Billing	18	100000	1020
Road	11	150370	231
Sepsis	16	1050	846

Definition 4 (Ω-Driven Alignment Bounds). *Let Σ denote the universe of activities, let M be a process model with corresponding language $\emptyset \subset \mathcal{L}(M) \subseteq \Sigma^*$*

and let $\Omega \subseteq \Sigma^*$ *be a proxy-set. Let* $\top_{\Omega,M} : \Sigma^* \to \mathbb{N}$ *denote the* Ω*-driven upper bound and* $\bot_{\Omega,M} : \Sigma^* \to \mathbb{N}$ *the* Ω*-driven lower bound, s.t.:*

$$\top_{\Omega,M}(\sigma) = \min_{\sigma' \in \Omega}(z(\sigma', \mathcal{L}(M)) + \delta(\sigma, \sigma')) \tag{1}$$

$$\bot_{\Omega,M}(\sigma) = \max(\max(0, \min_{\sigma' \in \mathcal{L}(M)}(|\sigma'|) - |\sigma|) + |\sigma_{\downarrow_{\Sigma \setminus \Sigma_M}}|, \max_{\sigma' \in \Omega}(z(\sigma', \mathcal{L}(M)) - \delta(\sigma, \sigma'))) \tag{2}$$

Finally, given $\top_{\Omega,M}$ and $\bot_{\Omega,M}$, we quantify the approximated alignment cost of $\sigma \in \Sigma^*$ as the average of bounds to minimize the possible approximation error, i.e., $\hat{z}_\Omega(\sigma, \mathcal{L}(M))$, as $\hat{z}_\Omega(\sigma, \mathcal{L}(M)) = \frac{\top_{\Omega,M}(\sigma) - \bot_{\Omega,M}(\sigma)}{2}$. In theory, it is possible to assign different weights to bounds based on additional knowledge or bias.

5 Evaluation

To assess the efficacy of the proposed error bounds, we conducted an extensive evaluation using multiple publicly available event logs. In particular, we explored the accuracy and the runtime performance of the proposed bounds. First, we briefly describe the implementation and evaluation setup (Sect. 5.1), followed by a discussion of the evaluation results (Sect. 5.2).

5.1 Experimental Setup

To evaluate the proposed error bounds, we implemented the *Conformance Approximation* plug-in in the `ProM` [17] framework[1], including various proxy-set generation methods (cf. Sect. 4.2).

The proposed methods were applied to six real event logs, and basic information about these logs, such as the number of distinct activities, traces, and variants, is provided in Table 1. For each event log, we apply conformance checking using process models obtained via the Inductive Miner algorithm [18] with infrequent thresholds of 0.2, 0.4, and 0.6. Four proxy-set generation methods are used:*random sampling*, *frequency-based sampling*, *K-Medoids clustering* and *K-Center clustering*. In random sampling, variants are uniformly sampled (without replacement) from the event log. In frequency-based sampling, traces are selected based on their $L(\sigma)$-values in descending order. K-Medoids clustering determines centroids by minimizing pairwise dissimilarity between traces, while K-Center clustering minimizes the maximum distance between centroids and traces. Proxy-set sizes were varied using different percentages (5%, %10, 20%, 30%, 50%) of the number of variants in the event logs. Each experiment was repeated four times.

---

[1] svn.win.tue.nl/repos/prom/Packages/LogFiltering.

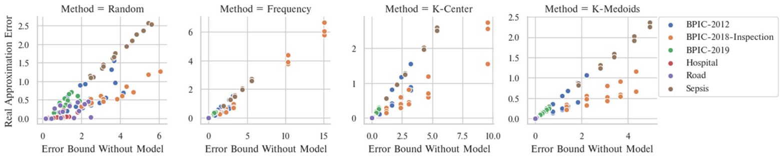


Fig. 3. Scatter plots of the maximum approximation error and the real approximation error using different proxy-set generation methods.

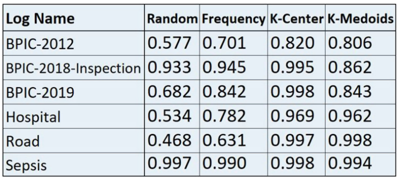
5.2 Results

First, we analyze the relationship between maximum and actual approximation error. Next, we examine the time performance of the estimation. Finally, we assess the effectiveness of the proposed lower bound.

Maximum Approximation Error Versus Approximation Error. Observe that minimizing the expected maximum error, e.g., by selecting a seemingly optimal proxy set, does not guarantee a minimal approximation error. For example, given some model M, $\sigma \in \Sigma^*$, $\Omega = \{\sigma_1, \sigma_2\}$ and $\Omega' = \{\sigma_1, \sigma_3\}$, assume that $\delta(\sigma, \sigma_1) = 2$, $\delta(\sigma, \sigma_2) = 3$ and $\delta(\sigma, \sigma_3) = 1$. Clearly, the maximal error based on Ω is 2, and, based on Ω', it is 1. As such, we intuitively favor Ω' over Ω. However, if $z(\sigma_1, \mathcal{L}(M)) = 7$, $z(\sigma_2, \mathcal{L}(M)) = 2$ and $z(\sigma_1, \mathcal{L}(M)) = 6$, we obtain $\bot_{\Omega,M}(\sigma) = \top_{\Omega,M}(\sigma) = 5$, whereas $\bot_{\Omega',M}(\sigma) = 5$ and $\top_{\Omega',M}(\sigma) = 7$. Hence, from Ω, we derive that $z(\sigma, \mathcal{L}(M)) = 5$ (note $\hat{z}_{\Omega}(\sigma, \mathcal{L}(M)) = 5$), whereas from Ω', we derive $5 \leq z(\sigma, \mathcal{L}(M)) \leq 7$ (with $\hat{z}_{\Omega'}(\sigma, \mathcal{L}(M)) = 6$). Thus, utilizing Ω gives the exact alignment value, whereas using Ω' yields an error of 1.

Given that there is no causal relation between the maximum approximation error and the actual error, we investigate, the strength of the correlation between the maximum approximation error and the effective approximation for each of the proposed proxy-set generation methods. The scatter plots in Fig. 3 illustrate these values for each method, distinguishing event logs with different colors.

Table 2. Pearson correlation coefficients between the maximum approximation error and the real approximation errors for different methods.



Log Name	Random	Frequency	K-Center	K-Medoids
BPIC-2012	0.577	0.701	0.820	0.806
BPIC-2018-Inspection	0.933	0.945	0.995	0.862
BPIC-2019	0.682	0.842	0.998	0.843
Hospital	0.534	0.782	0.969	0.962
Road	0.468	0.631	0.997	0.998
Sepsis	0.997	0.990	0.998	0.994

Moreover, we present the Pearson correlation coefficients in Table 2. The K-Center method demonstrates the highest correlation across all event logs. Notably, strong correlations between the maximum approximation error and the effective approximation error are observed for frequency-based sampling, K-Center, and K-Medoids. In contrast, random sampling exhibits a weaker correlation, especially for the Hospital-Billing and Road logs, where representative variants are limited, and random sampling fails to prioritize them.

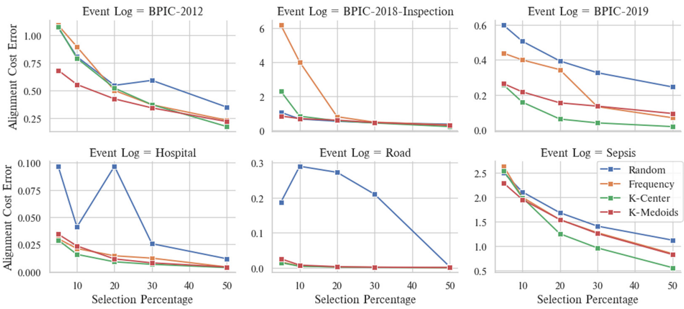
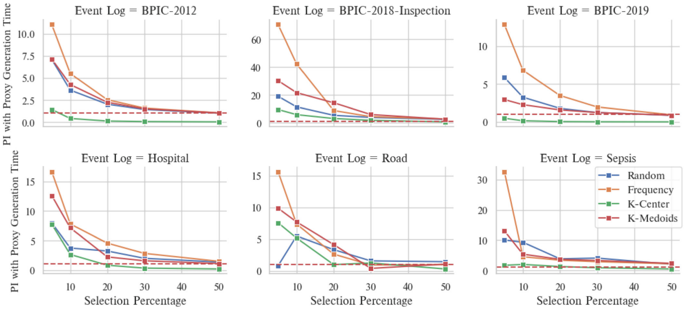


Fig. 4. Effect of increasing the selected percentage of variants on approximated alignments' accuracy for different methods.

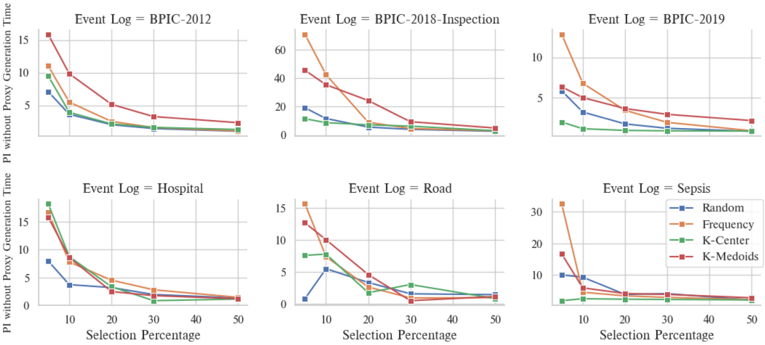
In Fig. 4, we demonstrate the impact of various proxy-set generation methods and trace variant percentages on approximated alignment cost accuracy. K-Center and K-Medoids show promising results, producing proxy-sets that improve accuracy. Additionally, larger proxy-set sizes reduce alignment cost errors, although the influence is constrained for similar event log variants.

Conformance Checking Performance Improvement. We evaluated the time performance of the proxy-set generation methods and observed performance improvements in conformance checking (Fig. 5). To compute the *performance improvement* PI, we divide the conventional alignment computation time by the alignment approximation time, including and excluding proxy-set generation time. Higher PI values indicate greater performance improvement, while a PI value less than 1 indicates additional overhead. The frequency-based method shows the greatest improvement, as it quickly selects variants for proxy-set generation. The Random method has a lower PI value as it may select variants that require more time for alignment computation. Increasing the proxy-set size reduces performance gains. In some cases, the performance does not improve when considering proxy generation time. Thus, it is important to avoid selecting too many traces as a proxy. The proxy generation time for K-Center and K-Medoids methods is notably higher, especially for larger proxy-set sizes. However, if we separate the proxy generation time (as explained in Sect. 1), we can still improve the efficiency of the conformance checking procedure.

Efficiency of the Proposed Lower Bound. Finally, in the last experiment, we compare the lower bound approximation without M', i.e., $\max(0, \min_{\sigma' \in \mathcal{L}(M)} (|\sigma'|) - |\sigma|) + |\sigma_{\downarrow_{\Sigma \setminus \Sigma_M}}|$ and the lower bound that incorporates M', i.,e., $\max_{\sigma' \in \Omega} (z(\sigma', \mathcal{L}(M)) - \delta(\sigma, \sigma')))$.



(a) Performance improvement with consideration of proxy-set generation time. without Proxy Generation Time.png

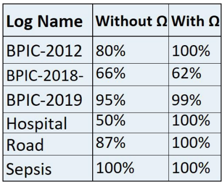


(b) Performance improvement without consideration of proxy selection time.

Fig. 5. Impact of variant selection and proxy methods on performance improvement.

Table 3 presents the percentages of traces with higher values using various bounds. When both methods yield the highest value, we acknowledge both. The findings suggest that, in the majority of situations, employing the proposed lower bound derived from the proxy-set and its alignments is satisfactory. This approach yields more precise approximations of error bounds, leading to more informative evaluations of alignment cost approximations with a given proxy-set.

Table 3. Average of times that lower bounds have the highest value.



Log Name	Without Ω	With Ω
BPIC-2012	80%	100%
BPIC-2018-	66%	62%
BPIC-2019	95%	99%
Hospital	50%	100%
Road	87%	100%
Sepsis	100%	100%

6 Conclusion

In this paper, we proposed a method to obtain bounds on the approximation error when alignment costs are approximated using a subset of traces. Evaluations on real event logs validate the accuracy of different non-optimal instance selection methods and the proposed error estimation technique, and show a reduction in error for the approximated alignment costs while reducing the runtime. We aim to enhance the derivation of k-primal proxy sets to minimize approximation error and improve selection strategies. This will lead to more accurate alignment cost approximations, making our methodology valuable for process mining and alignment analysis.

References

1. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance Checking. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-99414-7
2. Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B.F., van der Aalst, W.M.P.: Alignment based precision checking. In: La Rosa, M., Soffer, P. (eds.) BPM 2012. LNBIP, vol. 132, pp. 137–149. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36285-9_15
3. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: On the role of fitness, precision, generalization and simplicity in process discovery. In: Meersman, R., et al. (eds.) OTM 2012. LNCS, vol. 7565, pp. 305–322. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33606-5_19
4. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.P.: Conformance checking approximation using subset selection and edit distance. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) CAiSE 2020. LNCS, vol. 12127, pp. 234–251. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49435-3_15
5. van Zelst, S.J., Bolt, A., Hassani, M., van Dongen, B.F., van der Aalst, W.M.: Online conformance checking: relating event streams to process models using prefix-alignments. Int. J. Data Sci. Anal. **8**, 1–16 (2017)
6. Bauer, M., van der Aa, H., Weidlich, M.: Sampling and approximation techniques for efficient process conformance checking. Inf. Syst. **104**, 101666 (2020)
7. Awad, A., Raun, K., Weidlich, M.: Efficient approximate conformance checking using trie data structures. In: 3rd International Conference on Process Mining (ICPM), pp. 1–8 (2021)
8. Fani Sani, M., Gonzalez, J.J.G., van Zelst, S.J., van der Aalst, W.M.P.: Alignment approximator: a prom plug-in to approximate conformance statistics. In: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum at BPM 2023, Utrecht, The Netherlands, pp. 102–106 (2023)
9. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **2**(2), 182–192 (2012)
10. van der Aalst, W.M.P.: Decomposing petri nets for process mining: a generic approach. Distrib. Parallel Databases **31**(4), 471–507 (2013)
11. Bauer, M., Senderovich, A., Gal, A., Grunske, L., Weidlich, M.: How much event data is enough? A statistical framework for process discovery. In: Krogstie, J., Reijers, H.A. (eds.) CAiSE 2018. LNCS, vol. 10816, pp. 239–256. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91563-0_15
12. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.P.: Improving the performance of process discovery algorithms by instance selection. Comput. Sci. Inf. Syst. **17**(3), 927–958 (2020)
13. Taymouri, F., Carmona, J.: A recursive paradigm for aligning observed behavior of large structured process models. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 197–214. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_12

14. Padró, L., Carmona, J.: Computation of alignments of business processes through relaxation labeling and local optimal search. Inf. Syst. **104** (2022)
15. Fani Sani, M., Garza Gonzalez, J.J., van Zelst, S.J., van der Aalst, W.M.P.: Conformance checking approximation using simulation. In: 2nd International Conference on Process Mining, ICPM 2020, Padua, Italy, pp. 105–112. IEEE (2020)
16. Park, H., Jun, C.: A simple and fast algorithm for k-medoids clustering. Expert Syst. Appl. **36**(2), 3336–3341 (2009)
17. van der Aalst, W.M.P., van Dongen, B., Günther, C.W., Rozinat, A., Verbeek, E., Weijters, T.: Prom: the process mining toolkit. BPM (Demos) **489**(31) (2009)
18. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs containing infrequent behaviour. In: Lohmann, N., Song, M., Wohed, P. (eds.) BPM 2013. LNBIP, vol. 171, pp. 66–78. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06257-0_6

Repair of Unsound Data-Aware Process Models

Matteo Zavatteri(✉), Davide Bresolin, and Massimiliano de Leoni

Department of Mathematics, University of Padova, Padua, Italy
{`matteo.zavatteri,davide.bresolin,massimiliano.deleoni`}`@unipd.it`

Abstract. Process-aware Information Systems support the enactment of business processes, and rely on a model that prescribes which executions are allowed. As a result, the model needs to be sound for the process to be carried out. Traditionally, soundness has been defined and studied by only focusing on the control-flow. Some works proposed techniques to repair the process model to ensure soundness, ignoring data and decision perspectives. This paper puts forward a technique to repair the data perspective of process models, keeping intact the control flow structure. Processes are modeled by acyclic Data Petri Nets. Our approach repairs the Constraint Graph, a finite symbolic abstraction of the infinite state-space of the underlying Data Petri Net. The changes in the Constraint Graph are then projected back onto the Data Petri Net.

Keywords: Data Petri Net · soundness · business process · model repair

1 Introduction

Process-aware Information Systems (PAISs) are instances of a type of system that supports the execution of processes within an organization. The main advance of PAISs is that they guarantee the compliance of process executions with respect to a process model that is provided as input.

PAISs can only function properly if the input process model is sound. Otherwise, some process executions carried on through a PAIS might remain blocked in a deadlock, or might never be completed. Consider, e.g., a financial institute that grants loans to customers: if the PAIS is configured via an unsound model, some loan applications might remain within the organization without ever being accepted or rejected. This is clearly not desired, because it affects the customer satisfaction and/or hampers the reputation of the organization.

Checking soundness of process models has attracted a lot of attention in the past [1–6]. Recently, Felli et al. [7] studied verification of soundness of DPNs with guards that include comparison of variables, and introduced the constraint graph, on which we also rely for model repair, but did not consider the repair

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 383–395, 2024.
https://doi.org/10.1007/978-3-031-50974-2_29

of unsound models. Later on, the framework was extended to support full arithmetic constraints [8], at the price of loosing decidability of the soundness verification problem, as the underlying constraint graph is not guaranteed to be finite anymore.

Conversely, few research works have focused on automatically repairing process models to ensure soundness, and they do not consider the data perspective, namely process variables and the related guards [9,10]. Indeed, processes manipulate data when being executed, and this data restricts the behavior that the process allows. Repairing the process model while ignoring data may cause the model to remain unsound, as activities and data are intertwined.

This work is the first attempt to repair the data perspective of data-aware process models, keeping intact the control flow structure. Process models are here represented in the form of Data Petri Nets (DPNs). A general algorithm is proposed that can repair acyclic DPNs and that keeps intact the place/transition structure of the network, and that tries to minimize the number of guards that ought to be changed. The underlying assumption is that unsoundness is caused by the data perspective: if the unsoudness were due to the control-flow (i.e. the structure of the Petri-net underneath), this should be fixed beforehand, using techniques such as those described in [9]. We support guards in form of difference constraints, namely $x - y \bowtie k$ where x and y are real variables, k is a constant, and $\bowtie$ can be either $<$ or $\leq$. Unary guards, e.g. $x < k$, can also be supported. We prove that the algorithm always terminates, returning a sound DPN.

Organization. Section 2 introduces the notions of *(i)* systems of difference constraints and their consistency (i.e., satisfiability), and *(ii)* DPNs and soundness. Section 3 reports on our algorithm for repairing DPNs and proves termination, while Sect. 4 draws conclusions and discusses future work.

2 Data Petri Nets: Syntax, Semantics and Verification

This section summarizes the main concepts and formalization of the syntax and semantics of Data Petri nets, which are adapted from Felli et al. [7] to support difference constraints.

Data Petri nets are Petri Nets that are complemented by a set V of variables, whose values are updated through the transition firings. Guards are also associated to transitions, and provide further constraints to have transitions enabled.

Given a variable $v \in V$ we write v^r or v^w to denote that the variable v is, respectively, read or written by an activity in the process, hence we consider two sets V^r and V^w defined as $V^r := \{v^r \mid v \in V\}$ and $V^w := \{v^w \mid v \in V\}$. Intuitively, since an activity of the process may require to read and/or update the value of variables, we use v^r and v^w to respectively denote the variable v before and after the transition is executed. For this reason, we also refer to them as read and written variables, respectively. We omit the superscripts r and w to refer to a variable that can be either read or written.

A *difference constraint* over two real variables x, y has the form $y - x \bowtie k$, where $\bowtie$ is a comparison operator that can be either $<$ or $\leq$ and $k \in \mathbb{R} \cup \{+\infty\}$.

Unary constraints of the form $x \bowtie k$ can be encoded into difference constraints $x - Z \bowtie k$ where Z is a fresh real variable intended to be always set to zero. An equality constraint $y - x = k$ can be encoded into two difference constraints $y - x \leq k$ and $x - y \leq -k$. This also holds for constraints $x = k$ (once rewritten as $x - Z = k$). A system of difference constraints is a set of difference constraints. A system of difference constraint is *consistent* if there exists an assignment of real values to the variables that satisfies all constraints. A consistent system of difference constraints admits a unique *canonical representation* that can be computed via a generalization of Floyd-Warshall algorithm using *difference-bound matrices* to represent the canonical form [11], with time and space complexity $\Theta(n^3)$ where n is the number of variables. We can now formalize DPNs as follows.

Definition 1 (Data Petri Nets). *Let V be a set of variables. Let $\mathcal{C}_V$ be the universe of difference constraints over $V^r \cup V^w \cup \{Z\}$. A Data Petri Net (DPN) $\mathcal{N} = (P, T, F, V, \alpha_I, guard)$ is a Petri net (P, T, F) with additional components describing the additional perspectives of the process model:*

- *V is a finite set of real process variables*
- *$\alpha_I : V \cup \{Z\} \mapsto \mathbb{R}$ is a function defining the initial assignment with $\alpha_I(Z) = 0$*
- *$guard : T \to \mathcal{C}_V$ is a function returning the guard of a transition.*

Conjunctions and disjunctions are not allowed in the guard for simplicity. However, disjunctive guards can be mimicked by having multiple transitions from and to the same places, whereas read-only conjunctive guards can be modeled as "non-interruptible" sequences, as discussed in [7].

Given $t \in T$, as a shorthand we write $read(t) := \{v \in V \mid v \in read(guard(t))\}$, and analogously $write(t)$. Moreover, we assume that a DPN is always associated with an arbitrary initial marking M_I and an arbitrary final marking M_F. When M_F is reached the execution of the process instance ends.

Consider the DPN in Fig. 1a. From the initial marking $M_I = \{p_1\}$, transition t_1 updates the value of `a` to a value greater than `5`. Then, t_2 or t_3 may be executable depending on the current value of `a` being greater or smaller than `10`. Similarly, t_4 can be executed only if the initial value of `b` is smaller than the current value of `a`. The only possible sequence of transitions that reaches the final marking is t_1, t_2, t_4, as $\alpha_I(\mathtt{b}) = \mathtt{10}$. A simplistic analysis that disregards the possible assignments of variables at each step, and thus only considers the control-flow of the net, would instead erroneously conclude that there are no dead transitions and that it is always possible to reach the final marking avoiding deadlocks, i.e., that $\mathcal{N}$ is classically sound [12].

Execution Semantics. By considering the usual semantics for the underlying Petri net together with the guards associated to each of its transitions, we define the resulting execution semantics for DPNs in terms of possible states and possible evolution from a state to the next. Let $\mathcal{N}$ as above be a DPN. Then, the possibly infinite set of states of $\mathcal{N}$ is formed by all pairs (M, α) where M is the marking of the Petri net, that is, a multiset of places from P, and α is an assignment of the variables in V. In any state, zero or more transitions of a

DPN may be able to fire. Firing a transition t updates the marking, reads the variables specified in *read*(t) and selects a new value for those in *write*(t). A DPN $\mathcal{N}$ evolves from state (M,α) to state (M',α') through t if:

- t is enabled and the new marking is M' (denoted $M[t\rangle M'$) according to the Petri net semantics;
- for each $v \in V$, if $v \notin$ *write*(t) then the value of v is unchanged: $\alpha'(v) = \alpha(v)$;
- the guard is satisfied when we assign values to read variables according to α and to written variables according to α'.

We denote a legal transition firing by writing $(M,\alpha) \xrightarrow{t} (M',\alpha')$. We also extend this definition to sequences of legal transition firings (runs), and we write $(M,\alpha) \xrightarrow{*} (M',\alpha')$ if there exists a sequence of legal transition firings $(M,\alpha) \xrightarrow{t^1} \dots \xrightarrow{t^n} (M',\alpha')$. For instance, referring to the simple DPN $\mathcal{N}$ in Fig. 1a, a possible legal transition firing from the initial state is $(\{p_1\},\{\alpha_I(\mathtt{a}) = \mathtt{0}, \alpha_I(\mathtt{b}) = \mathtt{10})\})$ $\xrightarrow{t_1} (\{p_2\},\{\alpha(\mathtt{a}) = \mathtt{7}, \alpha(\mathtt{b}) = \mathtt{10})\})$.

Finally, recall that a Petri net (P,T,F) is unbounded when there exists a place $p \in P$ such that there exists no finite bound $k \in \mathbb{N}$ so that $M(p) \leq k$ for all reachable markings M. The notion trivially extends to DPNs: a DPN is unbounded when there exists a place $p \in P$ so that there is no finite bound k such that $M(p) \leq k$ for all reachable states (M,α).

Data-Aware Soundness. We recall here the lifting of the standard notion of soundness [12] to the data-aware setting of DPNs, as illustrated in [7]. The resulting notion is data-aware, as it requires not only to quantify over the reachable markings of the net, but also on the SV assignments for the variables.

Given a DPN $\mathcal{N}$, in what follows we write $(M,\alpha) \xrightarrow{*} (M',\alpha')$ to mean that there exists a trace σ such that $(M,\alpha) \xrightarrow{\sigma} (M',\alpha')$ or that $(M,\alpha) = (M',\alpha')$. Also, given two markings M' and M'' of a DPN $\mathcal{N}$, we write $M'' \geq M'$ iff for all $p \in P$ of $\mathcal{N}$ we have $M''(p) \geq M'(p)$, and we write $M'' > M'$ iff $M'' \geq M'$ and there exists $p \in P$ s.t. $M''(p) > M'(p)$.

Definition 2 (Data-aware soundness [7]). *A DPN with initial marking M_I and final marking M_F is data-aware sound iff all the following properties hold.*

P1. *For every reachable state (M,α), $\exists \alpha_F.\ (M,\alpha) \xrightarrow{*} (M_F,\alpha_F)$*
P2. *For every reachable state (M,α), $M \geq M_F \Rightarrow (M = M_F)$*
P3. *For every transition $t \in T$, there exist two reachable states (M_1,α_1) and (M_2,α_2) such that $(M_1,\alpha_1) \xrightarrow{t} (M_2,\alpha_2)$.*

The first condition imposes that it is *always* possible to reach the final marking by suitably choosing a continuation of the current run (i.e., legal transition firings). The second condition captures that the final marking is always reached in a "clean" way, i.e., without having tokens in the rest of the net. The third condition verifies the absence of dead transitions, where a transition is considered dead if there is no way to enable it through the execution of the process. For

Algorithm 1: Procedure for computing $C \oplus c$.

```
if $c = y^r - x^r \bowtie k$ then                              ▷ $c$ is a read-only constraint
    $C' := C \cup \{y - x \bowtie k\}$
    return CanonicalForm($C'$)
else                                                           ▷ $c$ writes some variable
    if $write(c) = \{x\}$ then
        $C' := C' \cup \{y - x^w \bowtie k\}$
    else if $write(c) = \{y\}$ then
        $C' := C' \cup \{y^w - x \bowtie k\}$
    else                                                       ▷ $write(c) = \{x, y\}$
        $C' := C' \cup \{y^w - x^w \bowtie k\}$
    $C'$ := CanonicalForm($C'$)
    $C' := C' \setminus \{x' - y' \bowtie' k' \mid x' \in write(c)$ or $y' \in write(c)\}$
    Rename all occurrences of $x^w$ to $x$ and all occurrences of $y^w$ to $y$ in $C'$
    return $C'$                                                ▷ $C'$ is already in canonical form
```

instance, $P1$ is false for the DPN in Fig. 1a: when transition t_1 assigns a value not greater than `10` to `a` there exists no run from there which marks M_F.

DPN Soundness Verification via Constraint Graph. This paper bases the soundness verification and the repair algorithm on the structure of the *constraint graph*, introduced by Felli et al. [7]. A constraint graph is a finite symbolic abstraction of the (possibly infinite) traces of a DPN, that allows to verify soundness of acyclic DPNs and to identify the changes needed to repair the DPN, if found unsound. A constraint graph is characterized by a state-transition structure where each node is associated with a marking and an abstraction of the data, given as a canonical representation of a system of difference constraints.

Given a set of difference constraints C, and a constraint c, we now define the procedure of computing the new constraint set C' resulting from the addition of a constraint c to C so that C' is uniquely determined, denoted $C' = C \oplus c$. This is shown in Algorithm 1, where we maintain the same notation as before, so that x, y, z can be either constants or read variables in V^r. It requires a `CanonicalForm` procedure that, given a set C' of difference constraints as input, returns the minimal constraint network derived from C' using the generalized Floyd-Warshall algorithm described above. When given an unsatisfiable constraint set, we assume `CanonicalForm` to return a null value, so that it can be used also to signal inconsistency of a set of difference constraints.

Definition 3 (Constraint Graph of a DPN). *Let $\mathcal{N} = (P, T, F, V, \alpha_I, guard)$ be a DPN, $\mathcal{M}$ be the set of markings of $\mathcal{N}$, M_I the initial marking and $\mathcal{C}^*$ be the set of consistent constraint sets on V. The* <u>*constraint graph*</u> *$CG_{\mathcal{N}}$ of $\mathcal{N}$ is a tuple $\langle N, n_0, A \rangle$ consisting of:*

- *$N \subseteq \mathcal{M} \times \mathcal{C}^*$ is the set of* nodes *of the graph.*
- *$n_0 = (M_I, C_0)$ is the initial node where C_0 is the canonical form of the system of difference constraints $\bigcup_{v \in V} \{v = \alpha_I(v)\}$;*
- *$A \subset N \times (T \cup \tau_T) \times N$ is the set of arcs such that:*

- *a transition* $((M, C), t, (M', C'))$*, where* $t \in T$*, is in* A *iff* $M[t\rangle M'$ *and* $C' = C \oplus guard(t)$ *is consistent;*
- *a transition* $((M, C), \tau_t, (M, C''))$*, where* $\tau_t \in \tau_T$*, is in* A *iff* $write(t) = \emptyset$*,* $\exists M'$ *s.t.* $M[t\rangle M'$*, and* $C'' = C \oplus \neg guard(t)$ *is consistent.*

3 The Repair of Data Petri Nets

In this section we describe the repair algorithm. As stressed in the introduction, we assume that the underlying Petri Net (i.e., without the data dimension) of the DPN to repair is sound and acyclic, and we focus on the repair of the data perspective only. Informally, given an acyclic DPN $\mathcal{N}$, we want to build a sound DPN $\mathcal{N}'$ by changing only the constants and comparison operators in the guards. Moreover, we look for a repair that limits the number of changes.

Definition 4. *Let* $\mathcal{N} = (P, T, F, V, \alpha_I, guard)$*. A repair of* $\mathcal{N}$ *is a DPN* $\mathcal{N}' = (P', T', F', V', \alpha'_I, guard')$ *meeting the following three conditions:*

1. $(P, T, F, V, \alpha_I) = (P', T', F', V', \alpha'_I)$*;*
2. *for each transition* $t \in T$ *with* $guard(t) = y - x \bowtie k$*,* $guard'(t) = y - x \bowtie' k'$*;*
3. $\mathcal{N}'$ *is data-aware sound.*

The cost of the repair $\mathcal{N}'$ *is the number of guards in* $\mathcal{N}'$ *that differ from* $\mathcal{N}$*.*

We show the repair algorithm for acyclic DPNs in Algorithm 2. The main procedure is `DPNRepair` which implements a Breadth First Search in a Dijkstra fashion to explore the space of possible Data Petri Nets. The function uses a priority queue to store the partial solutions that are being explored. At each iteration the algorithm extracts a DPN with minimum priority, builds the corresponding constraint graph, and, if such a graph is unsound, it calls `FixDead` and `FixMissing`. `FixDead` processes dead nodes, that is, nodes that violate condition P1 of the definition of data-aware soundness (Definition 2). For each dead node, it identifies a set of transitions to operate on. For each transition, it computes a new guard from the systems of difference constraints belonging to the dead node. After that, it pushes in the queue the resulting new DPN.

`FixMissing` processes transitions that do not appear in the constraint graph, namely those that violate condition P3 of Definition 2. For each transition, it identifies the set of nodes in the constraint graph from which the transition might fire, and another set of transitions to operate on. For each transition in this set, it computes a new guard and pushes the resulting DPN in the queue.

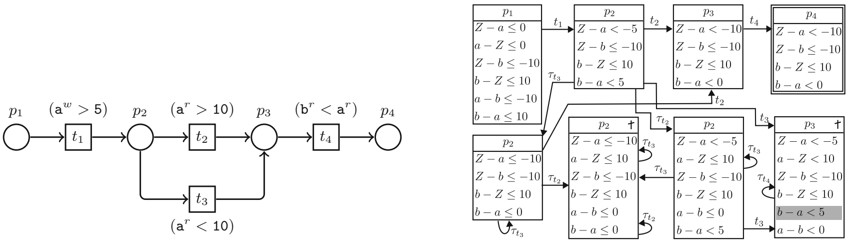
Dead nodes can be treated in two possible ways. Either we enable a transition that is currently prevented to fire from that node, or we prevent the execution to reach that node. The first case is managed by `ForwardRepair`, whereas the second one by `BackwardRepair`.

Consider Fig. 1. The initial DPN (Fig. 1a) is data-aware unsound. Indeed, its constraint graph (Fig. 1b) reveals the presence of two dead nodes (identified by a cross). Consider the rightmost dead node. That node has marking $M = \{p_3\}$

Algorithm 2: An algorithm to repair a DPN.

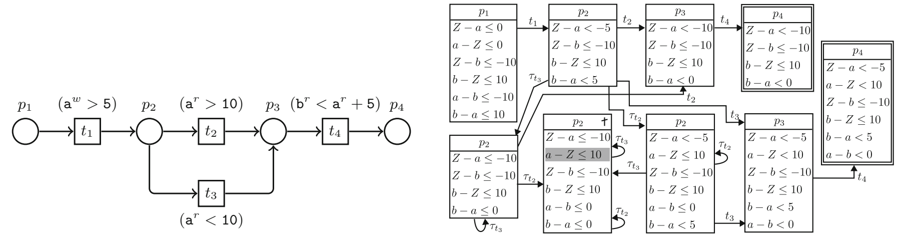
```
DPNRepair ($\mathcal{N}$)                                   ▷ Assume that $\mathcal{N}$ has global visibility
    Let $\mathcal{Q}$ be an empty priority queue.                          ▷ global variable
    Push the input DPN $\mathcal{N}$ in $\mathcal{Q}$ with priority 0.
    Let $\mathcal{N}'$ be an empty DPN.                         ▷ the one to return eventually
    while true do
        Extract from $\mathcal{Q}$ a DPN with minimum priority $p$ and save it in $\mathcal{N}'$
        Let $CG_{\mathcal{N}'}$ be the constraint graph of $\mathcal{N}'$
        if $CG_{\mathcal{N}'}$ is data-aware sound then break       ▷ we are done, exit the while loop
        FixDead($\mathcal{N}'$, $CG_{\mathcal{N}'}$)
        FixMissing($\mathcal{N}'$, $CG_{\mathcal{N}'}$)
    return $\mathcal{N}'$                           ▷ the repaired DPN (last extracted from $\mathcal{Q}$)
UpdateQ ($\mathcal{N}'$)                                     ▷ used in FixDead, FixMissing
    if $\mathcal{N}'$ has not been visited yet then
        Let $p$ be the number of transitions of $\mathcal{N}'$ with a different guard in $\mathcal{N}$.
        Push $\mathcal{N}'$ in $\mathcal{Q}$ with priority $p$.
FixDead ($\mathcal{N}'$, $CG_{\mathcal{N}'}$)
    foreach dead node $(M, C)$ in $CG_{\mathcal{N}'}$ do
        Let $FW := \{t \in T \mid M[t\rangle M'$ for some marking $M'\}$.
        foreach $t \in FW$ do ForwardRepair($\mathcal{N}', t, C$)
        Let $BW$ be the set of non-silent transitions in all paths $(M_0, C_0) \rightsquigarrow (M, C)$.
        foreach $t \in BW$ do BackwardRepair($\mathcal{N}', t, C$)
ForwardRepair ($\mathcal{N}', t, C$)                     ▷ "replace with the same constraint of $C$"
    Let $\mathcal{N}'' := (P, T, F, V, \alpha_I, guard'')$ be a copy of $\mathcal{N}'$.
    Let $y - x \bowtie k$ be the guard of $t$.
    Let $y - x \bowtie' k'$ be the corresponding constraint in $C$.
    $guard''(t) := y - x \bowtie' k'$
    UpdateQ($\mathcal{N}''$)
BackwardRepair ($\mathcal{N}', t, C$)                ▷ "replace with the opposite constraint of $C$"
    Let $\mathcal{N}'' := (P, T, F, V, \alpha_I, guard')$ be a copy of $\mathcal{N}'$.
    Let $y - x \bowtie k$ be the guard of $t$.
    if $x - y \bowtie' k'$ in $C$ is such that $k' \neq \infty$ then
        if $\bowtie'$ is $\leq$ then $guard''(t) := y - x < -k'$
        else $guard''(t) := y - x \leq -k'$
        UpdateQ($\mathcal{N}''$)
FixMissing ($\mathcal{N}'$, $CG_{\mathcal{N}'}$)
    Let Missing be the set of missing transitions in $CG_{\mathcal{N}'}$.
    foreach $t \in Missing$ do
        Let $Nodes := \{(M, C) \mid \exists M'.M[t\rangle M'\}$
        foreach $(M, C) \in Nodes$ do
            ForwardRepair($\mathcal{N}', t, C$)
            Let $BW$ be the set of non-silent transitions in all paths $(M_0, C_0) \rightsquigarrow (M, C)$.
            foreach $t \in BW$ do BackForwardRepair($\mathcal{N}', t$)
BackForwardRepair ($\mathcal{N}', t$)                             ▷ "make the guard true"
    Let $\mathcal{N}'' := (P, T, F, V, \alpha_I, guard')$ be a copy of $\mathcal{N}'$.
    Let $y - x \bowtie k$ be the guard of $t$.
    $guard''(t) := y - x \leq \infty$
    UpdateQ($\mathcal{N}''$)
```

and thus the only transition that might fire from the node is t_4. However, t_4 is not enabled because the set of constraints in the node augmented with the guard of t_4 is inconsistent. The current guard of t_4 is $\mathtt{b}^r < \mathtt{a}^r$, that in difference constraint form, regardless of read and write operations, is $\mathtt{b} - \mathtt{a} < 0$. The same



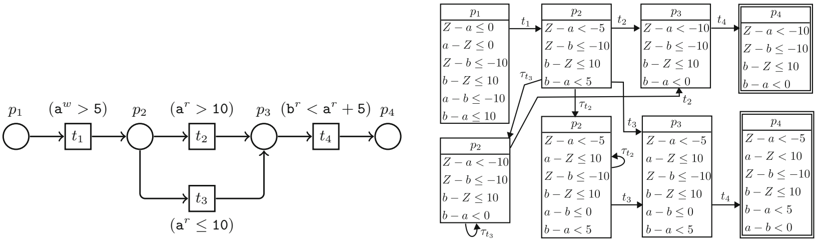
(a) Example of DPN $\mathcal{N}$ [7]. $M_I = \{p_1\}$, $M_F = \{p_4\}$, $\alpha_I(\mathsf{a}) = 0$, $\alpha_I(\mathsf{b}) = 10$.

(b) $CG_{\mathcal{N}}$ showing unsoundness of $\mathcal{N}$ (dead nodes).



(c) Modification 1: $\mathcal{N}'$ is $\mathcal{N}$ with the guard of t_4 modified as $\mathsf{b}^r < \mathsf{a}^r + 5$.

(d) $CG_{\mathcal{N}'}$ showing unsoundness of $\mathcal{N}'$ (dead nodes).

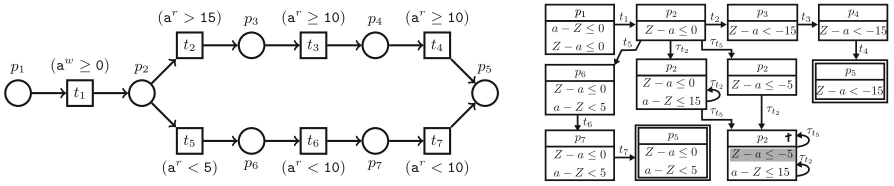


(e) Modification 2: $\mathcal{N}''$ is $\mathcal{N}'$ with the guard of t_3 modified as $\mathsf{a}^r \leq 10$.

(f) $CG_{\mathcal{N}''}$ showing soundness of $\mathcal{N}''$.

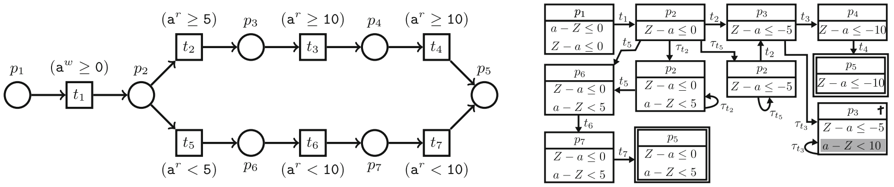
Fig. 1. Repair algorithm using `ForwardRepair` only.

constraint in the set of constraints C of the dead node is $\mathsf{b} - \mathsf{a} < 5$. This says that if the guard of t_4 was $\mathsf{b}^r < \mathsf{a}^r + 5$, then t_4 would be able to fire. Therefore, such constraint becomes the new guard of t_4. The resulting DPN $\mathcal{N}'$ (Fig. 1d) is computed by $\texttt{ForwardRepair}(\mathcal{N}, t_4, C)$ by applying this modification and it is pushed in the queue with priority 1, since the number of guards differing from the original DPN is currently one. The algorithm proceeds with a breadth-first search by exploring the other modifications given by the subsequent calls to `ForwardRepair` and `BackwardRepair` and pushing other candidate DPNs in the queue. When $\mathcal{N}'$ is extracted from the queue, the algorithm builds its constraint graph (Fig. 1d) discovering another dead node. By proceeding similarly, we can push a new DPN $\mathcal{N}''$ (Fig. 1e) in the queue, obtained by modifying the current guard of t_3 which is $\mathsf{a}^r < 10$ (i.e., $\mathsf{a} - Z < 10$) to $\mathsf{a}^r \leq 10$ (i.e., $\mathsf{a} - Z \leq 10$). When $\mathcal{N}''$ is extracted from the queue the algorithm verifies that it is data-aware



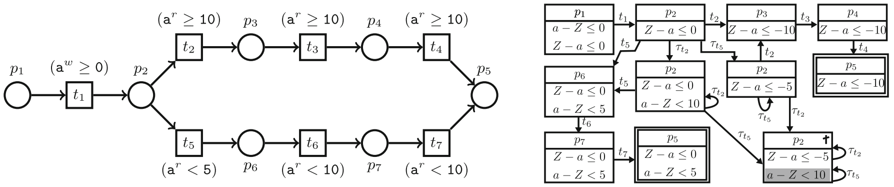
(a) Example of DPN $\mathcal{N}$. $M_I = \{p_1\}$, $M_F = \{p_5\}$, and $\alpha_I(\mathsf{a}) = \mathsf{0}$.

(b) $CG_{\mathcal{N}}$ showing unsoundness of $\mathcal{N}$ (dead node).



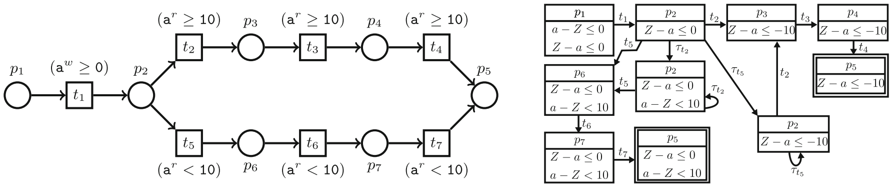
(c) Modification 1: $\mathcal{N}'$ is $\mathcal{N}$ with the guard of t_2 modified as $Z - a \leq -5$.

(d) $CG_{\mathcal{N}'}$ showing unsoundness of $\mathcal{N}'$ (dead nodes).



(e) Modification 2: $\mathcal{N}''$ is $\mathcal{N}'$ with the guard of t_2 modified as $Z - a \leq -10$.

(f) $CG_{\mathcal{N}''}$ showing unsoundness of $\mathcal{N}''$ (dead nodes).



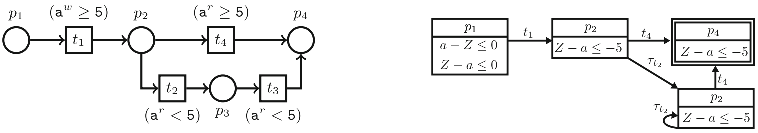
(g) Modification 3: $\mathcal{N}'''$ is $\mathcal{N}''$ with the guard of t_5 modified as $a - Z < 10$.

(h) $CG_{\mathcal{N}'''}$ showing soundness of $\mathcal{N}'''$.

Fig. 2. Repair algorithm using `ForwardRepair` and `BackwardRepair` only.

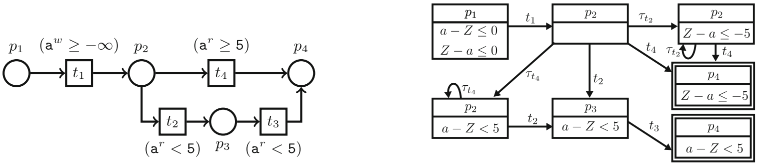
sound and returns $\mathcal{N}''$ as the repaired DPN. This example shows a possible path to a solution that uses only `ForwardRepair`.

Yet, there are cases in which solutions with smaller costs can be found if we also use `BackwardRepair`. For instance, consider $\mathcal{N}$ in Fig. 2a. In the constraint graph of $\mathcal{N}$ (Fig. 2b) there is a dead node. Consider the application of `ForwardRepair`$(\mathcal{N}, t_2, C)$, where C is the set of difference constraints of the dead node. The result is $\mathcal{N}'$, shown in Fig. 2c, still contains a dead node. One possible path to continue is to apply `ForwardRepair` to t_3 and then to t_4 (since changing



(a) Example of DPN $\mathcal{N}$. $M_I = \{p_1\}$, $M_F = \{p_3\}$, and $\alpha_I(\mathtt{a}) = \mathtt{0}$.

(b) $CG_{\mathcal{N}}$ showing unsoundness of $\mathcal{N}$ (missing transitions).



(c) Modification 1: $\mathcal{N}'$ is $\mathcal{N}$ with the guard of t_1 modified as $\mathtt{a}^w \geq -\infty$.

(d) $CG_{\mathcal{N}'}$ showing soundness of $\mathcal{N}'$.

Fig. 3. One-shot repair using `BackForwardRepair` only.

t_3 generates a dead node from which t_4 cannot be executed) to obtain a solution of cost 3. However, a solution with smaller cost can be found by following a different path, that proceeds by applying `BackwardRepair` to t_2, since t_2 is a transition belonging to a path that can reach the dead node from the initial one. The current guard of t_2 is $\mathtt{a}^r \geq \mathtt{5}$ (i.e., $Z - \mathtt{a} \leq -\mathtt{5}$). To prevent the execution to reach the dead node we can restrict the guard of t_2 so that the constraint system of the dead node becomes inconsistent with this new guard. To do so, we need to modify the guard of t_2 to $\mathtt{a}^r \geq \mathtt{10}$ (i.e., $Z - \mathtt{a} \leq -\mathtt{10}$). This modification generates a negative weight cycle (i.e., a certificate of inconsistency) with the constraint $\mathtt{a} - Z < \mathtt{10}$ contained in the system of constraints of the dead node. Hence, that node will no longer exist in the constraint graph. Figure 2e shows the DPN $\mathcal{N}''$ obtained by this last modification, whose constraint graph (Fig. 2f) still contains a dead node. If we apply `ForwardRepair` to t_5 we obtain the repaired DPN $\mathcal{N}'''$ in Fig. 2g with a total cost of 2. Thus, there are cases in which, by operating several times on the same transition, we can obtain repairs with smaller costs.

A constraint graph can also be unsound because of missing transitions. This situation does not necessarily imply the existence of dead nodes and it is therefore handled by the function `FixMissing`. Missing transitions can be treated in two possible ways. Either we enable the missing transitions to fire from the nodes where the marking allows them but the data does not, or we remove constraints by operating on transitions along the paths from the initial node to the node under analysis. The former case is still handled by `ForwardRepair`, whereas the latter is handled by `BackForwardRepair` that mixes ideas from `ForwardRepair` and `BackwardRepair`. Once again, we proceed by discussing a concrete example. Consider the DPN $\mathcal{N}$ in Fig. 3a and its constraint graph in Fig. 3b. There are no dead nodes in the constraint graph. However, t_2 (and thus t_3 occurring after t_2) are missing in the constraint graph. A solution of cost 1 can be found by applying `BackForwardRepair` to t_1 since t_1 is in the path that goes from the initial node to a node with the marking $M = \{p_2\}$ from which t_2 can fire in the underlying "dataless" Petri Net. `BackForwardRepair` replaces the guard of t_1 with $\mathtt{a}^w \geq -\infty$ (i.e., $Z - \mathtt{a}^w \leq \infty$) by obtaining the data-aware sound $\mathcal{N}'$ (Fig. 3c and Fig. 3d). Despite `BackForwardRepair` basically sets a guard to "true", we recall that the same guard might be later be processed by `BackwardRepair` in order to be restricted adequately.

Theorem 1. *Let $\mathcal{N}$ be an acyclic DPN where the underlying dataless Petri net is sound. Algorithm 2 terminates on $\mathcal{N}$ by returning a data-aware sound DPN.*

Proof. Let $\mathcal{N}$ be an acyclic DPN where the underlying Petri net is sound. First, notice that by neglecting self-loop silent transitions, the constraint graph of an acyclic DPN is a DAG. Since the set of transitions T is finite and the underlying DPN is bounded, the branching factor of each node in the constraint graph is bounded by $2 \cdot |T|$ (i.e., all transitions t plus the corresponding silent transitions τ_t), and the longest path from the initial node to a final one cannot exceed $|T|$. As a result, there are at most $(2 \cdot |T|)^{|T|}$ nodes in the constraint graph. Among the possible sequences of modifications that Algorithm 2 can follow there always exists one that uses `ForwardRepair` only. Such a sequence can always be explored since (i) all constraint graphs built along the way are finite, (ii) the possible modifications applied to a constraint graph are finite, and (iii) such modifications are explored following a BFS strategy. Consider therefore the sequence that calls only `ForwardRepair`. Every time a new DPN is generated the guard of a transition t is replaced with some constraint in the constraint system of some node: either a dead node if `ForwardRepair` is called inside `FixDead`, or a node from which a missing transition can be fired if `ForwardRepair` is called inside `FixMissing`. In the former case (fixing a dead node), some paths in the constraint graph starting from the initial node and ending with the silent transition τ_t are removed. Also, such paths can never be introduced again by subsequent modifications: if t is processed again, the current guard $y - x \bowtie k$ is replaced by a weaker guard $y - x \bowtie' k'$. In the latter case (fixing a missing transition), some paths in the constraint graph are extended with the transition t. By the same monotonicity argument on subsequent modifications of t, such extended paths can never be removed in subsequent applications of `ForwardRepair`. Since the number of paths in a constraint graph is finite, the sequence of `ForwardRepair` reaches a sound DPN in a finite number of steps. □

4 Conclusions

This paper focuses on repairing data-aware process models to ensure soundness. We use DPNs as modelling formalism, and employ difference constraints over real variables as transition guards. We defined a general algorithm that can repair acyclic DPNs, keeps intact the place/transition structure of the network, and tries to minimize the number of guards that ought to be changed. The algorithm exploits the full power of difference constraint to build a repaired network with as few changes as possible. We rely on the canonical form of systems of difference constraints to compute the modifications on the guards of transitions. We proved that the algorithm terminates, returning a repaired DPN.

As future work, we plan to implement the algorithm and experimentally evaluate the efficiency with models of increasing complexity. We aim to investigate the optimality of the algorithm, and to extend the algorithm to support repair of cyclic DPNs. Finally, we are currently assuming that every guard change is equivalent to repair the model: in reality, process modellers and analysts may favor certain changes over others. This requires to define a cost framework where certain guard changes come at lower costs, thus being preferable.

Acknowledgements. This work was supported by the Project funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.5 - Call for tender No. 3277 of 30 dicembre 2021 of Italian Ministry of University and Research funded by the European Union - NextGenerationEU; Project code: ECS00000043, Concession Decree No. 1058 of June 23, 2022, CUP C43C22000340006, Project title "iNEST: Interconnected Nord-Est Innovation Ecosystem".

This work was also supported by INdAM, GNCS 2023, Project "Analisi simbolica e numerica di sistemi ciberfisici".

References

1. Morimoto, S.: A survey of formal verification for business process modeling. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008. LNCS, vol. 5102, pp. 514–522. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69387-1_58
2. Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F.: A formal approach to modeling and verification of business process collaborations. Sci. Comput. Program. **166**, 35–70 (2018)
3. Sidorova, N., Stahl, C., Trčka, N.: Soundness verification for conceptual workflow nets with data: early detection of errors with the most precision possible. Inf. Syst. **36**(7), 1026–1043 (2011)
4. Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics and analysis of DMN decision tables. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 217–233. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_13
5. Batoulis, K., Weske, M.: Soundness of decision-aware business processes. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNBIP, vol. 297, pp. 106–124. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65015-9_7
6. Batoulis, K., Haarmann, S., Weske, M.: Various notions of soundness for decision-aware business processes. In: Mayr, H.C., Guizzardi, G., Ma, H., Pastor, O. (eds.) ER 2017. LNCS, vol. 10650, pp. 403–418. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69904-2_31
7. Felli, P., de Leoni, M., Montali, M.: Soundness verification of data-aware process models with variable-to-variable conditions. Fund. Inform. **182**(1), 1–29 (2021)
8. Felli, P., Montali, M., Winkler, S.: Soundness of data-aware processes with arithmetic conditions. In: Franch, X., Poels, G., Gailly, F., Snoeck, M. (eds.) CAiSE 2022. LNCS, vol. 13295, pp. 389–406. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-07472-1_23

9. Gambini, M., La Rosa, M., Migliorini, S., Ter Hofstede, A.H.M.: Automated error correction of business process models. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 148–165. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23059-2_14
10. Armas Cervantes, A., van Beest, N.R.T.P., La Rosa, M., Dumas, M., García-Bañuelos, L.: Interactive and incremental business process model repair. In: Panetto, H., et al. (eds.) OTM 2017. LNCS, vol. 10573, pp. 53–74. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69462-7_5
11. Dill, D.L.: Timing assumptions and verification of finite-state concurrent systems. In: Sifakis, J. (ed.) CAV 1989. LNCS, vol. 407, pp. 197–212. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-52148-8_17
12. van der Aalst, W.M.P.: The application of petri nets to workflow management. J. Circ. Syst. Comput. **8**(1), 21–66 (1998)

Non-Automata Based Conformance Checking of Declarative Process Specifications Based on ASP

Isabelle Kuhlmann[1(✉)], Carl Corea[2], and John Grant[3]

[1] Artificial Intelligence Group, University of Hagen, Hagen, Germany
isabelle.kuhlmann@fernuni-hagen.de
[2] Institute for IS Research, University of Koblenz, Koblenz, Germany
ccorea@uni-koblenz.de
[3] Department of Computer Science and UMIACS, University of Maryland, College Park, USA
grant@cs.umd.edu

Abstract. We investigate the use of Answer Set Programming (ASP) for the problem of conformance checking of LTL-based Declarative Process Specifications. In particular, we propose ASP solutions that are independent of automata. That is: in related works, the semantics of the declarative process specifications are often captured by means of finite state automata. This means that for conformance checking, the constraints of the specification first have to be transformed into a corresponding automata representation, which introduces a computational burden. In this work, we present a new ASP-based approach which encodes the constraint semantics directly and therefore can be used to check conformance without the need of performing automata operations. We implement our approach and perform experiments with real-life datasets, comparing our approach to a selection of state-of-the-art approaches. Our experiments show that our approach can outperform existing approaches in some cases. Furthermore, our approach can easily be extended to check whether the considered constraint sets are satisfiable (i.e., consistent).

Keywords: Conformance Checking · Declarative Process Specifications · Answer Set Programming

1 Introduction

Within the field of Process Mining [1], the problem of *conformance checking* is considered a major task [1,2]. This relates to verifying the conformance, or compliance, of observed process *behavior* w.r.t. a given process *specification*. In this context, typically, some form of process specification is assumed to be known. Then, actually observed process behavior, usually in the form of event logs (consisting of individual *traces*), can be evaluated against the process specification

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 396–408, 2024.
https://doi.org/10.1007/978-3-031-50974-2_30

to verify which traces, i.e., which behaviors, conform to the specification. Essentially, a trace models atomic activities over a linear sequence of time points, with the intuition that in every instant of time, exactly one activity holds. For conformance checking, it can then be verified whether this assignment of truth values over the time sequence is compliant with the specification. In this work, we focus on *declarative process specifications* based on linear temporal logic (LTL).

In related works [3], conformance checking has been approached by representing the logical constraints as *finite state automata* (FSAs). This allows for various approaches: for example, to "replay" the traces on these automata. While such approaches are already very valuable, they require the transformation of the constraints into an automaton representation in advance [4]. As such operations introduce a computational burden [5], in this work, we aim to present a novel approach for conformance checking in declarative process specifications without the need of transforming constraints into automata. Also, there are other existing approaches that are "optimized" for the Declare modeling language [6], meaning that they rely on tailored verification logic for a predefined set of constraint types. Instead, we aim to implement LTL semantics directly within ASP, to allow for a support of arbitrary LTL-formulas.

A directly related work is by CHIARIELLO ET AL. [3], which also presents an approach for ASP-based conformance checking. However, those authors follow the described approach of encoding constraints as FSAs first. To the best of our knowledge, our work is the first to present an ASP-based solution which does not use *automata representations*, and which allows the verification of *arbitrary* LTL formulas out-of-the-box. Our contributions are as follows:

- We present an approach for conformance checking of declarative process specifications based on ASP (Sect. 3). Importantly, our approach encodes the Declare semantics in ASP natively, meaning that no translation to FSAs is necessary. Furthermore, this allows us to consider arbitrary LTL formulas.
- We implement our approach and perform runtime experiments with real-life datasets (Sect. 4). Here, we show that our approach can outperform existing approaches in some cases.
- Additionally, we show how our approach can be extended to *satisfiability checking* (i.e., checking whether there exists at least one trace that satisfies the specification, where otherwise the specification would be inconsistent). As we will show, our "automata-free" approach has major advantages over existing satisfiability checking approaches (which usually verify inconsistency by checking whether the automata product of all constraints is empty), as our approach does not require the construction of the actual automata product.

2 Preliminaries

The following sections provide fundamental definitions from the fields of linear temporal logic, Declare, and answer set programming. Moreover, we describe some existing conformance checking methods from the literature.

2.1 Linear Temporal Logic on Fixed Traces, Declare

In this work, we build on a specific variant of $\mathrm{LTL_f}$ from a previous work [7], named linear temporal logic on *fixed* traces ($\mathrm{LTL_{ff}}$). For this, we consider a linear sequence of states $t_0, \ldots, t_m$, where every t_i is the state at instant i. We assume that $m > 1$ to avoid the trivial case.

The syntax of $\mathrm{LTL_{ff}}$ is the same as the syntax of LTL and $\mathrm{LTL_f}$ [8]. Formulas are built from a set of propositional symbols At and are closed under the Boolean connectives, the unary operator $\mathbf{X}$ (*next*), and the binary operator $\mathbf{U}$ (*until*). Formally, any formula φ of $\mathrm{LTL_{ff}}$ is built using the grammar rule

$$\varphi ::= a|(\neg\varphi)|(\varphi_1 \wedge \varphi_2)|(\varphi_1 \vee \varphi_2)|(\mathbf{X}\varphi)|(\varphi_1 \mathbf{U} \varphi_2).$$

with $a \in \mathsf{At}$. Intuitively, $\mathbf{X}\varphi$ denotes that φ will hold at the next state and $\varphi_1 \mathbf{U} \varphi_2$ denotes that φ_1 will hold until the state when φ_2 holds. Let $d(\varphi) \in \mathbb{N}$ denote the maximal number of nested temporal operators in φ.[1]

Let $\top$ be any tautology and $\bot$ any contradiction. From the basic operators, some useful abbreviations can be derived, including $\mathbf{F}\varphi$ (defined as $\top\mathbf{U}\varphi$), which denotes that φ will hold (eventually) in the future and $\mathbf{G}\varphi$ (defined as $\neg\mathbf{F}\neg\varphi$), which denotes that φ will hold for all following states.

An $\mathrm{LTL_{ff}}$-interpretation $\hat{\omega}$ w.r.t. At is a function mapping each state and proposition to 0 or 1, meaning that $\hat{\omega}(t, a) = 1$ if a is assigned 1 (true) in state t.[2] The satisfaction of a formula φ by an interpretation $\hat{\omega}$, denoted by $\hat{\omega} \models \varphi$, is defined via $\hat{\omega} \models \varphi \Leftrightarrow \hat{\omega}, t_0 \models \varphi$, where $\hat{\omega}, t_i \models \varphi$ for any interpretation $\hat{\omega}$ as above and for every $t_i \in \{t_0, \ldots, t_m\}$ is inductively defined as follows:

$$\begin{aligned}
&\hat{\omega}, t_i \models a \text{ iff } \hat{\omega}(t_i, a) = 1 \text{ for } a \in \mathsf{At}\\
&\hat{\omega}, t_i \models \neg\varphi \text{ iff } \hat{\omega}, t_i \not\models \varphi\\
&\hat{\omega}, t_i \models \varphi_1 \wedge \varphi_2 \text{ iff } \hat{\omega}, t_i \models \varphi_1 \text{ and } \hat{\omega}, t_i \models \varphi_2\\
&\hat{\omega}, t_i \models \varphi_1 \vee \varphi_2 \text{ iff } \hat{\omega}, t_i \models \varphi_1 \text{ or } \hat{\omega}, t_i \models \varphi_2\\
&\hat{\omega}, t_i \models \mathbf{X}\varphi \text{ iff } i < m \text{ and } \hat{\omega}, t_{i+1} \models \varphi\\
&\hat{\omega}, t_i \models \varphi_1 \mathbf{U} \varphi_2 \text{ iff } \hat{\omega}, t_j \models \varphi_2 \text{ for some } j \in \{i+1, \ldots, m\}\\
&\qquad\qquad\qquad\qquad \text{and } \hat{\omega}, t_k \models \varphi_1 \text{ for all } k \in \{i, \ldots, j-1\}
\end{aligned}$$

An interpretation $\hat{\omega}$ satisfies a set of formulas K iff $\hat{\omega} \models \varphi$ for all $\varphi \in K$. A set K is consistent iff there exists $\hat{\omega}$ such that $\hat{\omega} \models K$. Define $X \models Y$ for (sets of) formulas X and Y if $\hat{\omega} \models X$ implies $\hat{\omega} \models Y$ for all $\hat{\omega}$.

Using $\mathrm{LTL_{ff}}$, it is possible to define higher-level modeling languages. Typically, these provide a set of predefined constraint *types* (also referred to as templates), which can be instantiated with atomic propositions of an alphabet

---

[1] $d(\varphi)$ is inductively defined via $d(a) = 0$ for $a \in \mathsf{At}$, $d(\neg\varphi) = d(\varphi)$, $d(\varphi_1 \wedge \varphi_2) = d(\varphi_1 \vee \varphi_2) = \max\{d(\varphi_1), d(\varphi_2)\}$, $d(\mathbf{X}\varphi) = 1 + d(\varphi)$, and $d(\varphi_1 \mathbf{U} \varphi_2) = 1 + \max\{d(\varphi_1), d(\varphi_2)\}$.

[2] Recall that we assume time of a fixed length $t_0, \ldots, t_m$ and interpretations only vary in what is true at each state.

At. This is useful to allow more non-technical experts to work with temporal constraints. To avoid confusion, we will refer to such instantiated formulas as *constraints*. Also, in this work, we will consider as an example the modeling language Declare [6]. Some Declare constraints used in this work are $\textsc{Init}$(a) ($\Leftrightarrow$ a), $\textsc{Response}$(a,b) ($\Leftrightarrow \mathbf{G}(a \rightarrow \mathbf{F}b)$) and $\textsc{ChainResponse}$(a,b) ($\Leftrightarrow \mathbf{G}(a \rightarrow \mathbf{X}b)$). For an overview of further Declare constraint types, we refer the reader to [4]. As Declare constraints can be re-written into $\mathrm{LTL_{ff}}$ (see above), we can check whether an interpretation $\hat{\omega}$ satisfies a set of Declare constraints K as before.

Importantly, in this work, we do not restrict ourselves to a predefined set of constraint types for conformance checking. Rather, our approach and implementation allows us to verify arbitrary $\mathrm{LTL_{ff}}$ formulas. This is a novel advantage over some existing approaches which are sometimes "hard-coded" in the sense that they rely on constraint-type dependent checking logic, e.g., $\textsc{Response}$(a,b) can be fulfilled differently than $\textsc{ChainResponse}$(a,b) (meaning that such approaches cannot cater to arbitrary LTL-formulas out-of-the-box). We will further differentiate our approach from related approaches in the following.

2.2 Conformance Checking and Related Work

In the scope of the introduced formalism, we assume a trace in an event log to be a finite sequence s over a fixed alphabet At (where each atom corresponds to an atomic activity).[3] For example, if $\mathsf{At} = \{a, b\}$, possible traces could be a, b, aa, ab, and so forth. Note that this is referred to as a *control-flow-only*-perspective [3], as we do not consider other attributes at this point. For each trace s, we denote the ith element of s as s^i, e.g., given the trace $s = ab$, we have that $s^0 = a$ and $s^1 = b$. We define an event log E as a multi-set of such traces. Then, for each trace s, we can obtain a corresponding interpretation $\hat{\omega}^s$ as follows:

1. Fix the length of $\hat{\omega}^s$ as the length of s.
2. For each ith item of s, set the truth value $\hat{\omega}^s(t_i, s^i)$ to 1.

Following the BPM literature, we assume that at every instant of time only one activity holds [8]. Hence, in addition to the above procedure, we assume that all truth values not explicitly set to 1 are 0 (false).

Example 1. Consider a set of atomic activities $\mathsf{At} = \{a, b\}$ and a trace $s = abb$. We then get the following interpretation $\hat{\omega}^s$ following the above steps:

$$\begin{array}{lll} \hat{\omega}(t_0, a) = 1 & \hat{\omega}(t_1, a) = 0 & \hat{\omega}(t_2, a) = 0 \\ \hat{\omega}(t_0, b) = 0 & \hat{\omega}(t_1, b) = 1 & \hat{\omega}(t_2, b) = 1 \end{array}$$

We refer to an interpretation $\hat{\omega}^s$ derived from this procedure as the "corresponding" interpretation to s for simplicity. Then, we define the conformance checking problem as follows.

---

[3] We will denote all sequences $s = \langle s_0, \ldots, s_m \rangle \in \mathsf{At}^*$ as $s_0 \ldots s_m$ for readability, e.g., abc instead of $\langle a, b, c \rangle$.

Definition 1 (Conformance Checking). *Let K be a finite set of LTL$_{\mathrm{ff}}$ formulas and s a trace with its corresponding interpretation $\hat{\omega}^s$. Then the conformance checking problem is the function* $c(K, s) = \begin{cases} true & if\ \hat{\omega}^s \models K \\ false & otherwise \end{cases}$

In some related works on conformance checking, the formulas are first transformed into (language-equivalent) deterministic finite state automata. For this, an automata can be constructed that accepts all words which are in the language of a formula.[4] Due to space limitations, we refer the reader to [4] for a formal definition of FSAs and automata products.

From a recent literature review by Dunzer et al. [10], we see many works for conformance checking, e.g., [3,11–15]. In the following, we want to highlight two concepts, namely replay-based approaches and approaches using FSAs.

- **Replay-based approaches.** In general, replay-based approaches check the conformance of a trace by iterating, or *replaying*, the entire trace (in a linear fashion). Then, it is checked at every position where the current atomic activity is the activation of a constraint. If so, it is verified if the constraint is *fullfilled* later in the trace. In a similar fashion, it can also be verified for "negative" constraints such as NotResponse(a,b) that after the activation a occurs, the reaction b must not occur (otherwise, the constraint is permanently violated). Two representative approaches of this group are [11,12]. A recent implementation of such an approach has also been made available in the *Declare4Py* library [13]. Note that a potential disadvantage of such approaches is that they are sometimes "hard-coded" for Declare, in the sense that they rely on constraint-type dependent checking logic (e.g., Response(a,b) can be fullfilled differently than ChainResponse(a,b)). The result is that such approaches cannot verify arbitrary formulas. On the contrary, in our approach, this is possible, as we present a direct ASP encoding of LTL$_{\mathrm{ff}}$ semantics.
- **Approaches using FSAs.** Many approaches also transform the LTL-based constraints into FSAs and then exploit these FSAs in some way.[5] This has the advantage that there is no need for constraint-type dependent checking-logic (see above). A potential disadvantage of such an approach is however that the LTL-based constraints need to be transformed into FSAs beforehand. Also, such approaches still essentially rely on replay, which might have scalability problems [3]. A recent approach to overcome this problem has been introduced in [3], where the authors propose an ASP-based checking mechanism. In theory, this allows conformance checking without any need of replay. However, a potential limitation we see in [3] is that the constraints are again encoded as FSAs, meaning that again FSA transformations must be performed. In this work, we therefore propose a new type of ASP encoding which implements the LTL semantics directly. This has the advantage of exploiting the amenities of ASP results, but without the "detour" of representing semantics via FSAs.

---

[4] See for example [9] for an algorithm for this problem.

[5] We refer the reader to [4] to further details on FSAs.

Two further acknowledgements are in order. First, the idea of using logic programming for conformance checking is not new, cf. e.g. [16]; however, newer ASP solvers may obtain faster results, as also suggested recently in [3,17]. Second, we explicitly acknowledge that there are other approaches for conformance checking, in particular, approaches using alignments between traces and FSAs [14,15,18]. As this is an initial work, it is our clearly confined scope to benchmark only against the introduced *replay-based* approaches, and we leave a comparison against *alignment-based* approaches for future work.

2.3 Answer Set Programming

Answer set programming (ASP) [19,20] is a declarative problem solving approach which has already been used for conformance checking [3], as well as for solving problems related to conformance checking, namely in the field of inconsistency measurement [21–23]. In ASP, the objective is to represent a given problem in a logical format (an *extended logic program*) such that the models of this representation (the answer sets) express solutions of the initial problem.

An extended logic program consists of *rules* of the form

$$r = a_0 \texttt{ :- } a_1, \ldots, a_n, \texttt{ not } a_{n+1}, \ldots, \texttt{ not } a_m. \tag{1}$$

with a_i $(0 \leq i \leq n \leq m)$ being atoms, and "`not`" indicating default negation [24]. Further, "`:-`" can be interpreted as "if", a "," as "and", and the end of a rule is marked by a ".". An atom is a predicate $p(v_1, \ldots, v_k)$ with $k \geq 0$, where each $v_1, \ldots, v_k$ is either a constant or a variable. Following the Clingo [25] syntax, we represent constants by strings starting with a lowercase letter, and variables by strings starting with an uppercase letter[6]. If an atom, a rule, or a program does not contain any variables, it is referred to as *ground*. Moreover, the arity k of a predicate p is expressed as p/k.

An ASP rule r (as illustrated in Eq. 1) is comprised of a *head* $\mathsf{head}(r) = a_0$ and a *body* $\mathsf{body}(r) = \{a_1, \ldots, a_n, \texttt{ not } a_{n+1}, \ldots, \texttt{ not } a_m\}$. If $\mathsf{head}(r)$ is empty, r is called a *constraint*, and if $\mathsf{body}(r) = \emptyset$, r is called a *fact*. We further divide the elements of $\mathsf{body}(r)$ into $\mathsf{body}^+(r) = \{a_1, \ldots, a_n\}$ and $\mathsf{body}^-(r) = \{a_{n+1}, \ldots, a_m\}$. A set X of ground atoms is a *model* of a ground logic program P if for all $r \in P$, $\mathsf{head}(r) \in X$ whenever $\mathsf{body}^-(r) \cap X = \emptyset$ and $\mathsf{body}^+(r) \subseteq X$. Moreover, we define the *reduct* [26] of a program P w.r.t. X as $P^X = \{\mathsf{head}(r) \texttt{ :- } \mathsf{body}^+(r) \mid \mathsf{body}^-(r) \cap X = \emptyset, r \in P\}$. If X is a subset-minimal model of P^X, then X is called an *answer set* of P.

Further language concepts we make use of are *cardinality constraints* and *aggregates*. A cardinality constraint is of the form $l\{a_1; \ldots; a_n\}u$, with l constituting a lower bound, and u an upper bound. Intuitively, this can be read as "at least l, and at most u of the atoms in $\{a_1, \ldots, a_n\}$ must be included in the answer set". Aggregates are built-in functions. We utilize the `#count` aggregate, which allows for counting the number of ground instances.

---

[6] Note that we also use *anonymous variables*. Such variables, denoted by "_", do not recur within the rule at hand.

3 Conformance Checking Based on Native ASP Encodings

In the following, we describe an approach of encoding the conformance checking problem in ASP, and how to modify it for satisfiability checking.

3.1 Conformance Checking

Given a set of constraints K and a single trace s, we propose an encoding that yields an answer set if the trace satisfies K. Otherwise, no answer set is derived.

We encode each atom $x \in \mathsf{At}(K)$ as `atom(`x`)`, and each formula $\varphi \in K$ as `kbElement(`φ`)`. Moreover, a conjunction $\varphi = \varphi_1 \wedge \varphi_2$ is represented as `conjunction(`$\varphi, \varphi_1, \varphi_2$`)`. In the same manner we model disjunctions and negations as `disjunction/3` and `negation/2`. Formulas with the LTL operators **X**, **U**, **G**, and **F** are represented as `next/2`, `until/3`, `globally/2`, and `eventually/2`. A formula φ which is an atom x is represented as `formulaIsAtom(`φ, x`)`.

In order to represent a given trace s in ASP, we first define $|s|$ states, where the final state $m = |s| - 1$ is `finalState(`m`)`. The states $\{t_0, \ldots, t_m\}$ are then modeled by adding the rule "`state(0..M) :- finalState(M).`" to the encoding. Further, let s^i be the ith item of s (with $i \in \{0, \ldots, m\}$) . We model that s^i is true in state t_i by adding `true(`s^i, i`)`. For example, for $s = abb$ we have

`true(`a`,0).` `true(`b`,1).` `true(`b`,2).`

We can now proceed to define the evaluation of all the logical operators (see Listing 1 for an overview). First, we model that a formula consisting of an individual atom x is true if x itself is true (line 1). Conjunction, disjunction and negation are modelled in the usual way (lines 2–4). With regard to the LTL-specific operators, a formula $\mathbf{X}\varphi$ is true in state t_i if φ is true in state t_{i+1} and $i < m$ (line 5). A formula $\varphi_1 \mathbf{U} \varphi_2$ is true in state t_i if there is a state t_j ($i < j \leq m$) such that φ_1 is true for all $t_i, \ldots, t_{j-1}$ and φ_2 is true in state t_j (line 6). We additionally encode the operators **G** and **F** (lines 7–8). Note that we do not need to explicitly model the *false* cases. We simply ensure that no answer set can be derived if a formula is not satisfied. This is achieved by adding an integrity constraint (line 9) which enforces that all constraints must be satisfied in order for an answer set to be derived.

```
1 true(F,S):- formulaIsAtom(F,A), state(S), true(A,S).
2 true(F,S):- conjunction(F,G,H), state(S), true(G,S),
      true(H,S).
3 true(F,S):- disjunction(F,G,H), state(S), 1{true(G,S);
      true(H,S)}.
4 true(F,S):- negation(F,G), state(S), not true(G,S).
5 true(F,S_i):- next(F,G), state(S_i), S_j=S_i+1, S_i<M,
      finalState(M), true(G,S_j).
```

```
true(F,S_i):- until(F,G,H), state(S_i), state(S_j), S_j>
    S_i, S_j<=M, finalState(M), X{true(G,S): state(S), S>=
    S_i, S<S_j}X, X=S_j-S_i, true(H,S_j).
true(F,S_i):- globally(F,G), state(S_i), X{true(G,S):
    state(S), S>S_i}X, finalState(M), X=M-S_i.
true(F,S_i):- eventually(F,G), state(S_i), true(G,S_j),
    state(S_j), S_j>S_i.
:- not true(F,0), kbElement(F), state(0).
```

Listing 1. Encoding of logical entailment.

The presented encoding allows us to verify conformance for a set of constraints w.r.t. a given trace. As can be seen, the logic of the temporal operators is encoded "directly", which brings two main advantages: a) constraints need not be transformed into automata representations, b) it is possible to directly consider arbitrary LTL_{ff} formulas. Regarding the latter case, note that for the related (ASP-based) approach in [3], it would be possible to consider arbitrary constraints as well; however, a corresponding FSA has to be manually created beforehand. In our approach, all formulas can be used without any prior effort.

In addition to the previously described approach which considers all constraints in K at once (i.e., within a single ASP solver call), we also propose a slightly modified version in which we check each constraint separately. Thus, for each $\varphi \in K$ we create an individual answer set program. This further allows us to implement an *early stopping* criterion: as soon as we get the result that a constraint $\varphi \in K$ is unsatisfiable w.r.t. a given trace s, we know that s is unsatisfiable. The early stopping criterion is useful to obtain conformance results in a binary manner, i.e., whether the trace is conformant to the entire specification.

3.2 Satisfiability Checking

The above approach for conformance checking can be extended to solve the problem of *satisfiability checking* in LTL_{ff}. For this, we can adapt our ASP encoding such that it returns an answer set if there exists a trace s of length m, and a corresponding interpretation $\hat{\omega}^s$, s.t. $\hat{\omega}^s \models K$, for a given specification K. If K is unsatisfiable, no answer set is returned. Note that in the satisfiable case, we can directly extract a "witness trace" from the answer set that satisfies K.

In the case of satisfiability checking, we consider only the constraints, so a rule is added that *guesses* a trace by selecting exactly one atom per state (see line 1 in Listing 2) and these atoms are set to *true* (line 2). To allow constraints to be vacuously satisfied, we represent each atom x in the signature of each constraint `atomInFormula(`x, φ`)`. We check if a constraint φ is vacuously satisfied by computing the size of the intersection of $\mathsf{At}(\varphi)$ and the elements in s (line 3). If the intersection is empty, we set the constraint to *true* in t_0 (line 4).

```
1{inTrace(A,S): atom(A)}1:- state(S).
true(A,S):- inTrace(A,S)}.
intersectionSize(F,X):- kbElement(F), X = #count{A:
    atomInFormula(A,F), inTrace(A,_)}.
true(F,0):- kbElement(F), intersectionSize(F,0).
```

Listing 2. Additional rules required for satisfiability checking.

Satisfiability checking is also performed over the temporal logic itself. This gives a major advantage over automata-based checking approaches where the automata product of all individual constraint FSAs must be computed (to check if the product is empty). This is known to be barely tractable, as computing the product may scale exponentially [5]. On the contrary, our approach allows us to check satisfiability directly over the LTL semantics. As we will show, this leads to very efficient satisfiability computations, often in seconds.

4 Experiments

We now continue to benchmark our proposed encodings.

4.1 Data Sets and Setup

For our experiments, we considered the real-life data sets of the Busines Process Intelligence Challenge (BPIC). From the provided BPIC event logs, it is possible to mine Declare specifications with various tools, and consequently, to check the conformance of these specifications w.r.t. the corresponding event logs.

For the experiments, we considered all available BPIC data sets (BPIC 2012–2020), shown in Table 1.[7] From the data sets, we first mined a set of Declare constraints using the Declare4Py library [13]. We used a standard support parameter of 70%. For each data set, the number of constraints mined this way is shown in Table 1. Due to hardware limitations, we were not able to successfully mine a Declare specification from the (large) BPIC 2018 data set, so we could not consider BPIC 2018 for the experiment. After the mining, we performed conformance checking with a number of approaches:

- **ASP_{AC}.** The main version of our ASP approach that considers *all* constraints within one ASP solver call.
- **ASP_{SC}.** The version of our ASP approach which considers each constraint individually (*single*).
- **ASP_{SC}+ES.** As ASP_{SC}, with *early stopping* enabled (see Sect. 3.1).
- **ASP-FSA (Related Work).** The automata-based ASP approach from [3].
- **Declare4Py (Related Work).** The replay-based approach provided by [13].

---

[7] For some years, e.g., 2020, there were multiple sub data sets - as these were similar, we only present the respective first sub data set of those years due to space reasons.

The goal of our experimental analysis is to compare the solving times of the different approaches. All experiments were run on a computer with 32 GB RAM and an AMD Ryzen 7 PRO 5850U CPU with a time limit of 10 minutes per data set. Our approach was implemented in C++[8], using Clingo 5.4.0.

4.2 Conformance Checking Results

In general, the evaluation results (presented in Table 1) are mixed, but there are some interesting observations. First, there were in fact two cases (BPIC-12 and BPIC-20) in which our (non-automata-based) approach (ASP_{AC}) outperformed the related (automata-based) ASP approach (ASP-FSA). Especially for BPIC-20, our approach significantly outperformed both ASP-FSA as well as Declare4Py (around 6 to 10-times faster). However, there was one case where our performance was roughly the same as the related ASP approach (BPIC-13), and some cases where our approach could not compute a result within the timecap (BPIC-15, BPIC-17, BPIC-19). In this regard, it is noteworthy that the replay-based Python implementation performed very well. It could compute most results w.r.t. the considered data sets, and was often the fastest approach. While we fully acknowledge this good performance, we recall however that our approach was significantly faster for BPIC-20. Also, for BPIC-19 (which had by far the greatest number of traces), no approach, including Declare4Py, could compute a conformance result for all constraints within the timecap. Here, our ASP_{SC}+ES (early stopping) approach was the only approach which was able to generate some form of conformance insight. This shows that this "binary" form of conformance checking (just checking whether a trace conforms to the specification) can be beneficial in settings where it is difficult to compute the

Table 1. Overview of the results. "ASP_{AC}" considers all constraints; "ASP_{SC}" considers single constraints where "ES" means "early stopping enabled". "ASP-FSA" refers the automata-based ASP approach by [3]. "Declare4Py" is the replay-based approach presented in [13]. Runtimes are in seconds. "–" indicates the result could not be computed within the time limit. The best result for each data set is in boldface.

BPIC	$\|K\|$	#Traces	This work			Related work	
			ASP_{AC}	ASP_{SC}	ASP_{SC}+ ES	ASP-FSA	Declare4Py
'12	25	1050	22.33	34.29	19.88	23.56	**3.14**
'13	21	1 487	1.38	2.29	**0.34**	0.38	0.51
'15	1322	1 199	–	–	42.31	310.40	**22.74**
'16	474	1 050	267.89	423.28	**1.08**	31.00	9.64
'17	343	31 509	–	–	–	–	**252.10**
'19	157	251 734	–	–	**149.25**	–	–
'20	139	10 500	3.56	7.47	**0.19**	33.12	18.64

---

[8] The implementation is available under https://e.feu.de/cc-asp.

full conformance results. While ASP_{SC}+ES is viewed separately, as it solves a different, in this case, binary, problem, overall fast runtimes can be noted.

To further understand the performance of the considered approaches, more experiments with synthetic data sets would be needed, which we aim to address in future works. Yet, as this is an early work, we see it at least as promising that our approach could outperform both related approaches for one dataset each.

4.3 Satisfiability Checking Results

For cases with extremely high number of traces (s.t. conformance checking is challenging), it may be a good investment to conduct a satisfiability check as a preceding step, to see if the constraints are even consistent, i.e., that there can even be traces that satisfy the specification. Using the encoding shown in Sect. 3.2, we computed satisfiability for the BPIC data sets. Note that for our logic, a trace length has to be fixed, and the runtime of our approach scales with the trace length, as more states also mean more possible solutions for the ASP solver. For the experiments, we set this length to 50 to test a larger configuration. This also ensures that we exclude the trivial case of an empty trace that may vacuously satisfy the specification. In general, it suffices to consider a trace length of the max. depth of all the constraints (cf. Prop. 3 in [7]).

The results, presented in Table 2, show that a satisfiability check can be performed mostly within seconds, and within the timecap for BPIC 2015 (with over 1300 constraints). These runtimes impose a great benefit over approaches that would need to compute an automata product over these constraint sets.

Table 2. Results for satisfiability checking.

BPIC	'12	'13	'15	'16	'17	'19	'20
Solving time (s)	0.08	0.08	392.31	46.47	20.15	6.02	4.23

5 Outlook and Conclusion

We presented an ASP-based approach for the problem of conformance checking which—in contrast to alternative approaches from the literature—does not rely on FSA representations. In addition, we showed how to modify this approach to solve the problem of satisfiability checking. Although the results are mixed, our approach outperformed two methods from the literature w.r.t. the BPIC 2020 dataset. It also outperformed the automata-based ASP approach (ASP-FSA) on BPIC 2012. Another advantage of our approach is the fact that is can be used on arbitrary LTL formulas, and is therefore more flexible than approaches that are "hard-coded", e.g., for Declare. As an additional contribution, the LTL encodings presented in this work can be re-used by other researchers for related

ASP projects. In future works, we aim to obtain a more detailed analysis of properties that lead to certain approaches performing superior (or inferior) to others with the aid of synthetic data.

Allthough this goes beyond the scope of traditional conformance checking, some recent works have also investigated means to enumerate possible continuations of an incomplete trace (s.t. the continuation is compliant with a specification) [27]. Here, our ASP-based approach could also be extended in future works to enumerate such continuations via answer sets.

References

1. van der Aalst, W.M.: Process mining: a 360 degree overview. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 3–34. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_1
2. Carmona, J., van Dongen, B., Weidlich, M.: Conformance checking: foundations, milestones and challenges. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 155–190. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_5
3. Chiariello, F., Maggi, F.M., Patrizi, F.: Asp-based declarative process mining. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 5539–5547 (2022)
4. Di Ciccio, C., Montali, M.: Declarative process specifications: reasoning, discovery, monitoring. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 108–152. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_4
5. Di Ciccio, C., Maggi, F.M., Montali, M., Mendling, J.: Resolving inconsistencies and redundancies in decl. process models. Inf. Syst. **64**, 425–446 (2017)
6. Pesic, M., Schonenberg, H., Van der Aalst, W.M.: Declare: full support for loosely-structured processes. In: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), p. 287. IEEE (2007)
7. Corea, C., Grant, J., Thimm, M.: Measuring inconsistency in declarative process specifications. In: Di Ciccio, C., Dijkman, R., del Rio Ortega, A., Rinderle-Ma, S. (eds.) Business Process Management. BPM 2022. LNCS, vol. 13420, pp. 289–306. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16103-2_20
8. Giacomo, G.D., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: Proceedings of the 23rd IJCAI, pp. 854–860, Beijing, AAAI (2013)
9. Westergaard, M., Stahl, C., Reijers, H.A.: Unconstrainedminer: efficient discovery of generalized declarative process models. BPM Center Rep. **207**, 28 (2013)
10. Dunzer, S., Stierle, M., Matzner, M., Baier, S.: Conformance checking: a state-of-the-art literature review. In: Proceedings of the 11th International Conference on Subject-Oriented Business Process Management, pp. 1–10 (2019)
11. Burattin, A., Maggi, F.M., van der Aalst, W.M., Sperduti, A.: Techniques for a posteriori analysis of declarative processes. In: 2012 IEEE 16th International Enterprise Distributed Object Computing Conference, pp. 41–50. IEEE (2012)
12. Burattin, A., Maggi, F.M., Sperduti, A.: Conformance checking based on MP declarative process models. Expert Syst. Appl. **65**, 194–211 (2016)
13. Donadello, I., Riva, F., Maggi, F.M., Shikhizada, A.: Declare4py: a python library for declarative process mining. BPM'22 Demo Track **3216**, 117–121 (2022). https://github.com/francxx96/declare4py

14. de Leoni, M., Maggi, F.M., van der Aalst, W.M.P.: Aligning event logs and declarative process models for conformance checking. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 82–97. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32885-5_6
15. van Dongen, B.F., De Smedt, J., Di Ciccio, C., Mendling, J.: Conformance checking of mixed-paradigm process models. Inf. Syst. **102**, 101685 (2021). https://www.sciencedirect.com/science/article/pii/S0306437920301344
16. Montali, M., Pesic, M., Aalst, W.M.V.D., Chesani, F., Mello, P., Storari, S.: Declarative specification and verification of service choreographiess. ACM Trans. Web (TWEB) **4**(1), 1–62 (2010)
17. Ielo, A., Ricca, F., Pontieri, L.: Declarative mining of business processes via ASP. In: Workshop on Process Management in the AI Era (PMAI 2022) co-located with IJCAI-ECAI '22, Wien, Austria, pp. 105–108. CEUR-WS.org (2022)
18. De Giacomo, G., Maggi, F.M., Marrella, A., Patrizi, F.: On the disruptive effectiveness of automated planning for LTLf-based trace alignment. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
19. Brewka, G., Eiter, T., Truszczynski, M.: Answer set programming at a glance. Commun. ACM **54**(12), 92–103 (2011)
20. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Answer Set Solving in Practice. In: Synthesis Lectures on AI and Machine Learning, vol. 6, no. 3, pp. 1–238 (2012). https://doi.org/10.1007/978-3-031-01561-8
21. Kuhlmann, I., Thimm, M.: An algorithm for the contension inconsistency measure using reductions to answer set programming. In: Davis, J., Tabia, K. (eds.) SUM 2020. LNCS (LNAI), vol. 12322, pp. 289–296. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58449-8_23
22. Kuhlmann, I., Thimm, M.: Algorithms for inconsistency measurement using answer set programming. In: NMR 2021, pp. 159–168 (2021)
23. Kuhlmann, I., Gessler, A., Laszlo, V., Thimm, M.: A comparison of ASP-based and SAT-based algorithms for the contension inconsistency measure. In: Dupin de Saint-Cyr, F., Ozturk-Escoffier, M., Potyka, N. (eds.) Scalable Uncertainty Management. SUM 2022. LNCS, vol. 13562, pp. 139–153. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-18843-5_10
24. Reiter, R.: A logic for default reasoning (1980)
25. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Wanko, P.: Theory solving made easy with Clingo 5. In: Technical Communications of ICLP, pp. 2:1–2:15. OASICS, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016)
26. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings of the ICLP/SLP, pp. 1070–1080, MIT Press (1988)
27. Käppel, M., Schützenmeier, N., Schönig, S., Ackermann, L., Jablonski, S.: Logic based look-ahead for the execution of multi-perspective declarative processes. In: Reinhartz-Berger, I., Zdravkovic, J., Gulden, J., Schmidt, R. (eds.) BPMDS/EMMSAD -2019. LNBIP, vol. 352, pp. 53–68. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20618-5_4

Equivalence of Data Petri Nets with Arithmetic

Marco Montali and Sarah Winkler(✉)

Free University of Bozen-Bolzano, Bolzano, Italy
{montali,winkler}@inf.unibz.it

Abstract. Data Petri nets (DPNs) with arithmetic have gained popularity as a model for data-aware processes, thanks to their ability to balance simplicity with expressiveness and because they can be automatically discovered from event logs. While model checking techniques for DPNs have been studied, there are analysis tasks highly relevant for BPM that are beyond these methods. We focus here on process equivalence and process refinement with respect to language and configuration spaces; such comparisons are important in the context of process repair and discovery. To solve these tasks, we propose an approach for bounded DPNs based on *constraint graphs*, which are faithful abstractions of the reachable state space. Though the considered verification tasks are undecidable in general, we show that our method is a decision procedure for large classes of DPNs relevant in practice.

Keywords: data-aware processes · data Petri nets · process equivalence · process refinement

1 Introduction

Within the growing area of data-aware processes, Data Petri nets (DPNs) with arithmetic data have recently gained increasing popularity thanks to their ability to balance simplicity with expressiveness. DPNs can also be mined automatically [5,17], but automatic mining techniques typically come without any correctness guarantees. However, the complex interplay between the control structure and data makes it hard to check whether DPNs satisfy properties of interest; indeed, all non-trivial verification tasks are undecidable. While linear- and branching-time model checking procedures for DPNs were developed [11–13,18], many analysis tasks relevant in BPM go beyond these techniques. Here we focus on checking equivalence and refinement of processes, which is an important task in many contexts [1,6]: to match an organization-specific model to a reference model, to relate an automatically mined model to a normative one by domain experts, or to compare a refined version of a process model with the original one.

This work was partially funded by the UNIBZ project ADAPTERS, and the PRIN MIUR project PINPOINT Prot. 2020FNEB27.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 409–421, 2024.
https://doi.org/10.1007/978-3-031-50974-2_31

Example 1. The DPN in Fig. 1 models a management process for road fines by the Italian police [20], where assignments with right-hand side *?* indicate a non-deterministic write operation.

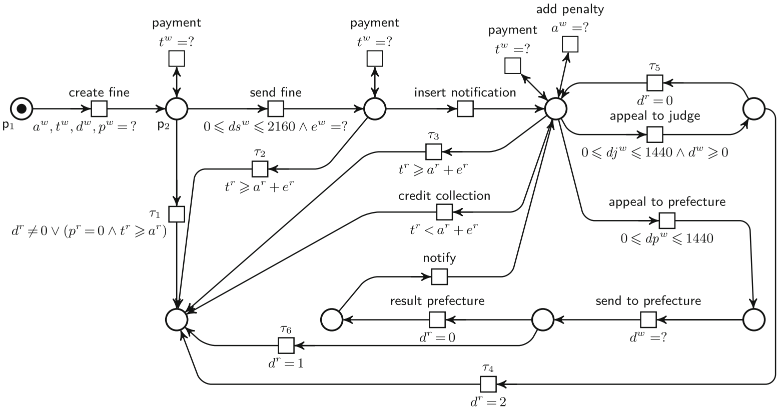


Fig. 1. Data Petri net for road fine management process

This normative process model was designed by domain experts, but other versions of this process were discovered by automatic techniques [19]. This raises a number of questions that are important in the context of process discovery and repair: Do the DPNs admit the same set of configurations and data values? Are all process runs of one model also possible in the other? Do the possible data values in a final state coincide?

To answer such questions, we consider in this paper different notions of behavioural refinement and equivalence for DPNs, comparing markings, configurations, or language, and we propose techniques to check them. Since even state reachability is undecidable for DPNs [12, Rem. 2.6], it does not come as a surpise that these verification tasks are in general undecidable as well.

In this paper, we thus impose two restrictions on DPNs to make verification decidable: (1) To tame the control flow perspective, we assume that the Petri nets underlying DPNs are *bounded*. (2) To tame the data perspective, we assume that the DPNs have a *finite constraint graph*. Constraint graphs (CGs) are symbolic abstractions of the reachable state space that are used for data-aware soundness and model checking of DPNs [10–13]. In general, CGs are infinite. However, it was shown that *finite* constraint graphs can in fact be computed for a wide range of DPNs from the literature. These include DPNs where all constraints are variable-to-variable/constant comparisons, as produced by automatic guard discovery techniques [5,17], or *bounded lookback* DPNs whose behavior depends only on a bounded amount of information from the process run [4,12]. It was shown that these classes comprise almost all DPNs in the literature [13], also e.g. the process of Example 1. Notably, we do *not* assume that DPNs are acyclic.

The contributions of this paper can be summarized as follows: (1) We show how natural notions of equivalence and refinement of DPNs with respect to markings, configurations, and language can be checked based on constraint graphs. (2) Our technique is a decision procedure for bounded DPNs where finite constraint graphs exist, which proves decidability of process refinement/equivalence for such DPNs. These include DPNs where all guards are variable-to-variable/-constant comparisons, and DPNs with bounded lookback. (3) If equivalence or refinement does not hold, counterexamples that distinguish the two processes can be computed by our approach.

Related work. For process models without data, a variety of comparison techniques were developed. An early first approach for process equivalence was presented in [1]. A basic taxonomy of similarity measures was proposed in [6], distinguishing similarity based on either *element labels*, *structure*, or *behaviour*. For the first kind, schema and ontology matching techniques are used [8]. For structural similarity measures, graph matching algorithms were studied [7]. Our approach falls within the class of behavioural similarity, and to the best of our knowledge, no respective approaches exist to compare data-aware processes. However, while most works on process comparison are quantitative (i.e., they quantify process similarity with respect to some measure [16,21], this paper is purely qualitative, in the sense that our techniques check process equivalence or refinement, but the difference between models is not quantified.

2 Background

In this section we summarize some background on constraints, DPNs and data-aware dynamic systems as process models, as well as constraint graphs.

We assume a set of *process variables* V, each of which is associated with a sort from the set $\Sigma = \{\mathtt{int}, \mathtt{rat}\}$ with associated domains integers $\mathcal{D}(\mathtt{int}) = \mathbb{Z}$ and rationals $\mathcal{D}(\mathtt{rat}) = \mathbb{Q}$. For instance, in Example 1 the set of process variables is $V = \{a, d, dj, dp, ds, p, t\}$, where a and t are of sort $\mathtt{rat}$ and the others of sort $\mathtt{int}$. For $\sigma \in \Sigma$, V_σ denotes the subset of variables in V of type σ. To manipulate variables, we consider linear arithmetic expressions c, called *constraints*:

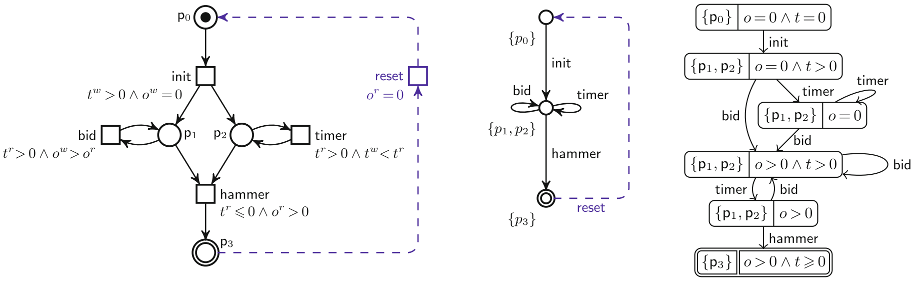
$$\begin{aligned} &c := n \geqslant n \mid n \neq n \mid n = n \mid r \geqslant r \mid r > r \mid r \neq r \mid r = r \mid c \wedge c \\ &n := v_i \mid k \mid n + n \mid -n \qquad\qquad r := v_r \mid q \mid r + r \mid -r \end{aligned}$$

where $v_i \in V_{\mathtt{int}}$, $v_r \in V_{\mathtt{rat}}$, $k \in \mathbb{Z}$, and $q \in \mathbb{Q}$. These expressions will be used to capture conditions on the values of variables that are read and written during the execution of process activities. The set of constraints over a set of variables V is denoted $\mathcal{C}(V)$. We will also consider first-order formulas that have constraints as atoms. Given the definition of constraints, such formulas are in the theory of linear arithmetic, which is decidable [2]. Moreover, *quantifier elimination* can produce a quantifier-free, equivalent for any formula of the form $\exists x.\varphi$, cf. [2]. We denote logical equivalence by $\equiv$, and logical entailment by $\models$.

Data Petri nets. We adopt the standard definition of Data Petri Nets (DPNs) [19,20]. We consider two disjoint, marked copies of the set of process variables V, denoted $V^r = \{v^r \mid v \in V\}$ and $V^w = \{v^w \mid v \in V\}$, called the *read* and *write* variables. They will refer to variable values before and after a transition, respectively. We also write $\overline{V}$ for a vector that orders V in an arbitrary, fixed way, and $\overline{V}^r$ and $\overline{V}^w$ for vectors ordering V^r and V^w in the same way.

Definition 1. *A* data Petri net *(DPN) is a tuple* $\mathcal{N} = \langle P, T, F, \ell, \mathcal{A}, V, guard\rangle$, *where (1)* $\langle P, T, F, \ell\rangle$ *is a labelled Petri net with non-empty, disjoint sets of places* P *and transitions* T, *a flow relation* $F\colon (P \times T) \cup (T \times P) \mapsto \mathbb{N}$ *and a labelling function* $\ell\colon T \mapsto \mathcal{A} \cup \{\tau\}$, *where* $\mathcal{A}$ *is a finite set of activity labels and* τ *is a special symbol for silent transitions; (2)* V *is a set of process variables with a sort in* Σ*; and (3)* $guard\colon T \mapsto \mathcal{C}(V^r \cup V^w)$ *is a guard mapping.*

Example 2. The process in Fig. 2(a) is a DPN modelling a simple auction process. It maintains the set of variables $V = \{o, t\}$ of sort `rat`, where o holds the last offer issued by a bidder, and t is a timer. The action init sets the timer t to a positive value and the offer o to 0; while the timer did not expire, it can be decreased (action timer), or bids can be issued, increasing the current offer (bid); the item can be sold if the timer expired and the offer is positive (hammer). We denote this DPN, consisting of all actions drawn in black in Fig. 2, by $\mathcal{N}$. Moreover, we consider a variant of this DPN with an additional reset action that restarts the process if the offer is 0 (drawn in blue), and call this DPN $\mathcal{N}_{\mathsf{reset}}$.



(a) (b) (c)

Fig. 2. A DPN with its DDSA and constraint graph.

Also the process shown in Example 1 is a DPN.

The variables read and written by a transition t are denoted by $read(t) = \{v \mid v^r \text{ occurs in } guard(t)\}$ and $write(t) = \{v \mid v^w \text{ occurs in } guard(t)\}$. For instance, for t the activity bid in Fig. 2, $write(t) = \{o\}$ and $read(t) = \{o, t\}$. An assignment with domain V is called a *state variable assignment*, to distinguish it from a *transition variable assignment* β that assigns values to the set of variables $V^r \cup V^w$. All assignments are supposed to map variables to elements of their domain.

A *configuration* in a DPN $\mathcal{N}$ is a pair (M, α) given by a marking $M\colon P \mapsto \mathbb{N}$ for the underlying Petri net, together with a state variable assignment α. A configuration thus simultaneously accounts for the control flow progress and for the current values of all variables in V, as specified by α. For instance, $(\{\mathsf{p}_0\}, \left[\begin{smallmatrix} t=0 \\ o=0 \end{smallmatrix}\right])$ is a configuration of the DPNs of Example 2.

Definition 2 (Transition firing). *A transition $t \in T$ is* enabled *in (M, α) if a transition variable assignment β exists such that:*

(i) $\beta(v^r) = \alpha(v)$ for every $v \in read(t)$, i.e., β is as α for read variables;
(ii) $\beta \models guard(t)$, i.e., β satisfies the guard; and
(iii) $M(p) \geqslant F(p,t)$ for every p so that $F(p,t) \geqslant 0$.

An enabled transition may fire*, producing a new configuration (M', α'), s.t. $M'(p) = M(p) - F(p,t) + F(t,p)$ for every $p \in P$, and $\alpha'(v) = \beta(v^w)$ for every $v \in write(t)$, and $\alpha'(v) = \alpha(v)$ for every $v \notin write(t)$. A pair (t, β) as above is called (valid)* transition firing*, and we denote its firing by $(M, \alpha) \xrightarrow{(t,\beta)} (M', \alpha')$.*

Thus, a guard simultaneously expresses a condition on read variables, and an update on written ones: e.g., bid in Fig. 2 requires the current value of t to be positive and non-deterministically sets o to a new value that exceeds the current.

Given $\mathcal{N}$, we fix one configuration (M_I, α_0) as *initial*, where M_I is the initial marking of the underlying Petri net and α_0 is a state variable assignment that specifies the initial values of all variables in V. The final marking is denoted M_F. For instance, $\mathcal{N}$ in Example 2 admits a transition firing $(\{\mathsf{p}_0\}, \left[\begin{smallmatrix} t=0 \\ o=0 \end{smallmatrix}\right]) \xrightarrow{\mathsf{init}} (\{\mathsf{p}_1, \mathsf{p}_2\}, \left[\begin{smallmatrix} t=1 \\ o=0 \end{smallmatrix}\right])$ from its initial state; and $\{\mathsf{p}_3\}$ is the final marking.

A state (M', α') is *reachable* in a DPN if it is reached by a transition sequence from the initial state $(M_I, \alpha_0) \xrightarrow{(t_1,\beta_1)} \dots \xrightarrow{(t_n,\beta_n)} (M', \alpha')$. Such a sequence is also written as $(M_I, \alpha_0) \rightarrow^* (M', \alpha')$. We denote by $\mathcal{M}ark(\mathcal{N})$ the set of all such M', and by $\mathcal{C}onf(\mathcal{N})$ the set of all such (M', α), i.e., the sets of reachable markings and configurations. A transition sequence as above is a *process run* if $M' = M_F$. In this paper, we will assume that DPNs are *bounded*, i.e., that the number of tokens in reachable markings is upper-bounded by some $k \in \mathbb{N}$.

Data-aware Dynamic Systems with Arithmetic (DDSAs) are a simpler, equivalent model [9,10] that we will use for analysis tasks.

Definition 3. *A* DDSA *$\mathcal{B} = \langle B, b_I, \mathcal{A}, T, B_F, V, \alpha_I, guard\rangle$ is a labeled transition system where (1) B is a finite set of* control states*, with $b_I \in B$ the initial one; (2) $\mathcal{A}$ is a set of* actions*; (3) $T \subseteq B \times \mathcal{A} \times B$ is a* transition relation*; (4) $B_F \subseteq B$ are* final states*; (5) V is the set of* process variables*; (6) α_I the* initial variable assignment*; and (7) $guard\colon \mathcal{A} \mapsto \mathcal{C}(V^r \cup V^w)$ specifies* executability constraints *for actions over variables $V^r \cup V^w$.*

Every bounded DPN $\mathcal{N}$ can be equivalently expressed as a DDSA $\mathcal{B}$ over the same set of process variables V, by unfolding all possible markings (see [13] for details). The set of control states of $\mathcal{B}$ coincides thus with the set of markings of $\mathcal{N}$. Figure 2(b) shows a DDSA which corresponds to the DPN in Fig. 2(a). The action guards are the same as in the DPN, but have been omitted for readability.

If a control state $b \in B$ admits a transition to b' via action a, i.e., $(b, a, b') \in T$, this is denoted by $b \xrightarrow{a} b'$. A *configuration* of $\mathcal{B}$ is a pair (b, α) where $b \in B$ and α is a state variable assignment, and (b_I, α_I) is the initial one. As defined next, an action a transforms a configuration (b, α) into a new configuration (b', α') by updating the assignment α according to the action guard, exactly as in DPNs:

Definition 4. *A DDSA* $\mathcal{B} = \langle B, b_I, \mathcal{A}, T, B_F, V, \alpha_I, \mathit{guard}\rangle$ admits a step *from configuration* (b, α) *to* (b', α') *via action a, denoted* $(b, \alpha) \xrightarrow{a} (b', \alpha')$*, if* $b \xrightarrow{a} b'$*,* $\alpha'(v) = \alpha(v)$ *for all* $v \in V \setminus \mathit{write}(a)$*, and the transition assignment* β *given by* $\beta(v^r) = \alpha(v)$ *and* $\beta(v^w) = \alpha'(v)$ *for all* $v \in V$*, satisfies* $\beta \models \mathit{guard}(a)$*.*

A *run* ρ of a DDSA $\mathcal{B}$ is a sequence of steps $\rho\colon (b_I, \alpha_I) = (b_0, \alpha_0) \xrightarrow{a_1} (b_1, \alpha_1) \xrightarrow{a_2} \dots \xrightarrow{a_n} (b_n, \alpha_n)$, and it is *final* if $b_n \in B_F$. We call the *abstraction* of a run the respective transition sequence $b_0 \xrightarrow{a_1} b_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} b_n$. For instance, for the DDSA in Fig. 2(b), the following is a run ending in a final state (note that each state corresponds to a marking of the DPN):

$$(\{\mathsf{p}_0\}, \left[\begin{smallmatrix} t=0 \\ o=0 \end{smallmatrix}\right]) \xrightarrow{\mathsf{init}} (\{\mathsf{p}_1, \mathsf{p}_2\}, \left[\begin{smallmatrix} t=1 \\ o=0 \end{smallmatrix}\right]) \xrightarrow{\mathsf{bid}} (\{\mathsf{p}_1, \mathsf{p}_2\}, \left[\begin{smallmatrix} t=1 \\ o=5 \end{smallmatrix}\right]) \xrightarrow{\mathsf{timer}} (\{\mathsf{p}_1, \mathsf{p}_2\}, \left[\begin{smallmatrix} t=0 \\ o=5 \end{smallmatrix}\right]) \xrightarrow{\mathsf{hammer}} (\{\mathsf{p}_3\}, \left[\begin{smallmatrix} t=0 \\ o=5 \end{smallmatrix}\right])$$

The number of runs and configurations of a DPN or DDSA are typically infinite, due to the infinite number of possible valuations. For analysis tasks, we thus resort to the following abstraction:

Constraint graphs (CGs) are an abstraction of the reachable state space that was introduced for soundness checking [10,13]. The key idea is that formulas are used to represent sets of configurations.

Let $\mathcal{B} = \langle B, b_I, \mathcal{A}, \Delta, B_F, V, \alpha_I, \mathit{guard}\rangle$ be a DDSA. The *transition formula* Δ_a of action a is given by $\Delta_a(\overline{V}^r, \overline{V}^w) = \mathit{guard}(a) \wedge \bigwedge_{v \notin \mathit{write}(a)} v^w = v^r$. This formula simply expresses conditions on variables *before and after* executing the action: $\mathit{guard}(a)$ must hold, and the values of all variables that are not written are copied. E.g., for action bid in Fig. 2(b), $\mathit{write}(\mathsf{bid}) = \{o\}$, so $\Delta_{\mathsf{bid}} = (t^r > 0) \wedge (o^w > o^r) \wedge (t^w = t^r)$. Next, we define an *update* operation, to express how a set of configurations, captured by formula φ, changes when executing an action.

Definition 5. *For a formula* φ *with free variables* V *and action* a*,* $\mathit{update}(\varphi, a) = \exists \overline{U}.\varphi[\overline{U}/\overline{V}] \wedge \Delta_a[\overline{U}/\overline{V}^r, \overline{V}/\overline{V}^w]$*, where* U *is a set of variables that has the same cardinality as* V *and is disjoint from all variables in* φ*.*

Here, $\varphi[\overline{U}/\overline{V}]$ is the result of replacing variables $\overline{V}$ in φ by $\overline{U}$, and similar for Δ_a. For instance, if $\overline{V} = (o, t)$ we can take the renamed variables $\overline{U} = (o', t')$; for $\varphi = (t > 0) \wedge (o = 0)$ we then get $\mathit{update}(\varphi, \mathsf{bid}) = \exists o'\, t'.(t' > 0) \wedge (o' = 0) \wedge (o > o') \wedge (t = t')$, which is equivalent to $(t > 0) \wedge (o > 0)$. This reflects that, if in the process of Fig. 2, $t > 0$ and $o = 0$ hold, and bid is executed, then afterwards we still have $t > 0$ but also o is positive. Below, let $c_{\alpha_I} := \bigwedge_{v \in V} v = \alpha_I(v)$.

Definition 6. *The* constraint graph $\mathrm{CG}_{\mathcal{B}}$ *of* $\mathcal{B}$ *is a quadruple* $\langle S, s_0, \gamma, S_F\rangle$ *where the set of nodes* S *consists of tuples* (b, φ) *for* $b \in B$ *and a formula* φ *with free variables* V*, and* $\gamma \subseteq S \times \mathcal{A} \times S$*, inductively defined as follows:*

(i) $s_0 := (b_0, c_{\alpha_I}) \in S$ *is the initial node; and*
(ii) if $(b, \varphi) \in S$ *and* $b \xrightarrow{a} b'$ *such that* $update(\varphi, a)$ *is satisfiable, there is some* $(b', \varphi') \in S$ *with* $\varphi' \equiv update(\varphi, a)$*, and* $(b, \varphi) \xrightarrow{a} (b', \varphi')$ *is in* γ*, and*
(iii) the set of final nodes S_F *consists of all* (b, φ) *such that* $b \in B_F$*.*

Intuitively, the constraint graph describes all configurations reachable in $\mathcal{B}$: Every node combines a control state b with a formula φ: it represents all configurations (b, α) such that α satisfies φ. Figure 2(c) shows the CG for the DDSA obtained from $\mathcal{N}$ in Example 2, (final nodes are drawn with a double border). In fact, Fig. 2(c) is also the CG for the DDSA of $\mathcal{N}_{\mathsf{reset}}$ (basically, because the transition reset is not reachable). The crucial property of CGs is that they faithfully and completely represent the configuration space, in the following sense:

Lemma 1 ([13, Lem. 2]). $\mathrm{CG}_{\mathcal{B}}$ *has a path* $\pi\colon (b_I, c_{\alpha_I}) \to^* (b, \varphi)$ *s.t.* φ *is satisfied by* α *iff* $\mathcal{B}$ *has a run* $(b_I, \alpha_I) \to^* (b, \alpha)$ *whose abstraction is* $\sigma(\pi)$*.*

Here, for a path π in the CG, $\sigma(\pi)$ is the DDSA transition sequence along this path. Thus a path π in the CG captures all runs ρ with the same sequence of control states and actions such that the last assignment in ρ satisfies the formula in the last node of π. CGs are infinite in general, but for many classes of DDSAs occurring in practice, finite CGs can be computed [10,13]. These include DDSAs where all constraints are variable-to-variable/constant comparisons over $\mathbb{Q}$ like Example 2, and *bounded lookback* DDSAs whose behaviour, intuitively, depends only on a bounded number of past steps (this holds e.g. for Example 1).

3 Marking and Configuration Equivalence

Two Petri nets are *marking equivalent* if their sets of reachable markings coincide. While marking equivalence is in general undecidable for Petri nets [14], it is easy to decide for bounded Petri nets, by enumerating all markings. Here we consider marking equivalence, as well as the related problem of marking inclusion, for bounded DPNs. First, we note that if a DPN $\mathcal{N}$ was transformed into a DDSA $\mathcal{B}$ as described in [13], then the set of possible markings $\mathcal{M}ark(\mathcal{N})$ coincides with the set of reachable states in $\mathcal{B}$. Two DPNs $\mathcal{N}_1$ and $\mathcal{N}_2$ with respective DDSAs $\mathcal{B}_1$ and $\mathcal{B}_2$ are thus marking equivalent iff $\mathcal{B}_1$ and $\mathcal{B}_2$ have the same sets of reachable states. Since reachability of a single state in a DDSA is already undecidable (cf. [12, Rem. 2.6]), also marking equivalence of DPNs is undecidable.

However, we show that for bounded DPNs with finite CGs, marking equivalence can be read off the CGs: Suppose two DPNs were transformed into DDSAs $\mathcal{B}_1 = \langle B, b_I, \mathcal{A}, T, B_F, V, \alpha_I, guard\rangle$ and $\mathcal{B}_2 = \langle B, b'_I, \mathcal{A}, T', B_F, V, \alpha'_I, guard\rangle$. We assume that all components of the DDSAs coincide, except for initial states and transitions, but this does not restrict generality as control states can be unreachable. For a DDSA $\mathcal{B}$, let $\mathit{MReach}(\mathcal{B}) = \{b \mid (b, \varphi) \in S\}$ for S the set of nodes in $\mathrm{CG}_{\mathcal{B}}$, i.e., $\mathit{MReach}(\mathcal{B})$ is the set of control states of $\mathcal{B}$ that occur in the CG of $\mathcal{B}$.

Proposition 1. *Two DPNs $\mathcal{N}_1$ and $\mathcal{N}_2$ that correspond to DDSAs $\mathcal{B}_1$ and $\mathcal{B}_2$ with finite CGs satisfy $\mathcal{M}ark(\mathcal{N}_1) \subseteq \mathcal{M}ark(\mathcal{N}_2)$ iff $\mathit{MReach}(\mathcal{B}_1) \subseteq \mathit{MReach}(\mathcal{B}_2)$, and are marking equivalent iff $\mathit{MReach}(\mathcal{B}_1) = \mathit{MReach}(\mathcal{B}_2)$.*

Proof. First, suppose $\mathcal{M}ark(\mathcal{N}_1) \subsetneq \mathcal{M}ark(\mathcal{N}_2)$, and let $M \in \mathit{MReach}(\mathcal{B}_1)$. By Lemma 1, there is a run of $\mathcal{B}_1$ ending in a configuration (M, α). Thus M is a reachable state of $\mathcal{B}_1$, i.e., a reachable marking of $\mathcal{N}_1$, and hence also of $\mathcal{N}_2$, so there is a process run of $\mathcal{N}_2$ (and thus a run of $\mathcal{B}_2$) ending in a configuration (M, α'). By Lemma 1, the CG for $\mathcal{B}_2$ has a node (M, φ), i.e., $M \in \mathit{MReach}(\mathcal{B}_2)$.

Second, if $\mathit{MReach}(\mathcal{B}_1) \subseteq \mathit{MReach}(\mathcal{B}_2)$ and $M \in \mathcal{M}ark(\mathcal{N}_1)$, some process run of $\mathcal{N}_1$ and run of $\mathcal{B}_1$ end in a configuration (M, α). By Lemma 1, the CG for $\mathcal{B}_1$ has a node (M, φ). Since $\mathit{MReach}(\mathcal{B}_1) \subseteq \mathit{MReach}(\mathcal{B}_2)$, the CG for $\mathcal{B}_2$ has a node (M, φ'). By Lemma 1, there is a run of $\mathcal{B}_2$ ending in a configuration (M, α'). This shows the inclusion statement, so the one for equivalence follows. □

Configuration Equivalence. For DPNs, the perhaps more relevant notion than marking equivalence is equivalence of sets of configurations. Let two DPNs be *configuration equivalent* if their sets of reachable configurations coincide. First, note that for a DPN $\mathcal{N}$ with associated DDSA $\mathcal{B}$, the sets of configurations of $\mathcal{N}$ and $\mathcal{B}$ coincide. Thus, we can again check the problem on the level of DDSAs. For a DDSA $\mathcal{B}$ with constraint graph with node set S, and M a state of $\mathcal{B}$, consider $\varphi_{reach}(\mathcal{B}, M) = \bigvee\{\varphi \mid (M, \varphi) \in S\}$ as a formula representation of the configurations that can occur together with M.

Proposition 2. *Let two DPNs $\mathcal{N}_1$ and $\mathcal{N}_2$ correspond to DDSAs $\mathcal{B}_1$ and $\mathcal{B}_2$.*

(1) $\mathcal{C}onf(\mathcal{N}_1) \subseteq \mathcal{C}onf(\mathcal{N}_2)$ iff $\mathcal{M}ark(\mathcal{N}_1) \subseteq \mathcal{M}ark(\mathcal{N}_2)$ and $\varphi_{reach}(\mathcal{B}_1, M) \models \varphi_{reach}(\mathcal{B}_2, M)$ for all $M \in \mathcal{M}ark(\mathcal{N}_1)$.
(2) $\mathcal{N}_1$ and $\mathcal{N}_2$ are configuration equivalent iff they are marking equivalent and $\varphi_{reach}(\mathcal{B}_1, M) \equiv \varphi_{reach}(\mathcal{B}_2, M)$ for all $M \in \mathcal{M}ark(\mathcal{N}_1)$.
(3) If $M \in \mathcal{M}ark(\mathcal{N}_1) \cap \mathcal{M}ark(\mathcal{N}_2)$ and there is some assignment α that satisfies $\varphi_{reach}(\mathcal{B}_1, M) \wedge \neg\varphi_{reach}(\mathcal{B}_2, M)$ then $(M, \alpha) \in \mathcal{C}onf(\mathcal{N}_1) \setminus \mathcal{C}onf(\mathcal{N}_2)$.

Proof. (1) First, assume $\mathcal{C}onf(\mathcal{N}_1) \subseteq \mathcal{C}onf(\mathcal{N}_2)$, so $\mathcal{M}ark(\mathcal{N}_1) \subseteq \mathcal{M}ark(\mathcal{N}_2)$. Let $\alpha \models \varphi_{reach}(\mathcal{B}_1, M)$, so $\alpha \models \varphi$ for some (M, φ) in the CG of $\mathcal{B}_1$. By Lemma 1, there is a run of $\mathcal{B}_1$ ending in a configuration (M, α). So (M, α) is a configuration of $\mathcal{N}_1$, and hence of $\mathcal{N}_2$, so (M, α) is also reachable in $\mathcal{B}_2$. Again by Lemma 1, the CG for $\mathcal{B}_2$ has a node (M, φ') such that $\alpha \models \varphi'$, so $\alpha \models \varphi_{reach}(\mathcal{B}_2, M)$.

Second, suppose $\mathcal{M}ark(\mathcal{N}_1) \subseteq \mathcal{M}ark(\mathcal{N}_2)$ and $\varphi_{reach}(\mathcal{B}_1, M) \models \varphi_{reach}(\mathcal{B}_2, M)$ for all $M \in \mathcal{M}ark(\mathcal{N}_1)$. Let $(M, \alpha) \in \mathcal{C}onf(\mathcal{N}_1)$, so reachable in $\mathcal{B}_1$. By Lemma 1, the CG for $\mathcal{B}_1$ has a node (M, φ) such that $\alpha \models \varphi$, so $\alpha \models \varphi_{reach}(\mathcal{B}_1, M)$, hence $\alpha \models \varphi_{reach}(\mathcal{B}_2, M)$. By Lemma 1, some run of $\mathcal{B}_2$ (hence of $\mathcal{N}_2$) ends in (M, α). This shows (1), which implies (2); for (3) the reasoning is similar. □

Note that marking equivalence can also be decided by finitely many reachability queries, but not configuration equivalence if the state space is infinite.

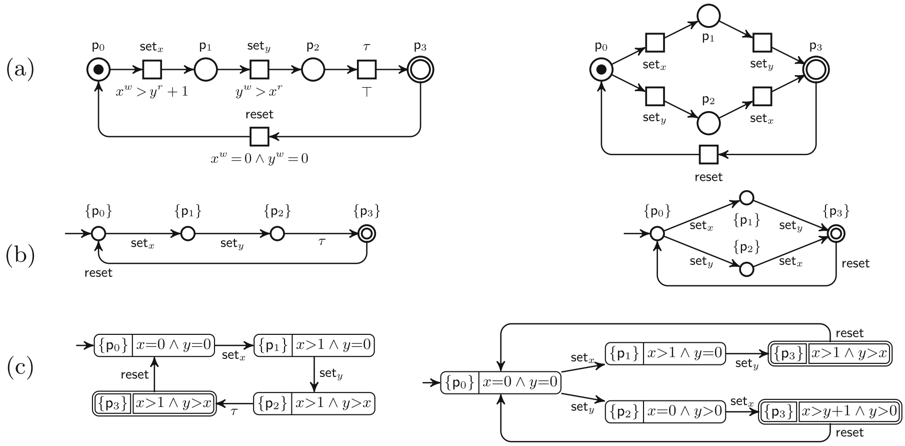


Fig. 3. Two DPNs with their DDSAs and constraint graphs.

Example 3. Consider the DPNs $\mathcal{N}_1$ and $\mathcal{N}_2$ in Fig. 3(a) over variables $V = \{x, y\}$ with sort `rat`, with $\alpha_I(x) = \alpha_I(y) = 0$ (the guards of actions in $\mathcal{N}_2$ coincide with those in $\mathcal{N}_1$). The respective DDSAs $\mathcal{B}_1$ and $\mathcal{B}_2$ and their constraint graphs are shown in Fig. 3(b) and (c). The markings occurring in the CGs coincide, so the two DPNs are marking equivalent. However, they are not configuration equivalent: For instance, the formulas $\varphi_{reach}(\mathcal{B}_1, \{\mathsf{p}_3\}) = (x{>}1 \wedge y{>}x)$ and $\varphi_{reach}(\mathcal{B}_2, \{\mathsf{p}_3\}) = (x{>}1 \wedge y{>}x) \vee (x{>}y{+}1 \wedge y{>}0)$ are not equivalent. This is witnessed by any assignment that satisfies $\varphi_{reach}(\mathcal{B}_2, M) \wedge \neg\varphi_{reach}(\mathcal{B}_1, M) \equiv (x{>}y{+}1 \wedge y{>}0)$, e.g., $\alpha(x) = 2$ and $\alpha(y) = 1$, so $(\{\mathsf{p}_3\}, \alpha)$ is a configuration of $\mathcal{N}_2$ but not of $\mathcal{N}_1$. However, $\varphi_{reach}(\mathcal{B}_1, M) \models \varphi_{reach}(\mathcal{B}_2, M)$ for all M, so $\mathcal{C}onf(\mathcal{N}_1) \subseteq \mathcal{C}onf(\mathcal{N}_2)$.

For another example, as the CGs of $\mathcal{N}$ and $\mathcal{N}_{\mathsf{reset}}$ from Example 2 coincide, the DPNs are marking and configuration equivalent. One can also use Propositions 1 and 2 to check that the DPN in Example 1, and the version in [19, Fig. 12.7] in which guards were discovered automatically, are marking but not configuration equivalent.

4 Language Equivalence

Language equivalence is undecidable for unbounded Petri nets [14], but decidable for bounded nets, in which case it basically amounts to checking equivalence of two regular languages [15]. Here we consider the respective problem for bounded

Algorithm 1. Checking language equivalence of node sets and DDSAs

```
procedure SetEquiv($X, Y, \mathcal{G}, \mathcal{G}'$)
    $R := \emptyset$, $todo := \{(X, Y)\}$
    while $todo \neq \emptyset$ do
        extract $(X, Y)$ from $todo$
        if $(X, Y) \in R$ then
            continue
        if $fin_{\mathcal{B}}(X) \neq fin_{\mathcal{B}'}(Y)$ then
            return false
        for each $a \in \mathcal{A}$ do
            add $(next_{\mathcal{G}}(X, a), next_{\mathcal{G}'}(Y, a))$ to $todo$
        add $(X, Y)$ to $R$
    return true
procedure Equiv($\mathcal{B}, \mathcal{B}'$)
    compute $\mathrm{CG}_{\mathcal{B}}$ and $\mathrm{CG}_{\mathcal{B}'}$, let $s_0$ and $s_0'$ be the initial states
    return SetEquiv($\{s_0\}, \{s_0'\}, \mathrm{CG}_{\mathcal{B}}, \mathrm{CG}_{\mathcal{B}'}$)
```

DPNs. Given a DPN $\mathcal{N}$, let its *language* $\mathcal{L}(\mathcal{N})$ be the set of all $\ell(t_1'), \dots \ell(t_m')$ such that there is a process run $(M_I, \alpha_0) \xrightarrow{(t_1,\beta_1)} \dots \xrightarrow{(t_n,\beta_n)} (M', \alpha')$ and $t_1', \dots, t_m'$ is the maximal subsequence of $t_1, \dots, t_n$ such that $\ell(t_i') \neq \tau$. We assume that silent transitions in DPNs do not write variables, and moreover, that there are no cycles that consist of silent transitions only. Thus, after transforming a DPN into a DDSA, we can eliminate silent transitions by replacing them with non-silent transitions, similar as done in NFAs: if $b \xrightarrow{t} b'$ with t silent with guard c, and $b' \xrightarrow{t_1} b_1, \dots, b' \xrightarrow{t_m} b_m$ are all transitions starting from b', we remove $b \xrightarrow{t} b'$ and add transitions $b \xrightarrow{t_1'} b_1, \dots, b \xrightarrow{t_m'} b_m$ with $guard(t_i') = guard(t_i) \wedge c$. This replacement can be repeated until there are no silent transitions left. Thus, we can consider the language equivalence problem for DDSAs without silent transitions. Let the language $\mathcal{L}(\mathcal{B})$ of a DDSA $\mathcal{B}$ be the set of all words $a_1, \dots, a_n$ such that $\mathcal{B}$ has a run $(b_I, \alpha_I) \xrightarrow{a_1} (b_1, \alpha_1) \xrightarrow{a_2} \dots \xrightarrow{a_n} (b_n, \alpha_n)$ and $b_n \in B_F$. Then, for every bounded DPN $\mathcal{N}$ there is a DDSA $\mathcal{B}$ such that $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{B})$.

We now consider two DPNs $\mathcal{N}_1$ and $\mathcal{N}_2$ that correspond to DDSAs $\mathcal{B}_1 = \langle B_1, b_I, \mathcal{A}, T, B_F, V, \alpha_I, guard \rangle$ and $\mathcal{B}_2 = \langle B_2, b_I', \mathcal{A}, T_2, B_F', V, \alpha_I', guard \rangle$. Note that the data variables, actions, and guards are supposed to coincide, but control states, transitions and the initial assignment may be different.

Language equivalence can be checked by an adaptation of an algorithm to check language equivalence of NFAs [3, Fig. 4], see the procedure Equiv in Algorithm 1. We use the following shorthands: for a set of nodes X in a constraint graph $\mathcal{G}$, $fin_{\mathcal{B}}(X)$ is true iff X contains a node that is final in $\mathcal{G}$. Moreover, $next_{\mathcal{G}}(X, a)$ is the set of all CG nodes s' such that $\mathcal{G}$ has an edge $s \xrightarrow{a} s'$ for some $s \in X$.

For a set of nodes X in a CG $\mathcal{G}$, we now write $\mathcal{L}_{\mathcal{G}}(X)$ for the set of words accepted starting from a state in X, i.e., the set of words $a_1, \dots, a_n$ such that $\mathcal{G}$ has a path $(b, \varphi_0) \xrightarrow{a_1} (b_1, \varphi_1) \xrightarrow{a_2} \dots \xrightarrow{a_n} (b_n, \varphi_n)$ from some $b \in X$ such that (b_n, φ_n) is final in $\mathcal{G}$. Then, procedure SetEquiv can be used to check whether two sets of states accept the same words. This is formally stated in the next result, which follows directly from [3, Prop. 2], considering $\mathcal{G}$ and $\mathcal{G}'$ as NFAs and X and Y as sets of states therein.

Proposition 3. *For sets of CG nodes X in $\mathcal{G}$ and Y in $\mathcal{G}'$,* $\textsc{SetEquiv}(X, Y, \mathcal{G}, \mathcal{G}')$ *is true iff* $\mathcal{L}_{\mathcal{G}}(X) = \mathcal{L}_{\mathcal{G}'}(Y)$.

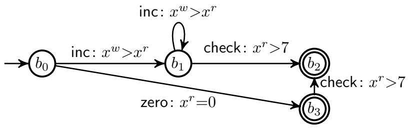
Proposition 4. *For two DPNs $\mathcal{N}_1$, $\mathcal{N}_2$ and DDSAs $\mathcal{B}_1$, $\mathcal{B}_2$ without silent transitions such that $\mathcal{L}(\mathcal{N}_1) = \mathcal{L}(\mathcal{B}_1)$ and $\mathcal{L}(\mathcal{N}_1) = \mathcal{L}(\mathcal{B}_1)$,* $\textsc{Equiv}(\mathcal{B}_1, \mathcal{B}_2) = \text{true}$ *iff $\mathcal{N}_1$ and $\mathcal{N}_2$ are language equivalent.*

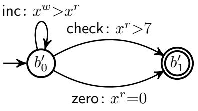
Proof (sketch). Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the CGs of $\mathcal{B}_1$ and $\mathcal{B}_2$. We have $\textsc{Equiv}(\mathcal{B}_1, \mathcal{B}_2) = \textsc{SetEquiv}(\{s_0\}, \{s_0'\}, \mathcal{G}_1, \mathcal{G}_2)$, which is true iff $\mathcal{L}_{\mathcal{G}_1}(\{s_0\}) = \mathcal{L}_{\mathcal{G}_2}(\{s_0'\})$ by Proposition 3. From Lemma 1 it follows that $\mathcal{L}_{\mathcal{G}_1}(\{s_0\})$ coincides with $\mathcal{L}(\mathcal{B}_1) = \mathcal{L}(\mathcal{N}_1)$, and similar for $\mathcal{B}_2$, so the claim follows. □

Language inclusion can be reduced to language equivalence since $\mathcal{L}(X) \subseteq \mathcal{L}(Y)$ iff $\mathcal{L}(X) \cup \mathcal{L}(Y) \subseteq \mathcal{L}(Y)$ iff $\mathcal{L}(X \cup Y) \subseteq \mathcal{L}(Y)$, cf. [3], so Algorithm 1 also serves to decide language inclusion.

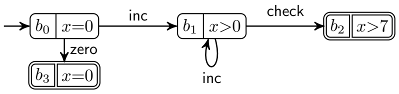
For instance, since the constraint graphs of $\mathcal{N}$ and $\mathcal{N}_{\mathsf{reset}}$ from Example 2 coincide, the DPNs are language equivalent. On the other hand, $\textsc{Equiv}$ can be used to detect that the languages of the DPNs in Fig. 3 are not equal (e.g., $\mathsf{set}_y, \mathsf{set}_x$ is not accepted by the first CG). The next example shows that it does not suffice to execute $\textsc{Equiv}$ on the DDSAs instead of the constraint graphs.

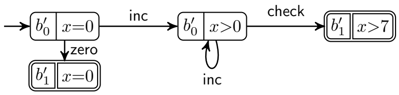
Example 4. Consider the following two DDSAs:





If both $\alpha_I(x) = \alpha_I'(x) = 0$, their CGs coincide, so $\textsc{Equiv}$ concludes language equivalence (the language being $\mathsf{inc}^{+}\mathsf{check} + \mathsf{zero}$, in regular expression notation).





However, note that the procedure $\textsc{Equiv}$ could not be run on the DDSAs directly, since detection of dead transition requires a reasoning based on reachable configurations, as done in CGs. Moreover, note that with e.g. $\alpha_I(x) = \alpha_I'(x) = 10$, the DDSAs would not be language equivalent because the single-letter word check would be in the language of the second, but not of the first DDSA.

Algorithm 1 can be modified to return a witness if equivalence does not hold. For reasons of space, we cannot formalize this here, but the main idea is straightforward: every pair in *todo* can be associated with a word that led to this pair of node sets, starting with the empty word ϵ for the initial nodes. When returning false in line 9, SetEquiv can then also return the word accumulated up to this point, which witnesses the difference.

It can also be noted that if $\mathcal{N}_1$ and $\mathcal{N}_2$ are language equivalent and $\alpha_I = \alpha'_I$, then the DPNs are also configuration equivalent, because actions have the same guards in both DPNs. However, the converse does not hold: if $\mathcal{N}_1$ is a copy of $\mathcal{N}_2$ where actions are renamed but have the same guards, then the two DPNs are configuration equivalent but not language equivalent.

Conclusion. We proposed techniques to check marking, configuration, and language equivalence for bounded DPNs with finite constraint graphs. Our correctness results thus imply that these notions are decidable for bounded DPNs with finite CGs, which captures many DPNs from practice [13]. To the best of our knowledge, these are the first results to compare DPNs based on behaviour. In future work, it would be interesting to study other notions of equivalence, e.g. language equivalence taking data into account. Also the study of quantitative similarity measures would be of interest, to e.g. express *how large* the intersection/difference of configuration spaces/language is. Our approach could also be implemented on top of the tool `ada` [11,13], which already computes CGs.

References

1. van der Aalst, W.M.P., de Medeiros, A.K.A., Weijters, A.J.M.M.: Process equivalence: comparing two process models based on observed behavior. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 129–144. Springer, Heidelberg (2006). https://doi.org/10.1007/11841760_10
2. Barrett, C., Sebastiani, R., Seshia, S., Tinelli, C.: Satisfiability modulo theories. In: Handbook of Satisfiability, 2nd edn., vol. 336, pp. 1267–1329. IOS Press (2021)
3. Bonchi, F., Pous, D.: Checking NFA equivalence with bisimulations up to congruence. In: Proceedings of 13th POPL, pp. 457–468. ACM (2013)
4. Damaggio, E., Deutsch, A., Vianu, V.: Artifact systems with data dependencies and arithmetic. ACM Trans. Database Syst. **37**(3), 22:1–22:36 (2012)
5. de Leoni, M., Mannhardt, F.: Decision discovery in business processes. In: Sakr, S., Zomaya, A. (eds.) Encyclopedia of Big Data Technologies, pp. 1–12. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-63962-8_96-1
6. Dijkman, R.M., Dumas, M., van Dongen, B.F., Käärik, R., Mendling, J.: Similarity of business process models: metrics and evaluation. Inf. Syst. **36**(2), 498–516 (2011)
7. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 48–63. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03848-8_5
8. Euzenat, J., Shvaiko, P.: Ontology Matching, 2nd edn. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38721-0
9. Felli, P., de Leoni, M., Montali, M.: Soundness verification of decision-aware process models with variable-to-variable conditions. In: Proceedings of 19th ACSD, pp. 82–91. IEEE (2019)

10. Felli, P., de Leoni, M., Montali, M.: Soundness verification of data-aware process models with variable-to-variable conditions. Fund. Inf. **182**(1), 1–29 (2021)
11. Felli, P., Montali, M., Winkler, S.: CTL* model checking for data-aware dynamic systems with arithmetic. In: Proceedings of 11th IJCAR, vol. 13385, pp. 36–56 (2022)
12. Felli, P., Montali, M., Winkler, S.: Linear-time verification of data-aware dynamic systems with arithmetic. In: Proceedings of 36th AAAI, pp. 5642–5650 (2022)
13. Felli, P., Montali, M., Winkler, S.: Soundness of data-aware processes with arithmetic conditions. In: Franch, X., Poels, G., Gailly, F., Snoeck, M. (eds.) Proceedings of 34th CAiSE, LNCS, vol. 13295, pp. 389–406. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-07472-1_23
14. Hack, M.: Decidability questions for Petri Nets. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge (1976)
15. Jančar, P., Moller, F.: Checking regular properties of petri nets. In: Lee, I., Smolka, S.A. (eds.) CONCUR 1995. LNCS, vol. 962, pp. 348–362. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60218-6_26
16. Kunze, M., Weidlich, M., Weske, M.: Behavioral similarity – a proper metric. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 166–181. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23059-2_15
17. de Leoni, M., Dumas, M., García-Bañuelos, L.: Discovering branching conditions from business process execution logs. In: Cortellessa, V., Varró, D. (eds.) FASE 2013. LNCS, vol. 7793, pp. 114–129. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37057-1_9
18. de Leoni, M., Felli, P., Montali, M.: A holistic approach for soundness verification of decision-aware process models. In: Trujillo, J.C., et al. (eds.) ER 2018. LNCS, vol. 11157, pp. 219–235. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_17
19. Mannhardt, F.: Multi-perspective Process Mining. Ph.D. thesis, Technical University of Eindhoven (2018)
20. Mannhardt, F., de Leoni, M., Reijers, H., van der Aalst, W.: Balanced multi-perspective checking of process conformance. Computing **98**(4), 407–437 (2016)
21. Schoknecht, A., Thaler, T., Fettke, P., Oberweis, A., Laue, R.: Similarity of business process models - a state-of-the-art analysis. ACM Comput. Surv. **50**(4), 52:1–52:33 (2017)

2nd International Workshop on Natural Language Processing for Business Process Management (NLP4BPM 2023)

2nd Workshop on Natural Language Processing for Business Process Management (NLP4BPM 2023)

Natural language plays a variety of roles in the context of business process management and analysis. Among other things, it can be used to describe processes in a comprehensible manner, define the meaning of events and activities, and provide support for the conduct of process analyses themselves, e.g., as an interface for process mining or modeling.

In this context, the goal of the NLP4BPM workshop is to bring together researchers and practitioners to present, discuss, and evaluate how natural language processing (NLP) can be used to establish new or improve existing methods, techniques, tools, and process-aware systems that support the different phases of the BPM life cycle. Furthermore, we aim to promote an exchange on the advances, challenges and barriers researchers encounter, and establish an environment where collaborations can naturally emerge.

Between the first and second edition of NLP4BPM, the attention that NLP receives in the public domain increased exponentially, due to the possibilities that Large Language Models (LLMs) and, particularly, generative NLP tools such as ChatGPT made available to virtually everyone. In the wake of these developments, the second edition of our workshop attracted nine submissions, comprising seven regular submissions and two dataset papers. The submissions were each single-blind reviewed by three members of the Program Committee. From these submissions, six were accepted for the workshop: four as regular papers for the main proceedings and two as short papers that will appear in associated CEUR proceedings. The six papers were presented at the BPM Conference in Utrecht, The Netherlands, attracting a large audience. After the two presentation sessions, we also hosted a round table on *The Impact of Large Language Models on Business Process Management*, in which a large number of attendees discussed a set of important and thought-provoking topics.

The four regular papers accepted include three research papers that address the use of textual attributes for predictive monitoring of knowledge-intensive processes and the application of LLMs to perform BPM tasks, and one dataset paper that introduces a collection of verb phrases that can be used to express activities and states in German business process models.

Brennig et al. present an approach for predictive process monitoring that focuses on prediction for knowledge-intensive processes (KiPs). A key characteristic is that the approach is text-aware, which allows it to consider the contents of textual attributes contained in an event log when making predictions. The evaluation of the approach shows that obtained predictions are more accurate than those made by state-of-the-art techniques, whereas it also reveals that, for such KiPs, the importance of the control-flow perspective is outweighed by other structured features and unstructured textual information that the approach specifically focuses on.

Laue et al. use their paper to introduce and describe a publicly available dataset containing 6,266 verb phrases that can be used to express activities and states in German business process models. The motivation behind the dataset is to address the problem that activities or states in business process models are often described using a multiword expression instead of a single verb. In the dataset, semantically equivalent verb phrases have been grouped into synsets. As a result, it is possible to identify the actual meaning of a textual label. The dataset has been compiled from a comprehensive analysis of 6,711 business process models with German labels and a study of already available collections of multiword expressions in the literature. The resource can be used for algorithms that analyze business process models with respect to the semantics of their labels.

Grohs et al. address the problem that existing NLP-based solutions in the BPM domain are highly specific to their respective tasks and cannot accomplish multiple process-related problems as a general-purpose instrument. Therefore, they use their paper to illustrate how LLMs can accomplish text-related BPM tasks by applying a specific LLM to three exemplary tasks: mining imperative process models from textual descriptions, mining declarative process models from textual descriptions, and assessing the suitability of process tasks from textual descriptions for robotic process automation. They show that, without extensive configuration or prompt engineering, LLMs perform comparably to or better than existing solutions. Based on their insights, they discuss implications for future BPM research as well as practical usage.

Berti et al. argue that the analysis of business processes could benefit from a natural process querying language and the domain knowledge encoded in LLMs. They identify, however, that a key issue in this context is that it is impossible to provide a complete database or event log as an input prompt due to size constraints. They therefore apply LLMs in the context of process mining by i) abstracting the information of standard process mining artifacts and ii) describing the prompting strategies applied. They implement the proposed abstraction techniques into pm4py and present a case study using available event logs. Starting from different abstractions and analysis questions, they formulate prompts and evaluate the quality of the answers.

The two short papers accepted introduce two additional applications of LLMs. *Farkas* uses them for recommending next elements in BPMN models, and *Jessen et al.* apply them for building chatbots for process mining tasks.

The organizers wish to thank all the people who submitted papers to the NLP4BPM 2023 workshop, the many participants creating fruitful discussions, and the NLP4BPM Program Committee members for their valuable work in reviewing the submissions. We look forward to future editions of the NLP4BPM workshop.

September 2023

Han van der Aa
Manuel Resinas
Adela del Río Ortega
Henrik Leopold

Organization

Organizing Committee

Han van der Aa	University of Mannheim, Germany
Manuel Resinas	University of Seville, Spain
Adela del Río Ortega	University of Seville, Spain
Henrik Leopold	Kühne Logistics University, Germany

Program Committee

Patrizio Bellan	Fondazione Bruno Kessler, Italy
Jordi Cabot	Open University of Catalonia, Spain
Josep Carmona	Process Talks, Spain
Fabiano Dalpiaz	Utrecht University, The Netherlands
Mauro Dragoni	Fondazione Bruno Kessler, Italy
Walid Gaaloul	Télécom SudParis, France
Chiara Ghidini	Fondazione Bruno Kessler, Italy
Daniela Grigori	Paris Dauphine University, France
Christoph Kecht	Technical University of Munich, Germany
Wolfgang Kratsch	FIM, Germany
Hugo A. López	University of Copenhagen, Denmark
Fabrizio Maria Maggi	Free University of Bozen-Bolzano, Italy
Jan Mendling	Humboldt-Universität zu Berlin, Germany
Adrian Rebmann	University of Mannheim, Germany
Hajo A. Reijers	Utrecht University, The Netherlands
Diana Sola	University of Mannheim, Germany
Sven Weinzierl	Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Karolin Winter	Eindhoven University of Technology, The Netherlands

Abstractions, Scenarios, and Prompt Definitions for Process Mining with LLMs: A Case Study

Alessandro Berti[1,2](✉), Daniel Schuster[1,2], and Wil M. P. van der Aalst[1,2]

[1] Process and Data Science Chair, RWTH Aachen University, Aachen, Germany
{a.berti,schuster,wvdaalst}@pads.rwth-aachen.de
[2] Fraunhofer FIT, Sankt Augustin, Germany

Abstract. Large Language Models (LLMs) are capable of answering questions in natural language for various purposes. With recent advancements (such as GPT-4), LLMs perform at a level comparable to humans for many proficient tasks. The analysis of business processes could benefit from a natural process querying language and using the domain knowledge on which LLMs have been trained. However, it is impossible to provide a complete database or event log as an input prompt due to size constraints. In this paper, we apply LLMs in the context of process mining by i) abstracting the information of standard process mining artifacts and ii) describing the prompting strategies. We implement the proposed abstraction techniques into *pm4py*, an open-source process mining library. We present a case study using available event logs. Starting from different abstractions and analysis questions, we formulate prompts and evaluate the quality of the answers.

Keywords: Process Querying · Prompting Engineering · Large Language Models · ChatGPT

1 Introduction

Process mining uses event data from information systems to enhance business processes, involving process discovery, conformance checking, model enhancement, and predictive analytics. This data science field provides insights for improving operational processes.

Transitioning from traditional process analysis, the emergence of Large Language Models (LLMs) like GPT-4 [17] adds a new perspective to data exploration. These advanced models, drawing on extensive training data, serve as versatile tools for general querying, enabling the extraction of valuable insights. They not only generate and retrieve information, but also hold potential to analyse and enhance business process outcomes.

In this paper, we investigate the applications of LLMs in the context of process mining, which are essential for process querying (i.e., in the verification of

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 427–439, 2024.
https://doi.org/10.1007/978-3-031-50974-2_32

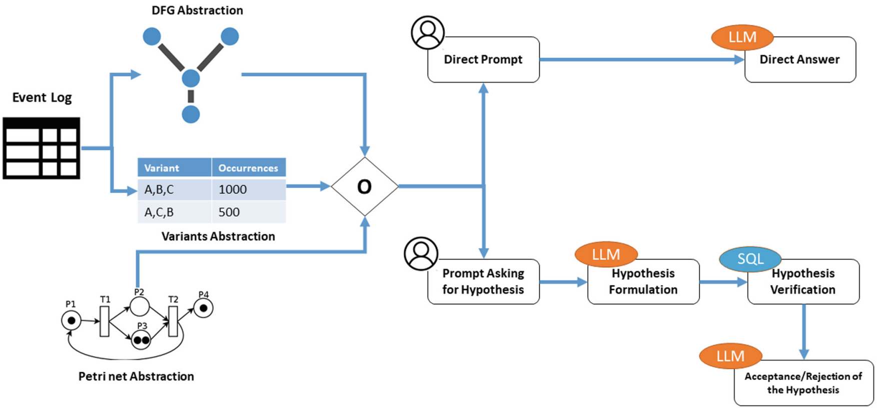


Fig. 1. Summary of the approach proposed in the paper: mainstream process mining artifacts can be textually abstracted and provided inside prompts directed to a LLM, such as GPT-4. Different prompting strategies can be considered.

properties against the event log or the preprocessing phase) and in embedding the domain knowledge (used to train the LLM) in the various process mining tasks. Despite their impressive performance, applying LLMs like GPT-4 to process mining presents challenges due to their 'context window' limitation [7,20], referring to the maximum sequence length these models can manage per interaction. This balancing act between information quantity and output quality can lead to significant data loss [12]. Strategies including text compression, context truncation, or improved prompts [9,21] are required to effectively encapsulate process mining information. Therefore, we explore in this paper the usage of textual abstractions of standard process mining artifacts, e.g., event logs and process models, that can embed the essential information of such artifacts.

This paper offers various prompting strategies to address the loss of information from proposed abstractions. A direct answer or a database query verified against the original object may be obtained, as summarized in Fig. 1. This study further presents the integration of the pm4py process mining library[1] with GPT-4 and provides a case study exploring these prompting strategies using public event logs. The case study examines responses under different abstractions and GPT-4's domain knowledge for various processes (medical, travel expense reporting, and fines management), alongside additional process mining knowledge required for specific use cases.

The remainder of the paper is organized as follows. Section 2 covers related work. Section 3 describes the abstractions and the different prompting strategies for LLMs. Section 4 describes the implementation. Section 5 presents a case study demonstrating the usage of different abstractions and prompting strategies for process mining tasks. Finally, Sect. 6 concludes this paper.

---

[1] https://pm4py.fit.fraunhofer.de.

2 Related Work

This section provides a brief overview of process querying and the usage of domain knowledge in process mining.

Several process-mining-specific querying languages exist [18]. In [19], a framework for devising process querying methods is proposed. SQL is widely used for process discovery [22], conformance checking [2], and data preprocessing [16]. Cypher, a graph-based querying language, has also been adopted for process mining [11]. Also, Celonis PQL [24] is a proprietary high-performance process querying language integrated into the Celonis platform. The mentioned languages are expressive and permit a versatile set of process mining inquiries. However, they require considerable expertise in the syntax and semantics of the query language in question and specialist knowledge.

The complexity of process querying can be reduced by translating natural language queries into database executable statements. As proposed in [4], a natural language querying interface aids in democratizing process mining, making it more accessible to non-technical users. The proposed reference architecture handles natural language questions and provides responses by integrating with process mining tools, using techniques such as entity recognition and semantic parsing. In [14], a natural language interface is proposed to assist the end-user in querying event data stored in a graph database. The natural language queries are translated into the Cypher language. Another study, [1], presents a conformance checking method based on NLP, which extracts business actions and objects from textual labels associated with events. Meanwhile, [26] focuses on identifying constraints for business process execution from natural language documents. In [13], chatbots are trained on customer service conversations to learn the underlying business process, showing the effectiveness of such an approach, though the generalization capabilities remain unclear.

Domain knowledge about a process can be expressed in natural language. For example, documents might contain the process execution rules if a formal model is not defined. Utilizing domain knowledge in process discovery has been investigated in [23]. In [10], the domain knowledge of the process analyst is used to modify/improve a discovered process model. In [3], an event log is abstracted to a level needed by the analyst using domain knowledge extracted from the documentation of the process to match semi-automatically events and activities.

The role of LLMs in the business process management field has been initially investigated in [8], where prompt engineering techniques to embed the required information about the business processes are discussed as an alternative to training a company/process-specific LLM.

This paper proposes the usage of LLMs for process mining tasks. LLMs such as GPT-4 know the domain knowledge and execution constraints for the set of business processes covered by the training data. Therefore, LLMs are not process-specific and can interpret and execute queries in natural language. In our case study, we show that the queries can be either executed directly against an abstraction of a given process mining artifact or database (SQL) queries can be automatically generated by GPT-4 to verify hypotheses.

3 Approach

When using LLMs for process mining, the original event logs or process model representations cannot be directly used due to size limitations. An abstraction of these artifacts must be obtained to execute specific queries, i.e., prompts, against an LLM. In the following subsections, we will explain textual abstractions (see Sect. 3.1) and different prompt generation strategies (see Sect. 3.2).

3.1 Abstracting Process Mining Objects

This section describes how textual abstractions of common process mining objects, i.e., event logs and process models, can be obtained. These abstractions are later used in the proposed case study.

3.1.1 Abstracting Event Logs Traditional event logs link each event with a single case identifier, enabling the computation of the directly-follows graph and the identification of traces and process variants [25]. These concepts can be associated with frequency and performance metrics

- In a directly-follows graph, frequency is quantified by the instances where a pair of activities are sequential, and performance is calculated as an aggregation, such as average or median, of recorded times between the two activities.
- For a process variant, frequency is determined by the count of cases following the given trace, while performance is an aggregation, such as average or median, of total throughput times for the cases.

This information can be textually represented to aid an LLM in responding to inquiries about event data. Section 5.2 and Listing 1.1 demonstrate the textual representation of variants and the top 5 relationships from a Directly-Follows Graph (DFG), respectively. When constructing the directly-follows graph, various notations may be employed such as → or the phrase “is followed by”. Despite the differences in representation, Large Language Models (LLMs) like GPT-4 interpret these notations equivalently.

Listing 1.1. Textual abstraction of a DFG.

```
Create Fine -> Send Fine ( frequency = 103392 performance = 7568635.65 )
Send Fine -> Insert Fine Notification ( frequency = 79757 performance = 1501626.95 )
Insert Fine Notification -> Add penalty ( frequency = 72334 performance = 5184000.0 )
Add penalty -> Send for Credit Collection ( frequency = 57182 performance = 45566346.44 )
Create Fine -> Payment ( frequency = 46952 performance = 905663.45 )
```

In the realm of object-centric event logs, wherein an event may associate with various object types, additional process modeling notations exist that can undergo textual abstraction. Specifically, object-centric directly-follows graphs [5] represent an assembly of directly follows graphs corresponding to distinct object types.

3.1.2 Abstractions of Process Models

Formal process models, e.g., Petri nets, BPMN, and declarative models, express constraints on the activities and the paths that are executable in a process. For example, the Petri net shown in Fig. 2 can be abstracted as in Listing 1.2. The method used for textually abstracting a Petri net is not fixed and can be approached in multiple ways, provided that the naming for places and transitions is unique. The choice of abstraction strategy is arbitrary and can be tailored to specific use cases or data structures. Similar textual abstractions of many other model formalisms (e.g., process trees, prefix trees, transition systems, BPMN models) are possible, but we do not describe them here.

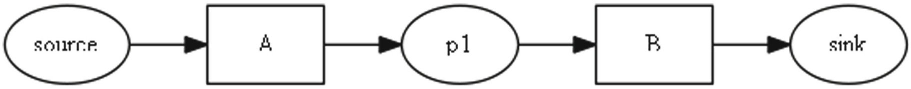


Fig. 2. Example sequential Petri net. From the initial state (source), the transitions A and B could be used to reach the final state (sink).

3.2 Prompt Generation

After obtaining the abstractions above, we can provide them to an LLM along with a query. These prompts could lead to two different types of answers, i.e., directly answering the original questions or leading to the formulation of hypotheses that can be verified against the data by means of database queries.

Listing 1.2. Textual abstraction of the Petri net represented in Fig. 2.

```
places: [ p1, sink, source ]
transitions: [ (A, ’A’), (B, ’B’) ]
arcs: [ (A, ’A’)->p1, (B, ’B’)->sink, p1->(B, ’B’), source->(A, ’A’) ]
initial marking: [’source:1’]
final marking: [’sink:1’]
```

3.2.1 Direct Answering

An LLM prompt can be formulated using abstractions, such as “Describe the meaning of the activity A”, which is particularly useful for descriptive or conformance checking purposes. It’s important that these prompts consider no more knowledge than the provided event log or process model abstraction.

Due to the inherently probabilistic behavior of LLMs like GPT-4, the same question might yield varying responses across different sessions. This feature, rather than being an issue, is part of the model’s design to promote diverse outputs and creative problem solving. If initial responses do not adequately meet the user’s need, refining the question or asking more specific follow-up questions is possible to address any perceived gaps in the information provided.

3.2.2 Hypothesis Formulation and Verification Certain process mining questions can be answered using the DFG/variants abstraction as they concern the order of activities. However, questions related to time and data perspectives of the event log, which require access to additional attributes or information, cannot be directly addressed by such abstractions. We may formulate hypotheses, such as impacts of specific activities on case duration, but these need further verification.

To verify a hypothesis, we can prompt an LLM, like GPT-4, with good SQL knowledge [15], to generate a database query that can be applied to the whole event log. The prompt uses the DFG/variants abstraction and an abstraction of event log attributes. Upon receiving the result of a query from the user, the LLM can then assess this information to confirm, refine, or dismiss the hypothesis.

It is also important to note that LLMs, provided with the top variants and attributes, can autonomously generate hypotheses on the data. Through provided abstractions, LLMs can make assertions and formulate database queries for hypothesis testing, demonstrating their flexibility and adaptability in process mining tasks.

Therefore, LLMs offer flexibility in formulating queries for hypothesis testing based on provided abstractions.

4 Implementation

In this section, we present the implementation of various abstractions (see Sect. 3.1) into the open-source process mining library, *pm4py* [6] (version 2.7.5 or later). The goal is to create textual abstractions of process mining artifacts, like traditional/object-centric event logs and process models (Petri nets), suitable for GPT-4's input limit. From these abstractions, specific queries are formulated for GPT-4 execution. Listing 1.3 demonstrates this integration, where an event log is ingested for root cause analysis, and the inductive miner algorithm discovers a process model for optimization suggestions.

Listing 1.3. Example usage of the pm4py's OpenAI/GPT-4 integration on traditional process mining objects

```
import pm4py

log = pm4py.read_xes(”tests/input_data/roadtraffic100traces.xes”)

iq1 = ”””\n What are the root causes of the performance issues in the process?
Please provide only process and data specific considerations,
no general considerations.”””
print(pm4py.llm.abstract_variants(log) + iq1)

net, im, fm = pm4py.discover_petri_net_inductive(log)


iq2 = ”””\n Can you provide suggestions to improve the process model
based on your domain knowledge?”””
print(pm4py.llm.abstract_petri_net(net, im, fm) + iq2)
```

5 Case Study

We present a case study using publicly available event logs and GPT-4 [17]. We propose an assessment of prompts that can be directly answered by GPT-4. Further, we propose an example of hypothesis formulation and verification against the entire dataset (by means of a SQL query).

5.1 Direct Answering

To assess prompts requiring direct answers from the LLM, we use publicly available event logs: (1) Road Traffic Fine Management process[2], which is related to the management of fines in an Italian municipality, (2) BPI Challenge 2020: Domestic Declarations[3], which is a travel expense approval process, (3) Sepsis Cases[4], which is a medical process for sepsis treatment, and (4) Conformance Checking Challenge 2019[5], which is a medical training process.

Table 1. Experimental results for the provided prompts (each containing an abstraction and a question) on publicly available event logs.

Question	Abstraction	Road Traffic	BPIC 2020	Sepsis	CCC 19
Descriptive Questions					
DQ1	DFG	🟩	🟩	🟩	🟩
DQ1	Variants	🟩	🟩	🟩	🟩
Conformance Questions					
CQ1	DFG	🟩	🟩	🟧	🟥
CQ1	Variants	🟩	🟩	🟥	🟧
Process Improvement Questions					
IQ1	DFG	🟧	🟧	🟧	🟧
IQ1	Variants	🟩	🟩	🟥	🟥
IQ2	Petri net	🟧	🟩	🟩	🟥

We have compiled a list of questions related to processes, sorted into various categories. Each question is accompanied by acceptance criteria to help determine if the response given by the LLM is satisfactory.

Descriptive Questions:

DQ1 Can you describe the process contained in this data?
  - GPT-4 should provide the name/category of the process underlying the data and the description of the main steps of the process).
  - If GPT-4 does not correctly understand the context and identifies the wrong name or category for the process, the response is considered unsatisfactory.

---

[2] https://data.4tu.nl/articles/_/12683249/1.

[3] https://data.4tu.nl/collections/_/5065541/1.

[4] https://data.4tu.nl/articles/_/12707639/1.

[5] https://data.4tu.nl/articles/_/12707639/1.

Conformance Questions:

CQ1 Can you pinpoint the central anomalies of the process from this data? Please only process and data-specific considerations, not general considerations.
  - Our expectation is that GPT-4, using its domain knowledge, is able to identify paths that are illogical, rework, or missing activities.
  - A response is deemed unsatisfactory if GPT-4 points to infrequent activities/paths, and to paths with high performance, without exploiting the domain knowledge about the process.

Process Improvement Questions:

IQ1 What are the root causes of performance issues specific to the process and related data? Please refrain from providing general considerations and focus on issues directly tied to the process and its data.
  - Our expectation is that GPT-4 should identify activities, paths, or rework that lead to higher throughput times.
  - A response is deemed unsatisfactory when GPT-4 identifies just the infrequent activities or paths, or is able to detect different execution orders for the activities but asks the user to verify if there is something wrong.

IQ2 Please suggest improving the process model based on your domain knowledge. Also, please compare it against implementations of similar processes. Provide only process and data-specific considerations, not general ones.
  - We expect that GPT-4 can suggest additional activities to optimize the throughput time and reduce rework. Also, it should be able to detect when the activities are executed in a suboptimal order.
  - A response is deemed unsatisfactory if general considerations about merging activities or reducing invisible steps are provided.

Certain queries align closely with those presented in [4]. Specifically, IQ1 and IQ2 correspond to questions 104 and 71 respectively, as listed in the provided resource (https://ic.unicamp.br/~luciana.barbieri/pmquestions.csv). Nevertheless, DQ1 and CQ1, which pertain to descriptive analytics and anomaly detection, exceed the capabilities offered by the Everflow tool.

All the considered prompts have been created starting from the result of abstraction and including one question. The prompts have been executed against GPT-4 [17]. We report on the quality of the answers in Table 1. Each row reports the results on the considered logs for a given abstraction and question. Different colors have been used: green indicates a useful answer for the given question, orange indicates a partly useful response, and red indicates that GPT-4 did not provide a good answer to the given question.

The response to descriptive questions has been satisfying for all event logs. GPT-4 for the considered questions and event logs is able to provide meaningful answers starting from both the directly-follows graph and variants. In particular, it is surprising that the DFG abstraction (which can be efficiently computed for large event logs) is sufficient not only to make local considerations (related to a path/activity of the process) but on the end-to-end execution of the process.

The conformance questions elicited satisfying responses for the “Road Traffic” and “BPI Challenge 2020” event logs (see Listing 1.4). Key issues like inconsistent activity order and multiple payment recordings were pinpointed for “Road Traffic”, while the “BPI Challenge 2020” rework in rejections/resubmissions were identified as problematic. Contrarily, responses for medical event logs “Sepsis” and “CCC19” were below expectations, possibly due to GPT-4’s limited domain knowledge of medical processes and difficulties handling their unstructured nature and high event count per case.

Responses to process improvement questions met expectations for the “Road Traffic” and “BPI Challenge 2020” event logs, especially with the variants abstraction. The directly-follows graph abstraction had limited utility in root cause analysis, as rework was more discernible in variants abstraction. However, for IQ2 employing model-based abstraction, GPT-4 offered valuable process optimization insights expressed in natural text.

For the contexts where both the DFG and variants abstraction are useful, we found that different viewpoints are provided. Therefore, the tips obtained thanks to different abstractions can be combined. Across different executions, we obtained slightly different answers (with similar degrees of usefulness), which can lead to a broader set of insights.

Listing 1.4. Response to CQ1 for the BPI Challenge 2020 event log.

```
From the given data, here are some of the potential anomalies I can identify in this process:

Inefficient Process Variants: There are several process variants where a declaration is rejected
    multiple times (by either ADMINISTRATION, PRE␣APPROVER, SUPERVISOR, or
    BUDGET OWNER) before it is finally approved. This suggests an inefficiency in the
    process that may require further investigation. It is possible that the declaration is not
    adequately prepared before submission or the criteria for approval are not clearly
    communicated. These inefficient process variants not only increase the frequency of steps in
    the process but also increase the overall performance time.

High Performance Time: Some process variants have remarkably high performance times. For
    instance, the process variant where the declaration is "SUBMITTED␣by␣EMPLOYEE␣−>␣
    Declaration␣REJECTED␣by␣ADMINISTRATION" has a performance time of 13805149.2.
    This high performance time could indicate a delay or bottleneck in the process that needs
    to be addressed.

Missing Steps: There is a process variant where a declaration is directly "FINAL␣APPROVED␣by
    ␣SUPERVISOR" without going through "APPROVED␣by␣ADMINISTRATION" or "
    APPROVED␣by␣BUDGET␣OWNER" or "APPROVED␣by␣PRE␣APPROVER". This could
    indicate a deviation from the standard process, which might either suggest an alternative
    process route or a process violation.

Unfinished Process: There are some process variants that do not end with "Payment␣Handled",
    such as "Declaration␣SUBMITTED␣by␣EMPLOYEE␣−>␣Declaration␣REJECTED␣by␣
    ADMINISTRATION␣−>␣Declaration␣REJECTED␣by␣EMPLOYEE" and "Declaration␣
    SUBMITTED␣by␣EMPLOYEE␣−>␣Declaration␣REJECTED␣by␣SUPERVISOR␣−>␣
    Declaration␣REJECTED␣by␣EMPLOYEE". This could mean that these processes are
    incomplete, or that the payment handling is being done outside of the tracked process.

Unexpected REJECTION: There are a few process variants where after being "
    FINAL␣APPROVED␣by␣SUPERVISOR", the declaration is "REJECTED␣by␣MISSING".
    This is unexpected as the declaration should ideally not be rejected after final approval.
    This anomaly might indicate some errors or issues in the process or system.
```

5.2 Hypothesis Formulation and Verification

We provide an example of hypothesis formulation and verification on top of the Road Traffic Fine Management event log. For this, we formulate the prompt in Listing 1.5, containing the top variants of the event log, and a summary of the numerical attributes of the event log.

In preliminary tests, GPT-4 required details about case identifier, activity, and timestamp attributes. It also attempted to access a non-existent variant attribute and needed guidance to compute the case's duration. Yet, given the limited information, GPT-4 surprisingly formulated plausible and testable hypotheses.

GPT-4 generates various hypotheses for the given event log, including a supposed influence of the ‘expense’ attribute on ‘Payment’ activity occurrence. Testing this hypothesis using the SQL query in Listing 1.6 shows it to be inaccurate, as the minor difference in average expenses between cases with and without payment isn't statistically significant. Given these results, GPT-4 suggests examining the ‘amount’ attribute's influence on payment presence, recognizing its initial hypothesis as unsubstantiated.

Listing 1.5. Prompt provided to GPT-4 for hypothesis formulation on the Road Traffic Fine Management event log.

```
If I have a process with the following process variants:
 Create Fine −> Send Fine −> Insert Fine Notification −> Add penalty −> Send for Credit
        Collection ( frequency = 56482 performance = 59591524.946000494 )
 Create Fine −> Payment ( frequency = 46371 performance = 889688.4000776347 )
 Create Fine −> Send Fine ( frequency = 20385 performance = 8380516.026490066 )
...

and the log of the process contains the following attributes:
amount empty: 331240 quantiles: {0.0: 0.0, 0.25: 33.6, 0.5: 38.0, 0.75: 71.5, 1.0: 8000.0}
article empty: 411100 quantiles: {0.0: 7.0, 0.25: 7.0, 0.5: 157.0, 0.75: 157.0, 1.0: 401.0}
expense empty: 457483 quantiles: {0.0: 0.0, 0.25: 10.0, 0.5: 11.88, 0.75: 13.5, 1.0: 76.0}
time:timestamp empty: 0 quantiles: {0.0: Timestamp(’2000−01−01␣00:00:00+0000’, tz=’UTC’),
        0.25: Timestamp(’2002−11−12␣00:00:00+0000’, tz=’UTC’), 0.5: Timestamp(’2006−02−28␣
        00:00:00+0000’, tz=’UTC’), 0.75: Timestamp(’2009−03−30␣00:00:00+0000’, tz=’UTC’), 1.0:
         Timestamp(’2013−06−18␣00:00:00+0000’, tz=’UTC’)}
...

Can you provide some hypothesis between the execution of the process and its attributes? I mean,
         can you provide me a DuckDB SQL query that I can execute, and return the results to
        you, in order for you to evaluate such hypothesis about the process? More in detail, the
        data is stored in a Pandas dataframe where each row is an event having the provided
        attributes (so there are no separate table containing the variant). Can you tell me in
        advance which hypothesis you want to verify? Please consider the following information: the
        case identifier is called ”case:concept:name”, the activity is stored inside the attribute ”
        concept:name”, the timestamp is stored inside the attribute ”time:timestamp”, the resource
        is stored inside the attribute ”org:resource”, there is not a variant column but that can be
        obtained as concatenation of the activities of a case, there is not a duration column but
        that can be obtained as difference between the timestamp of the first and the last event.
        Also, the dataframe is called ”dataframe”. You should use the EPOCH function of DuckDB
        to get the timestamp from the date.
```

Listing 1.6. SQL code provided by GPT-4 for the verification of an hypothesis on the entire dataset

```
WITH case_has_payment AS (
    SELECT "case:concept:name",
           MAX(CASE WHEN "concept:name" = 'Payment' THEN 1 ELSE 0 END)
                   as has_payment
    FROM dataframe
    GROUP BY "case:concept:name"
)
SELECT has_payment, AVG(expense) as avg_expense
FROM dataframe
JOIN case_has_payment ON
dataframe."case:concept:name" = case_has_payment."case:concept:name"
GROUP BY has_payment;
```

5.3 Limitations, Open Challenges, and Opportunities

The results indicate that GPT-4's proficiency in addressing advanced conformance and process improvement queries improves with mainstream and standardized processes. Generally, GPT-4 exhibits substantial process mining understanding, albeit with the need for simple instructions for computing variants and throughput time. Notably, it was intriguing that GPT-4 could decipher the entire process execution from the DFG abstraction.

Nonetheless, these insights warrant validation against a wider array of questions and event logs. Additionally, the assessment of the proposed questions was based on the stated acceptance criteria, which is somewhat subjective, and alternative criteria could be employed. Consequently, the presented case study should be regarded as a preliminary exploration of LLMs' applicability in process mining.

6 Conclusion

The findings of this study provide promising indications for the application of Large Language Models (LLMs) in process mining, underscoring their potential in handling complex queries and process interpretations. LLMs, such as GPT-4, demonstrate impressive proficiency in understanding and analyzing process structures, highlighting the vast opportunities these models could bring to the field.

However, several challenges persist. One key concern is privacy - a considerable number of companies may be reticent to upload their core data to public LLMs like GPT-4 due to the sensitivity of the information involved. This brings to the fore the need for private LLMs, which can balance the utility of large-scale language models with the security needs of individual organizations.

To address privacy concerns, proprietary LLMs could be developed, trained on a mix of general and company-specific data. While current open-source models lag behind GPT-4, they're improving, suggesting the feasibility of private, customized LLMs. These models could potentially enhance process mining's efficiency and adaptability.

References

1. van der Aa, H., Rebmann, A., Leopold, H.: Natural language-based detection of semantic execution anomalies in event logs. Inf. Syst. **102**, 101824 (2021)
2. Baader, G., Krcmar, H.: Reducing false positives in fraud detection: combining the red flag approach with process mining. Int. J. Account. Inf. Syst. **31**, 1–16 (2018)
3. Baier, T., Mendling, J., Weske, M.: Bridging abstraction layers in process mining. Inf. Syst. **46**, 123–139 (2014)
4. Barbieri, L., Madeira, E., Stroeh, K., van der Aalst, W.M.P.: A natural language querying interface for process mining. J. Intell. Inf. Syst. **61**, 1–30 (2022)
5. Berti, A., van der Aalst, W.M.P.: OC-PM: analyzing object-centric event logs and process models. Int. J. Softw. Tools Technol. Transf. **25**(1), 1–17 (2023)
6. Berti, A., van Zelst, S., Schuster, D.: PM4Py: a process mining library for python. Softw. Impacts **17**, 100556 (2023). https://www.sciencedirect.com/science/article/pii/S2665963823000933
7. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
8. Busch, K., Rochlitzer, A., Sola, D., Leopold, H.: Just tell me: prompt engineering in business process management. In: van der Aa, H., Bork, D., Proper, H.A., Schmidt, R. (eds.) BPMDS 2023. LNBIP, vol. 479, pp. 3–11. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-34241-7_1
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Dixit, P.M., Buijs, J.C.A.M., van der Aalst, W.M.P., Hompes, B., Buurman, H.: Enhancing process mining results using domain knowledge. In: Proceedings of the 5th International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA 2015), pp. 79–94. CEUR-WS.org (2015)
11. Esser, S., Fahland, D.: Multi-dimensional event data in graph databases. J. Data Semant. **10**(1–2), 109–141 (2021)
12. Kaplan, J., et al.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020)
13. Kecht, C., Egger, A., Kratsch, W., Röglinger, M.: Quantifying chatbots' ability to learn business processes. Inf. Syst. **113**, 102176 (2023)
14. Kobeissi, M., Assy, N., Gaaloul, W., Defude, B., Benatallah, B., Haidar, B.: Natural language querying of process execution data. Inf. Syst. **116**, 102227 (2023)
15. Liu, A., Hu, X., Wen, L., Yu, P.S.: A comprehensive evaluation of ChatGPT's zero-shot text-to-SQL capability. CoRR abs/2303.13547 (2023)
16. de Murillas, E.G.L., Reijers, H.A., van der Aalst, W.M.P.: Connecting databases with process mining: a meta model and toolset. Softw. Syst. Model. **18**(2), 1209–1247 (2019)
17. OpenAI: GPT-4 technical report. CoRR abs/2303.08774 (2023)
18. Polyvyanyy, A.: Process Querying Methods. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-92875-9
19. Polyvyanyy, A., Ouyang, C., Barros, A., van der Aalst, W.M.P.: Process querying: enabling business intelligence through query-based process analytics. Decis. Support Syst. **100**, 41–56 (2017)
20. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)

21. Reimers, N., Gurevych, I.: Sentence-bert: sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
22. Schönig, S., Rogge-Solti, A., Cabanillas, C., Jablonski, S., Mendling, J.: Efficient and customisable declarative process mining with SQL. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) CAiSE 2016. LNCS, vol. 9694, pp. 290–305. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39696-5_18
23. Schuster, D., van Zelst, S.J., van der Aalst, W.M.P.: Utilizing domain knowledge in data-driven process discovery: a literature review. Comput. Ind. **137**, 103612 (2022)
24. Vogelgesang, T., Ambrosy, J., Becher, D., Seilbeck, R., Geyer-Klingeberg, J., Klenk, M.: Celonis PQL: a query language for process mining. In: Polyvyanyy, A. (ed.) Process Querying Methods, pp. 377–408. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-92875-9_13
25. Weerdt, J.D., Wynn, M.T.: Foundations of process event data. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 193–211. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_6
26. Winter, K., Rinderle-Ma, S.: Detecting constraints and their relations from regulatory documents using NLP techniques. In: Panetto, H., Debruyne, C., Proper, H.A., Ardagna, C.A., Roman, D., Meersman, R. (eds.) OTM 2018. LNCS, vol. 11229, pp. 261–278. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_15

Text-Aware Predictive Process Monitoring of Knowledge-Intensive Processes: Does Control Flow Matter?

Katharina Brennig(✉), Kay Benkert, Bernd Löhr, and Oliver Müller

Faculty of Business Administration and Economics, Paderborn University, Warburger Str. 100, 33098 Paderborn, Germany
{katharina.brennig,bernd.loehr,oliver.mueller}@uni-paderborn.de,
kbenkert@mail.uni-paderborn.de

Abstract. Predictive process monitoring (PPM) enables organizations to predict the behavior of ongoing processes, e.g., the lead time. This is of great interest for knowledge-intensive processes (KIPs), which often cover long time spans. With such insights, resource allocation or customer relationship management could be improved. While already many PPM methods exist, they have not yet been applied to KIPs. Thus, we extend PPM research by using machine learning and natural language processing (NLP) to develop and evaluate a novel text-aware PPM approach tailored towards monitoring KIPs. By developing suitable features and considering various time intervals, our approach encodes and aggregates the event log. Using two real-world event logs, we assess our methodology. We demonstrate that the MAE improves as compared to state-of-the-art PPM methods. It shows that the control flow perspective of KIPs should primarily be neglected, while considering more structured features and unstructured textual information is essential.

Keywords: Predictive Process Monitoring · Knowledge-Intensive Processes · Machine Learning · Natural Language Processing

1 Introduction

A process mining approach that has received increased attention in recent years, is predictive process monitoring (PPM) [5,13,15]. The goal of PPM is to continuously provide predictions about the future behavior of ongoing process instances based on an examination of historical event logs of other instances of the same process as well as the recent past of the current instance [5]. The aspects of future behavior that can be predicted are manifold; for example, process outcome, next events, resource consumption, or (remaining) lead time [13]. Knowing in advance about deviations and delays before they occur in the execution of a process, enables organizations, among other things, taking preventive measures to avoid the development of those events and to avert financial loss [5,15].

PPM techniques work at the intersection of process mining and machine learning and are able to consider the control flow perspective of event data as

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 440–452, 2024.
https://doi.org/10.1007/978-3-031-50974-2_33

well as additional data related to cases and events [15,25]. Only few existing approaches can make use of additional textual information expressed in natural language (e.g., documents, notes, messages), although they can contain information essential to the prediction objective [2,15,20]. Furthermore, existing PPM approaches focus primarily on predicting the behavior of well-structured processes (e.g., administrative or manufacturing business processes), partially due to the availability of corresponding public data sets, e.g., through the BPI Challenge [5]. For these public data sets, performance benchmarks for a wide array of different PPM techniques exist [21,25]. However, not all processes in organizations are well structured. Knowledge-intensive processes (KIPs), for example, strongly depend on the application of tacit human-centered knowledge and are characterized by a high degree of flexibility, complexity and creativity [7,9,22]. As a consequence, due to the unexpected and unscheduled nature of knowledge generation, the process structure is less rigid [4]. Thus, it is frequently impossible to prescribe the exact order of events. They rather evolve during their instantiation, which makes it even more difficult for organizations to determine how long the process will take [4,7–9].

Hence, we argue that instead of focusing primarily on the control flow perspective of event logs, PPM of KIPs should consider additional structured and unstructured features [11]. However, to the best of our knowledge, the utility of text-aware PPM of KIPs has not yet been evaluated empirically. Against this background, this study makes the following contributions: Theoretically, we contribute a new text-aware PPM approach for aggregating and encoding the event log by neglecting the control flow, creating appropriate features, and considering different time intervals to predict the total lead time. We evaluate our approach using two real-life event logs showing that KIPs that have a limited number of process instances with high levels of complexity and flexibility are suitable for text-aware PPM. From a managerial perspective, as our results demonstrate the importance of unstructured data and textual information for PPM, it can guide organizations in how information systems ought to be designed to improve the data quality for future PPM approaches. In addition, as KIPs often cover long time spans, organizations can make more informed decisions about the time the process will need to be completed. With insight into this behavior, e.g., resource allocation or customer relationship management can be improved.

2 Background and Related Work

2.1 Knowledge-Intensive Processes

Approaches to business process management assume that processes are made up of repetitive activities that are carried out in accordance with a process model that specifies the execution flow in its entirety [4]. However, organizations also have processes that are characterized by high degrees of creativity, flexibility, and complexity and whose activities depend on the application of tacit human-centered knowledge; these processes are often referred to as KIPs [4,7–9,22].

Due to the choices of individual process stakeholders and unforeseen circumstances (e.g., unpredictable events, autonomous user decisions), variations from structured reference models are frequent. As a result, only in some cases KIPs can be linked to formal process models because knowledge application and generation is unplanned and unscheduled [4]. One common model template in the manufacturing domain is the stage-gate structure, where the creativity takes place in the stages and are accumulated within the gates to decide the upcoming direction of the process instance [3]. Further, KIPs frequently incorporate creative components with a large contribution coming from human-centered knowledge, and frequently does not include organized work operations [4]. Process workers use knowledge from, about, and obtained from a process in knowledge-intensive activities to follow organizational objectives and add value [7,12]. Whereas it is simple to encode, store, and communicate explicit knowledge (i.e., stored as unstructured data such as documents, or notes) tacit knowledge cannot be easily formalized and often remains hidden [4]. Since human-centered and knowledge-intensive activities are an essential component of the process, KIPs cannot be entirely automated or standardized [4,9,12]. This causes the issue, that event logs of KIPs with a sufficient maturity level are only scarcely available [8,9], raising the importance to consider unstructured data as well [11].

2.2 Predictive Process Monitoring

Predictive Process Monitoring (PPM) is part of the operational support techniques of process mining [23]. PPM approaches consider the control flow of event logs, which build the basis for PPM, as well as additional data related to cases and events [13,15,25]. To predict the future behavior of a running case, historical event logs are encoded and a prediction model is trained and evaluated [5,13,15].

Recent benchmark studies [21,25] have summarized valuable insights about the performance of different PPM approaches. Verenich et al. (2019) [25] provide a survey and an evaluation of existing lead time PPM approaches, resulting in a taxonomy. As processes are complex and comprise many different aspects, a plethora of different PPM approaches were created [13], especially for predicting the total lead time. This includes among others a technique based on random forests and logistic regression [20]. Further, stochastic Petri nets [18] and annotated transition systems [24] have been used for lead time prediction. Polato et al. (2018) [16] extended the latter technique [24] enabling data-aware prediction of the lead time and the next activity, by adding an ensemble of Naive Bayes and support vector regressors to a discovered transition system. Nevertheless, it was revealed that neural networks often outperform generative methods based on transition systems and stochastic Petri nets. However, they represent black box models that require large amounts of training data, making it difficult to interpret [25]. Thus, Tax et al. (2017) [19] proposed an LSTM approach, which was extended by Navarin et al. [14] by introducing extra features into the LSTM, resulting in data-aware prediction.

With respect to textual process data, a recent rise in works is noticeable. Developed approaches range from the use of Bag-of-Words (BoW), to topic modeling, and Paragraph Vectors, covering different degrees on how much the sequence of words matters. Current results suggest that BoW models perform generally well, especially on sufficiently large textual data sets [20]. Pegoraro et al. (2021) [15] extended the body of knowledge by leveraging LSTMs to process textual features by testing different text encoding approaches. Independent of the chosen encoding approach, they found that adding textual features led to an improvement in predictive performance [15]. Another recent work by Cabrera et al. (2023) [2] leverages Bidirectional Encoder Representations from Transfomers (BERT) and combines these with contextual word embeddings and LSTMs. They could achieve consistently better results than just using a LSTM with structured data and simple text encodings (e.g., BoW) [2,15]. Another interesting approach is to leverage text data from external but related sources like news. Following this approach, Yeshchenko et al. (2018) [27] were able improve the prediction consistently. Another idea by Weinzierl et al. (2019) [26] used optical character recognition to leverage information from scanned documents for PPM; yet, this approach remained a concept and was not evaluated.

As these existing approaches primarily focus on predicting behavior of well-structured processes (e.g., administrative or manufacturing processes) [5], the utility of leveraging PPM, especially incorporating NLP, has not yet been evaluated empirically for KIPs. This motivates our presented approach while also checking the applicability of the approach by Cabrera et al. (2023) [2].

3 Framework for NLP-Driven PPM of KIPs

Figure 1 demonstrates our framework for applying text-aware PPM to KIPs. At different stages of a running process instance, the (1) target variable is predicted. Afterwards, (2) an initial time-based train/test split of the event log is performed to avoid temporal leakage [10]. As we focus on KIPs with a stage-gate structure [3] (see Fig. 2), we take an incremental approach by using all information available up to the current gate-as the data is mainly created within the preceding stage- to predict the total lead time of a case, which can then be leveraged at the gate to decide on the next actions [11]. That means, if a case consists of n gates with corresponding preceding stages, we denote the beginning of the case as $t = 0$, the completion of the first gate as $t = 1$, and so on, while the completion of the total case corresponds to $t = n$. Following this approach, we have a growing set of features (see Fig. 2). Thus, (3) feature engineering is performed for each $t < n$ by extracting and encoding structured numerical and categorical features as well as unstructured textual information. Enriching the event log with unstructured textual information (e.g., documents, messages, notes) is important as this type of information usually plays a vital role in KIPs [11]. As our approach uses an event log only as a source for feature engineering, traces are represented as one-dimensional feature vectors. This design decision is based on the first design principle proposed by Löhr et al. (2022) [11]. It is suggested to neglect the control flow to find a balance between quality assurance and flexibility while performing KIPs. Whereas knowledge-intensive activities are performed within a stage, decision-intensive activities are relevant for each gate [11], as illustrated in Fig. 2.

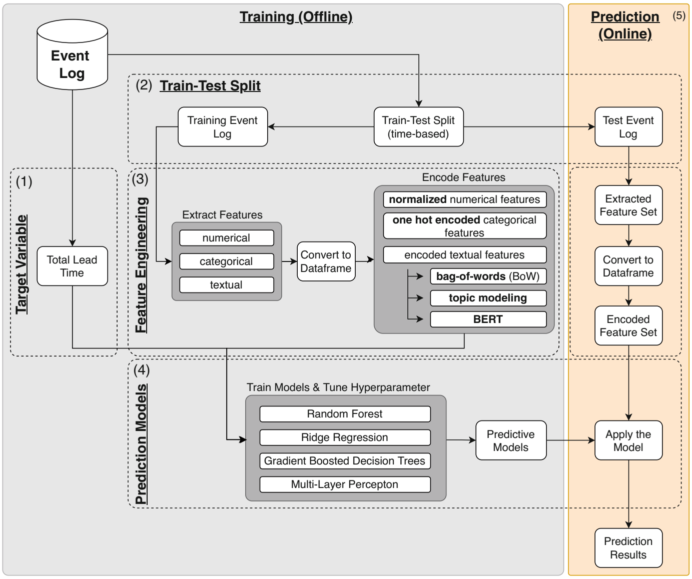


Fig. 1. Machine Learning Pipeline.

During the feature engineering phase, the numerical features are scaled using min-max normalization and the categorical features are encoded using one-hot encoding. To encode the textual information, we applied topic modeling, BoW and contextual embeddings (BERT), which all proved to be useful in prior PPM studies [2,15,20]. To perform topic modeling, we used a generative probabilistic model called latent dirichlet allocation that is used for discrete datasets like text corpora. This was implemented with its default parameters (i.e., number of topics, learning-method) using Python's *scikit-learn* package. To create BoW features, the tfidf vectorizer of the Python *scikit-learn* package with its default parameters (i.e., ngram-range, minimum document frequency of 1, maximum document frequency of 100%) was used. However, we set the parameter *sublinear-tf* to true as this applies a sublinear transformation to the term frequency (tf) values in the BoW representation. Nevertheless, before the text can be encoded using topic modeling and BoW we cleaned the text by removing stopwords, html tags and punctuation. Further, all whitespaces are converted to single whitespaces and stemming has been performed. To compute contextual embeddings, which are able to capture syntactic and semantic relationships between words as well as information about word order, we used SentenceTransformers [17]. More specifically, we used the *distiluse-base-multilingual-cased-v1*. The model produces contextual embeddings with 768 dimensions and can process sentences with up to 512 tokens. To derive predictions, (4) different learning algorithms are trained for each $t < n$. The prediction models receive as input a dataframe, where each row corresponds to a specific process instance and each column corresponds to one processed feature. During the training of the models feature elimination and hyperparameter tuning can be performed. Subsequently, these trained models are then (5) applied to the test set, evaluating how good the trained models perform on unseen data. To avoid leakage and mimic a realistic application scenario, the feature engineering was solely performed on the training set. The fitted feature transformers are then reused on the test set.

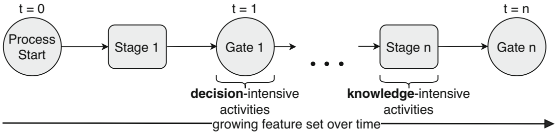

t = 0
Process Start
Stage 1
t = 1
Gate 1
Stage n
t = n
Gate n
decision-intensive activities
knowledge-intensive activities
growing feature set over time

Fig. 2. Stage-Gate Approach.

4 Evaluation

The performance of our approach is evaluated by considering several machine learning algorithms and comparing it to a NLP-based benchmark [2] using two real-world event logs from a manufacturing organization. The source code that was utilized to develop and test our models is accessible[1]. The event logs can not be made publicly available as they contain sensitive internal information of the organization involved. Below we describe our experimental setup in more detail.

4.1 Data Set

The used event logs describe a product innovation process. While the first one focuses on the *development* of new industrial connectors, the second one focuses on the *modification* of already existing industrial connectors. Within each stage, the activities can be carried out in arbitrary order to grant process participants maximal flexibility. These two processes can be identified as KIPs [7,9,22], characterized by a stage-gate structure, typical for product innovation processes [3]. Whereas the first event log contains three gates, the second event log contains two gates. The stages and gates of the respective processes were previously defined by the organization itself, as the stage-gate structure is firmly anchored in the process description. Both event logs have been exported from the organization's information system. Therefore, a special exporter was built, which manages to extract unstructured data and attach it to the event log. A more detailed overview of the event logs is given in Table 1.

---

[1] https://github.com/olivermueller/bpmi40_ppm.

Table 1. Data set description.

Event Log	Product Development	Product Modification
Cases	504	1710
Trace variants (determined by control flow)	504	1710
Completed events in total	32,258	62,903
Completed events per case (mean)	64	37
Number unique events in total	287	428
Median case duration (days)	210.880	41.010
Mean case duration (days)	237.250	64.510
Processed parts per case (mean)	12	15
Processed documents per case (mean)	10	6
Text size (word counts) per case (mean)	430.758	323.332

By following our approach, it is essential to define beforehand which features can be known at a given point in time. To determine whether the usage of a feature is appropriate for the learning task, domain expertise is needed. Otherwise, leakage could occur if the model has access to features that are not allowed to be used in the modeling process, or due to the absence of important features, resulting from a lack of knowledge of the data [10]. Therefore, we conducted several workshops with employees of the manufacturing organization to extract meaningful features for every t.

With respect to our processes, categorical features include information about the case (e.g., *Start Month*), resources (e.g., *Process Participants*, *Involved Divisions*, or *Start Division*), and individual activities (e.g., *Quality Check*, or *Most Repeated Activity*). Numerical features extracted from the case are comprised of the *Resources Count*, *Divisions Count*, and the *Lead Time of Completed Stages* of a case. Further, intended deviations are considered. Thereby, we focus on the *Number of Activities excluded on Purpose* that are normally included in the process, and on the *Number of Additional Added Activities*. We enriched the event log with textual information (e.g., documents, messages, notes), as they are very important for KIPs [11]. Text information about the whole process (e.g., reasons for performing the process, customer information) were attached as attributes at the trace level. Information about added documents and parts contain further textual descriptions and a categorical classification. This also allows for the extraction of additional features (e.g., *Processed Documents/Parts Count*, etc.).

4.2 Experimental Setting

As event logs represent sequences of events of different traces, they can be compared with time series data. As any good validation of time-series data should stay clear of future-looking, we performed a time-based train-test split (training

(75%), and a test set (25%)). We used the timestamp of the first event in each trace to sort the event log. As the prediction of the total lead time of a trace is a regression task, we report the Mean Absolute Error (MAE) for each prediction model and each $t < n$. For comparison, we use the approach of Cabrera et al. (2023) [2] as a benchmark. They provide a text-aware PPM technique. Their technique outperforms other text-aware PPM approaches that use non-contextualized word embeddings (e.g., the approach proposed by Pegoraro et al. (2021) [15]) [2]. Fine-tuning BERT toward predicting the next activity, outperforms all baselines they used for both prediction tasks. As activities determine the process control flow, or the sequencing of process stages, they hypothesize that fine-tuning toward the next activity leads to higher prediction outcomes [2]. Therefore, we selected their pre-trained and fine-tuned BERT toward the next activity as a benchmark and adjusted it to predict the total lead time.

For our approach, we used the models *Random Forest*, *Ridge Regression*, *Gradient Boosted Decision Trees*, and *Multi-Layer Percepton (MLP)*. The *MLP* was used with an adaptive learning rate, early stopping, a batch size of 32, and a tolerance for the optimization of 2.0. We included this neural network as they often outperform other methods [25]. As there exists no universally best model we tested several models that are suitable for regression. Further, recursive feature elimination (RFE) has been performed on *Random Forest*, *Ridge Regression*, and *Gradient Boosted Decision Trees*. RFE aims to select features by repeatedly taking into account smaller and smaller sets of features, given an external estimator that gives weights to features (e.g., the coefficients of a linear model)[2]. While training the models, the tuning of additional hyperparameters has been performed using Optuna Search. Optuna Search automates hyperparameter search with cross-validation, implementing a Bayesian optimization algorithm by default [1]. Applying Optuna Search, we performed a 5-fold cross-validation with 15 trials and set the scoring to the MAE. Table 2 defines the tuned hyperparameters for each model. Compared to Cabrera et al. (2023) [2], the main differentiating factor is the focus on aggregating the information according to the stage-gate structure, thus neglecting the control-flow while also enabling a prediction before the process did fully start.

5 Results

Table 3 shows the results of our approach as well as the benchmark [2]. The results demonstrate that our approach clearly outperforms the benchmark. At almost any point in time, the MAE of the baseline is substantially higher than the MAE of the other prediction models. Only for the *Product Development Process* in $t = 1$ the benchmark performs better than the *Gradient Boosted Decision Trees* both with BoW and with topic modeling. Nevertheless, the benchmark cannot produce results in $t = 0$, as it takes a prefix as its input and thus cannot make any predictions before at least some events have occurred. In addition,

---

[2] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html.

Table 2. Tuned hyperparameters for each model.

Model	Hyperparameters
Random Forest	n_features_to_select (for RFE), n_estimators, max_depth, max_features
Ridge	n_features_to_select (for RFE), alpha
Gradient Boosted Decision Trees	n_features_to_select (for RFE), n_estimators, learning_rate, max_depth
Multi-Layer Percepton	hidden_layer_sizes, learning_rate_init, max_iter

using our developed approach the MAE is improving significantly as the process progresses. Unfortunately, this does not apply to the benchmark since here the improvement in MAE is only very slightly, ranging from 169.247 days in $t = 1$ to 163.036 days in $t = 3$ for the *Product Development Process* and from 66.803 days in $t = 1$ to 65.266 days in $t = 2$ for the *Product Modification Process*.

The results reveal further that using our developed approach, BERT performs best in $t = 0$ with an MAE of 101.729 days up to $t = 2$ with an MAE of 64.009

Table 3. MAE (in days) of predicting the total lead time of a running case at different gates. For each gate, the lowest MAE value is underlined.

Language Model	Product Development				Product Modification		
	t = 0	t = 1	t = 2	t = 3	t = 0	t = 1	t = 2
Benchmark (based on [2])							
LSTM	N/A	169.247	163.229	163.036	N/A	66.803	65.266
Models with BoW							
Random Forest	107.776	126.274	68.036	45.041	45.431	42.889	**27.308**
Ridge	113.446	106.832	97.356	94.032	45.245	43.571	36.593
Gradient Boosted Decision Trees	131.427	209.513	80.327	49.725	50.021	45.140	27.833
Multi-Layer Percepton	125.962	108.655	112.309	110.526	46.607	46.327	46.429
Models with Topic Modeling							
Random Forest	112.454	127.281	66.676	**44.138**	45.183	**42.724**	27.467
Ridge	117.872	110.751	97.654	89.963	45.360	43.784	37.168
Gradient Boosted Decision Trees	126.960	171.267	80.340	54.013	50.249	44.660	28.173
Multi-Layer Percepton	186.053	111.725	110.727	123.186	46.544	46.566	46.372
Models with BERT							
Random Forest	**101.729**	111.828	**64.009**	48.745	44.790	43.029	28.771
Ridge	103.539	**106.342**	99.403	86.279	**44.734**	43.419	37.430
Gradient Boosted Decision Trees	145.253	146.917	94.964	55.671	45.770	44.200	30.357
Multi-Layer Percepton	117.895	107.424	102.197	133.601	46.446	46.520	49.662

days for the *Product Development Process*. Only in $t = 3$ topic modeling performs better than BERT with an MAE of 44.138 days. For the *Product Modification Process* BERT outperforms in $t = 0$ (MAE of 44.734 days), whereas in $t = 1$ topic modeling (MAE of 42.724 days) and in $t = 2$ BoW (MAE of 27.308 days) produce the best results. Nevertheless, it should be noticed that the differences in MAE between BoW, topic modeling, and BERT are marginal. Additionally, for both processes either *Random Forest* or *Ridge Regression* outperform the other models regardless of the used language models. Furthermore, it should be noted, that all MAE values are still high in absolute terms compared to mean and median case duration.

6 Discussion

PPM relies on the availability of large amounts of data based on structured processes [23]. However, our results expose that text-aware PPM techniques can also be applied to KIPs that have a limited number of process instances with high levels of complexity and flexibility [4,7]. Our results show that our approach can outperform the used benchmark in terms of predicting the total lead time of a case. In contrast to other PPM approaches, our approach indicates that focusing primarily on the control flow perspective is not necessary when performing text-aware PPM to KIPs [11]. Following our incremental approach, we have a growing set of features while also the effort for deriving time-specific features increases, as the quantity and quality of data available increase over time. This method makes it possible to incorporate new knowledge about the process as it progresses, making it possible to incrementally predict the total lead time. This, however, creates the requirement to train individual models at each gate instead of one uniform predictive model. Additionally, there is typically information available already at the beginning of a process which can be used to make early predictions [6]. Supported by the way of encoding and aggregating the event log while considering process information and unstructured text data that are independent of the control flow, it enables us to already make initial predictions in $t = 0$. Other approaches [2,15] use an event log as input, enabling them to only make predictions once a sequence of events has occurred. Further, using neural networks for PPM often outperform other models [25]. In contrast, our results show that *Random Forest* and *Ridge Regression* always performed best. Further, consistent with the benchmark [2], our results show that BERT performs best in most of the cases. Nevertheless, the differences in MAE for all language models are often only marginal (see Table 3). This leaves room for discussion of whether it is always necessary to train complex models, if less complex approaches achieve similar results and need less computational effort.

7 Conclusion

We propose a text-aware PPM approach to predict the total lead time of KIPs. By developing suitable features and taking into account various time intervals, our approach contains a novel method for encoding and aggregating the event log. Using two real-world event logs, we assess our methodology. We demonstrate that this approach improves the MAE as compared to the most recent state-of-the-art PPM methods. At the moment the manufacturing organization is integrating this approach into their information system so that they are able to incrementally predict the total lead time of their processes. This also enables other organizations to make more informed decisions about their processes, e.g., improving resource allocation or customer relationship management.

Although our results were significant, our study features limitations. Thus, we believe that the predictive performance of our approach can be further enhanced having more cases on which the prediction models can be trained. Additionally, using a higher amount of unstructured text data and fine tuning the used language models can also improve the predictive performance. In future research, the approach should be tested against further benchmarks and other text preprocessing steps, e.g., lemmatization. Additionally, it should be evaluated through other real-world event logs of KIPs (e.g., KIPs with a higher number of gates). Further, unstructured data such as drawings-which play a significant role in manufacturing KIPs-might be valuable to consider. Considering these limitations can help improving the practical value of our approach.

Acknowledgements. This research and development project is funded by the Ministry of Economic Affairs, Innovation, Digitalization, and Energy of the State of North Rhine-Westphalia (MWIDE) as part of the Leading-Edge Cluster, Intelligente Technische Systeme OstWestfalenLippe (it's OWL) and supervised by the project administration in Jülich (PtJ). The responsibility for the content of this publication lies with the authors.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631 (2019)
2. Cabrera, L., Weinzierl, S., Zilker, S., Matzner, M.: Text-aware predictive process monitoring with contextualized word embeddings. In: Business Process Management Workshops, pp. 303–314. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-25383-6_22
3. Cooper, R.G.: Stage-gate systems: a new tool for managing new products. Bus. Horiz. **33**(3), 44–54 (1990)
4. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. J. Data Semant. **4**(1), 29–57 (2015)
5. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Milani, F.: Predictive process monitoring methods: which one suits me best? In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 462–479. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_27
6. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Petrucci, G., Yeshchenko, A.: An eye into the future: leveraging a-priori knowledge in predictive business process

monitoring. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNCS, vol. 10445, pp. 252–268. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_15

7. Eppler, M.J., Seifried, P.M., Röpnack, A.: Improving knowledge intensive processes through an enterprise knowledge medium. In: Proceedings of the 1999 ACM SIGCPR Conference on Computer Personnel Research - SIGCPR 1999, pp. 222–230 (1999)
8. Gronau, N., Weber, E.: Management of knowledge intensive business processes. In: Desel, J., Pernici, B., Weske, M. (eds.) BPM 2004. LNCS, vol. 3080, pp. 163–178. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-25970-1_11
9. Isik, O., Van den Bergh, J., Mertens, W.: Knowledge intensive business processes: an exploratory study. In: 2012 45th Hawaii International Conference on System Sciences, pp. 3817–3826. IEEE (2012)
10. Kapoor, S., Narayanan, A.: Leakage and the reproducibility crisis in machine-learning-based science. Patterns **4**(9), 100804 (2023). https://doi.org/10.1016/j.patter.2023.100804. https://www.sciencedirect.com/science/article/pii/S2666389923001599
11. Löhr, B., Brennig, K., Bartelheimer, C., Beverungen, D., Müller, O.: Process mining of knowledge-intensive processes: an action design research study in manufacturing. In: Business Process Management, vol. 13420, pp. 251–267. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-16103-2_18
12. Marjanovic, O., Freeze, R.: Knowledge intensive business processes: theoretical foundations and research challenges. In: 2011 44th Hawaii International Conference on System Sciences, pp. 1–10. IEEE (2011)
13. Marquez-Chamorro, A.E., Resinas, M., Ruiz-Cortes, A.: Predictive monitoring of business processes: a survey. IEEE Trans. Serv. Comput. **11**(6), 962–977 (2018)
14. Navarin, N., Vincenzi, B., Polato, M., Sperduti, A.: LSTM networks for data-aware remaining time prediction of business process instances. In: Symposium Series on Computational Intelligence (SSCI), pp. 1–7. IEEE (2017)
15. Pegoraro, M., Uysal, M.S., Georgi, D.B., Van der Aalst, W.M.: Text-aware predictive monitoring of business processes. Bus. Inf. Syst. 221–232 (2021)
16. Polato, M., Sperduti, A., Burattin, A., Leoni, M.D.: Time and activity sequence prediction of business process instances. Computing **100**(9), 1005–1031 (2018)
17. Reimers, N., Gurevych, I.: Sentence-bert: sentence embeddings using siamese bert-networks. In: Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019)
18. Rogge-Solti, A., Weske, M.: Prediction of business process durations using non-Markovian stochastic Petri nets. Inf. Syst. **54**, 1–14 (2015)
19. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 477–492. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_30
20. Teinemaa, I., Dumas, M., Maggi, F.M., Di Francescomarino, C.: Predictive business process monitoring with structured and unstructured data. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 401–417. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_23
21. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: review and benchmark. ACM Trans. Knowl. Discov. Data **13**(2), 1–57 (2019)

22. Vaculin, R., Hull, R., Heath, T., Cochran, C., Nigam, A., Sukaviriya, P.: Declarative business artifact centric modeling of decision and knowledge intensive business processes. In: 15th International Enterprise Distributed Object Computing Conference, pp. 151–160. IEEE (2011)
23. van der Aalst, W.: Process Mining - Data Science in Action. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
24. Van Der Aalst, W., Schonenberg, M., Song, M.: Time prediction based on process mining. Inf. Syst. **36**(2), 450–475 (2011)
25. Verenich, I., Dumas, M., Rosa, M.L., Maggi, F.M., Teinemaa, I.: Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. ACM Trans. ITS **10**(4), 1–34 (2019)
26. Weinzierl, S., Revoredo, K., Matzner, M.: Predictive business process monitoring with context information from documents. In: Proceedings of the 27th European Conference on Information Systems (ECIS) (2019)
27. Yeshchenko, A., Durier, F., Revoredo, K., Mendling, J., Santoro, F.: Context-aware predictive process monitoring: the impact of news sentiment. In: Panetto, H., Debruyne, C., Proper, H.A., Ardagna, C.A., Roman, D., Meersman, R. (eds.) OTM 2018. LNCS, vol. 11229, pp. 586–603. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_33

Large Language Models Can Accomplish Business Process Management Tasks

Michael Grohs(✉), Luka Abb, Nourhan Elsayed, and Jana-Rebecca Rehse

University of Mannheim, Mannheim, Germany
`{michael.grohs,luka.abb,elsayed,rehse}@uni-mannheim.de`

Abstract. Business Process Management (BPM) aims to improve organizational activities and their outcomes by managing the underlying processes. To achieve this, it is often necessary to consider information from various sources, including unstructured textual documents. Therefore, researchers have developed several BPM-specific solutions that extract information from textual documents using Natural Language Processing techniques. These solutions are specific to their respective tasks and cannot accomplish multiple process-related problems as a general-purpose instrument. However, in light of the recent emergence of Large Language Models (LLMs) with remarkable reasoning capabilities, such a general-purpose instrument with multiple applications now appears attainable. In this paper, we illustrate how LLMs can accomplish text-related BPM tasks by applying a specific LLM to three exemplary tasks: mining imperative process models from textual descriptions, mining declarative process models from textual descriptions, and assessing the suitability of process tasks from textual descriptions for robotic process automation. We show that, without extensive configuration or prompt engineering, LLMs perform comparably to or better than existing solutions and discuss implications for future BPM research as well as practical usage.

Keywords: Business Process Management · Natural Language Processing · Large Language Models · ChatGPT

1 Introduction

The objective of Business Process Management (BPM) is to understand and supervise the execution of work within an organization. This ensures consistent outcomes and allows for the identification of improvement opportunities [6]. To accomplish this, BPM researchers and practitioners make use of diverse sources of information pertaining to business processes. These sources range from well-structured process models and event logs to unstructured textual documents [18]. In the past decade, BPM researchers have increasingly turned to Natural Language Processing (NLP) techniques to automatically extract process-related information from the abundant textual data found in real-world organizations.

M. Grohs, L. Abb, N. Elsayed—Equal contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 453–465, 2024.
https://doi.org/10.1007/978-3-031-50974-2_34

Many existing approaches utilize textual data for a wide range of BPM tasks. Examples of such tasks include the mining of imperative or declarative process models from textual process descriptions [8,19], process redesign for classifying end-user feedback [11], identifying suitable tasks for robotic process automation (RPA) in textual process descriptions [10], assessing process complexity based on textual data [16], or extracting semantic process information from natural language [13]. Although a few approaches also incorporate machine learning methods, the majority rely on extensive rule sets.

Each existing approach is designed for a specific purpose, meaning that it can only be applied to one specific task. A versatile general-purpose model that comprehends process-related text and seamlessly integrates it into various BPM tasks does not yet exist. However, the recent emergence of pre-trained Large Language Models (LLMs), which have demonstrated remarkable reasoning abilities across diverse domains and tasks [17], offers promising prospects for developing such a system. Already, multiple research groups are actively exploring the potential of these models in the BPM field, for example by analyzing which opportunities and challenges LLMs pose for the individual stages of the BPM lifecycle [20], how LLMs input should look like such that the output supports BPM [5], or whether conversational process modeling is possible [9].

These recent publications and pre-prints mostly illustrate the potential and difficulties of LLMs on a high level, but they do not showcase concrete applications. In this paper, we take a more application-oriented approach by investigating whether an LLM can accomplish three text-related BPM tasks: (1) mining imperative process models from textual descriptions, (2) mining declarative process models from textual descriptions, and (3) assessing the suitability of process tasks for RPA from textual descriptions. We selected these tasks because they are practically relevant and have previously been addressed in research. We evaluate how well the LLM can perform these tasks by benchmarking them against existing approaches that were specifically developed for the respective task. Based on the results, we discuss implications for future research in the field of BPM and illustrate how LLMs can support practitioners in their daily work.

The paper is structured as follows: In Sect. 2, we introduce the general solution approach that we followed for all three tasks. The task-specific applications and results are described in Sect. 4, Sect. 3, and Sect. 5, respectively. Section 6 discusses the future usage of LLMs in practice as well as implications for future research, before we conclude the paper in Sect. 7.

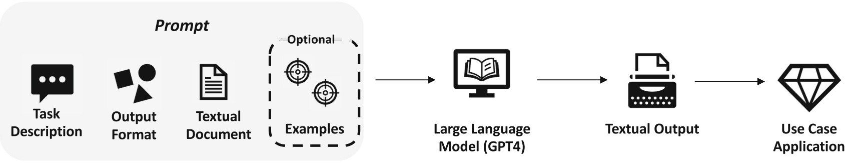


Fig. 1. Overview of our Approach

2 Approach

In this paper, we illustrate how LLMs can be utilized for three BPM tasks that require textual documents as input. For all tasks, we follow the same approach, illustrated in Fig. 1. We start by assembling a prompt with the following parts:

1. A general description of the BPM task that is to be accomplished.
2. A specification of a particular output format that the LLM should adhere to. This ensures that the generated text output has a certain level of consistency and that results are sufficiently standardized so that they can be further processed by, for example, parsing algorithms.
3. The natural language text that we want to abstract information from, e.g., a textual process description
4. Optionally, if suitable for a given task, few input-output pairs as examples

The complete prompt is then entered into the current state-of-the-art instruction-following LLM, ChatGPT with GPT4 backend [12] (henceforth referred to as GPT4). The textual output of GPT4 (i.e., the response to the prompt) is then evaluated with respect to its utility in solving the respective task and benchmarked against an existing approach. All parts of the prompt have not been specifically engineered but rather included such that the output is actually solving the tasks. The prompts were not optimized with respect to any metric.

In all applications, we provide the model with several prompts in order to check input robustness (i.e., how prompts from different authors influence the results) and output robustness (i.e., how the results change for different tries of the same prompt). By this, we aim to analyze whether GPT4 is able to accomplish specific tasks sufficiently well to be used by a diverse group of people and whether the results remain consistent despite the inherent randomness of the model output. For each task, we start with an "original" prompt written by one of the authors of this paper and enter this prompt three times (Tries 1 to 3). Two more prompts are then created by two other authors, who are given a general description of the task to be accomplished and the exact output format that they should specify, but who have not seen the original prompt. Finally, where appropriate, we also enter the original prompt without examples to evaluate the effect those have on the result. Each prompt is entered in a separate conversation window in the GPT4 web interface so that the model cannot draw on previous prompts as context.

All prompts, responses, and detailed evaluation results are available online[1].

3 Mining BPMN Process Models from Natural Language Descriptions

3.1 Motivation

Process models are the predominant tool for representing organizational activities and are often the starting point for process analyses [19]. Constructing

---

[1] https://gitlab.uni-mannheim.de/jpmac/llms-in-bpm.

such models requires knowledge of the process and proficiency in the creation of formal models [8]. However, the actors with process knowledge commonly are not experienced process modelers [8]. Therefore, modeling procedures can be very time-consuming and error-prone [15]. This holds true even though detailed textual descriptions of process requirements are often available in the form of policies, guidelines, or e-mail conversations, which can be considered relevant sources of information [8]. Approaches that extract process models from natural language can speed up the modeling and also enable managers to frequently update their process models without requiring extensive modeling experience.

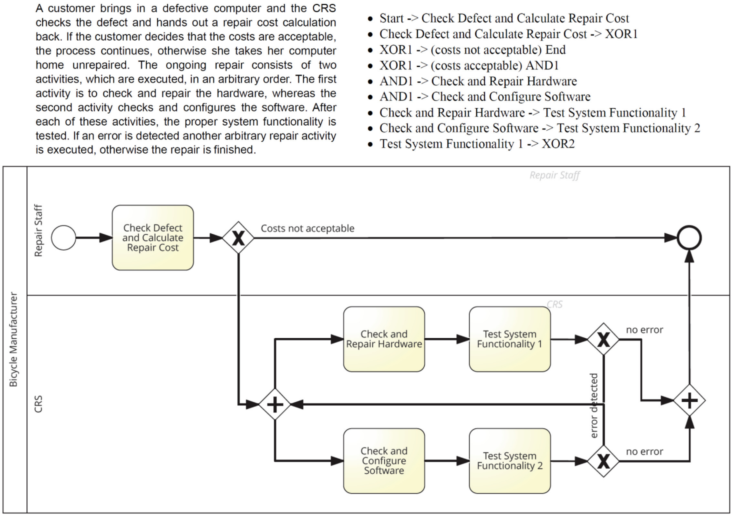
A rule-based approach to extract Business Process Model and Notation (BPMN) process models from textual process descriptions was first proposed in [8]. This remains the only generally-applicable, end-to-end technique able to produce a full imperative process model from text input, though several other publications with a more narrow scope or a focus on mining partial models exist (see [2] for a short review). There are also papers that investigate the ability of LLMs to extract process entities and relations from textual descriptions [3,9]. Though their approaches have some similarities to ours, neither ends up producing an actual process model from the text.

3.2 Evaluation

Following Fig. 1, we ask GPT4 to create a BPMN model for a process described in text. At the time of writing, the web interface version of GPT4 has a token output limit that prevents it from generating sequences at the length that would be required to generate BPMN models as XML files. We, therefore, prompt it to produce a model in a pre-specified intermediary notation as an output format that includes the main elements of BPMN and is straightforward to parse into a proper model representation. The template we provided in the prompt represents task nodes as natural language words, arcs between model elements as arrows (->), and exclusive and parallel gateways as `XOR` and `AND`, respectively. We also specify that outgoing arcs of exclusive gateways can be labeled to represent decision criteria, e.g., `XOR (Proposal accepted) -> Task1`. Finally, we ask the model to provide an actor-to-activity mapping that can be used to construct lanes, in the format `actor: [activity1, ...]`. Other elements (e.g., messages) are not included. We also do not provide example pairs of text and corresponding full or partial models to the LLM to avoid bias towards a certain modeling style.

Figure 2 shows an example of a textual description of a computer repair process, an excerpt of the response that GPT4 gave when presented with this description, and a visualization of the derived BPMN model. The generated model accurately represents the process described in the text. It could, however, be made slightly simpler by combining the two separate *Test System Functionality* activities and the subsequent exclusive gateways into one each.

For our evaluation, we use six process descriptions from [7] (1.1–1.4, 2.1, and 2.2). We selected these with the goal of applying our technique to a mix of short and simple as well as longer and more sophisticated textual descriptions. As ground truth, we use the annotations provided for these descriptions in the



A customer brings in a defective computer and the CRS checks the defect and hands out a repair cost calculation back. If the customer decides that the costs are acceptable, the process continues, otherwise she takes her computer home unrepaired. The ongoing repair consists of two activities, which are executed, in an arbitrary order. The first activity is to check and repair the hardware, whereas the second activity checks and configures the software. After each of these activities, the proper system functionality is tested. If an error is detected another arbitrary repair activity is executed, otherwise the repair is finished.

- Start -> Check Defect and Calculate Repair Cost
- Check Defect and Calculate Repair Cost -> XOR1
- XOR1 -> (costs not acceptable) End
- XOR1 -> (costs acceptable) AND1
- AND1 -> Check and Repair Hardware
- AND1 -> Check and Configure Software
- Check and Repair Hardware -> Test System Functionality 1
- Check and Configure Software -> Test System Functionality 2
- Test System Functionality 1 -> XOR2

Repair Staff
Bicycle Manufacturer
Repair Staff
CRS
Check Defect and Calculate Repair Cost
Costs not acceptable
CRS
Check and Repair Hardware
Test System Functionality 1
no error
error detected
Check and Configure Software
Test System Functionality 2
no error

Fig. 2. Example of a textual process description (top left), an excerpt of the generated LLM response (top right, from Prompt 1 Try 1), and visualization of the corresponding BPMN diagram (bottom).

PET dataset [1]. Specifically, we evaluate the output of the LLM with regard to how many of the relations described in the textual description are correctly identified (i.e., recall). Note that this allows us to simultaneously evaluate how many entities (task names and actors) are correctly identified since a relation that involves an unknown entity will be counted as not identified. We do not evaluate the models with regard to how many superfluous entities or relations they produce (i.e., precision) as that would raises several conceptual questions that require answers (e.g., how to treat a task that is correctly identified but in the wrong position), which would go beyond the intended scope of this paper.

We further restrict our evaluation to *flow* and *actor performer* relations, i.e., those that are present in the intermediary notation we provide in the prompts. Since the ground truth annotation applies only to the textual descriptions, we manually establish a mapping between the entities identified in the dataset and the ones produced by GPT4. In some cases, the relations produced by the LLM do not exactly match the ground truth (e.g., `Write Report` and `Send Report` are combined to `Write and Send Report`). For these, we follow the same approach as [3], i.e., we evaluate them on a case-by-case basis and count them as correct if they are semantically correct. As a benchmark, we use the process models produced by [7], applying the same evaluation criteria as described above.

The results of our evaluation are shown in Table 1, subdivided by the evaluation of output robustness (OR) and input robustness (IR). Overall, regarding the proportion of relations (and entities) that are correctly extracted from the textual process description, the models generated by GPT4 are comparable to the ones produced by [8]. Note that the absolute numbers reported should be

Table 1. Recall for the Text-to-BPMN Task

		Text 1.1	Text 1.2	Text 1.3	Text 1.4	Text 2.1	Text 2.2	Overall
OR	Prompt 1 Try 1	0.42	**0.58**	0.46	0.50	0.57	0.45	0.50
	Prompt 1 Try 2	**0.54**	**0.58**	0.38	**0.70**	**0.61**	0.42	0.54
	Prompt 1 Try 3	**0.54**	**0.58**	0.50	0.60	0.53	0.53	0.54
IR	Other Author (1)	**0.54**	0.47	0.54	0.50	0.47	0.34	0.48
	Other Author (2)	0.46	0.42	0.35	0.47	0.43	0.39	0.42
Benchmark [7]		**0.54**	0.47	**0.58**	0.55	0.55	**0.66**	**0.56**

interpreted with caution, because the PET ground truth is very fine-granular and we weigh all relation types equally, so that, for example, a single missing exclusive gateway (with the gateway itself, two decision criteria on the outgoing arcs, and two subsequent activities) would be counted as five non-identified relations. Consequently, a recall value of 0.5 should not be understood to indicate that the model only includes half of the relevant process behavior described in the text. Furthermore, the models generated by GPT4 are very precise in the sense that they tend to include a minimal (often insufficient) set of tasks, whereas the rule-based approach of [8] tends to produce models with several superfluent activities (e.g., `Begin Process` following a start event). Since our evaluation does not include a notion of false positive relations, it could be argued that we somewhat underestimate the quality of the LLM output relative to the benchmark.

Overall, an LLM-based text-to-BPMN technique produces reasonably good results. The model also produces consistent answers in the same intermediary notation when provided with the exact same description of the target template, so parsing its output into XML is possible. With prompt fine-tuning, and especially with subsequent prompting that asks the model to fix common issues, it is not unfeasible to create a reliable text-to-BPMN pipeline based on an LLM.

4 Mining Declarative Process Models from Natural Language Descriptions

4.1 Motivation

Not all business processes can be adequately captured by imperative modeling notations such as BPMN. For instance, knowledge-intensive processes have execution orders that cannot always be fully specified in advance [19]. These are better modeled using declarative process models, i.e., a set of formal constraints that do not rely on an explicit definition of allowed behavior [4]. They provide a flexible way of modeling processes, especially suitable in complex settings [4].

An approach that extracts declarative process models from natural language has been proposed in [19]. It uses the common declarative modeling language *Declare*, which is based on constraint templates grounded in Linear

Temporal Logic (LTL) [4]. By applying rule-based NLP techniques to sentences, the approach in [19] generates declarative constraints for five LTL templates: precedence, response, succession, initialization (init), and end. *Precedence(A, B)* (formal as `NOT(B) U A`) means that activity B should only occur after activity A. *Response(A, B)* (formal as `A -> B`) means that B must follow whenever A occurs. *Succession(A, B)* is the combination of *Precedence(A, B)* and *Response(A, B)*. *Init(A)* (formal as `START -> A`) prescribes that all process instances must start with A and *End(A)* (formal as `END -> A`) indicates that they must end with A.

4.2 Evaluation

In our experiment, we recreate the set-up from [19], applying GPT4 on the same five LTL templates and 104 test sentences. Following Fig. 1, we create a prompt that asks GPT4 to create LTL formulas in the form of precedence, response, succession, init, and end. For each template, we provide the output format and an example. As a result, the LLM outputs one or more discovered constraints in the format prescribed by the prompt, as shown in the exemplary excerpt of the output in Table 2. This output can then be compiled and translated into declarative modeling languages like *Declare*.

In addition to the three identical prompts for output robustness, we use two other formulations by different authors and, as we use examples in the original prompt, also one prompt without examples for input robustness. Table 3 displays precision (Prec.), recall (Rec.), and F1-score (F1) as used by [19] for each of the five LTL templates of the six different prompts compared to the benchmark.[2] We only consider syntactically correct classifications as true positives.

Except for the response template, GPT4 outperforms the benchmark and has a high precision value of close to 1. Further, we see that precision does not vary significantly with respect to output robustness for all LTL templates. With respect to recall, we see lower values for precedence. This is because many precedence constraints are misclassified as a response, which also explains the lower precision for this template. For succession and end, we see a high variation in the recall. This is due to a few constraints of these types in the 104 sentences,

Table 2. Exemplary Output of GPT4 for the Text-to-LTL Task

Sentence Input	GPT4 Output	LTL-Template
A claim should be created before it can be approved	NOT(approve claim) U create claim	Precedence
The process begins with the booking of the ticket	START -> book ticket	Init
Every provided laundry service must be billed	provide laundry service -> F(bill)	Response

---

[2] The corresponding confusion matrices can be found in our repository.

Table 3. Precision, Recall, and F1-Score for the Text-to-LTL Task

			Precedence	Response	Succession	Init	End	Overall
Output Robustness	Prompt 1 Try 1	Prec.	**0.96**	0.68	**1.00**	**1.00**	**1.00**	0.84
		Rec.	0.53	**0.96**	**1.00**	0.82	0.17	0.76
		F1	0.68	**0.80**	**1.00**	0.90	0.29	0.79
	Prompt 1 Try 2	Prec.	0.91	0.65	**1.00**	**1.00**	**1.00**	0.80
		Rec.	0.57	0.75	0.33	**1.00**	0.50	0.68
		F1	0.70	0.70	0.50	**1.00**	0.67	0.74
	Prompt 1 Try 3	Prec.	0.94	0.68	**1.00**	**1.00**	**1.00**	0.83
		Rec.	0.61	0.88	0.25	**1.00**	0.60	0.76
		F1	0.74	0.77	0.40	**1.00**	0.75	0.79
Input Robustness	No Examples	Prec.	0.57	0.51	0.33	0.83	**1.00**	0.58
		Rec.	0.08	0.79	0.67	0.91	0.50	0.49
		F1	0.14	0.62	0.44	0.87	0.67	0.53
	Other Author (1)	Prec.	0.94	0.72	**1.00**	**1.00**	**1.00**	**0.86**
		Rec.	0.65	0.88	**1.00**	0.82	0.67	**0.77**
		F1	**0.77**	0.79	**1.00**	0.90	0.80	**0.81**
	Other Author (2)	Prec.	0.91	0.71	0.75	0.90	**1.00**	0.83
		Rec.	0.61	0.77	**1.00**	0.82	0.83	0.72
		F1	0.73	0.74	0.86	0.86	**0.91**	0.77
Benchmark [19]		Prec.	0.78	**0.8**	0.68	0.75	0.88	0.77
		Rec.	**0.71**	0.77	0.68	0.82	**0.88**	0.72
		F1	0.74	0.75	0.68	0.78	0.88	0.74

meaning that few misclassifications have a high impact. With respect to input robustness, the evaluation metrics are worse if no examples for the LTL templates are provided. This is especially visible for the precedence template. In contrast to that, different formulations from other authors do not have a significant impact on the metrics. Rather, stability across different prompts is visible.

The F1-score shows that all prompts with examples for the LTL templates yield equal or higher scores than the benchmark. This illustrates that GPT4 outperforms the specific approach from [19] if it is provided with proper examples. This is an important finding as it indicates that prompts yield different results based on their fit to the task. Further, for tasks like this with short input text to be classified and a few classification targets, we recommend that the prompt should include examples. It should be noted that other prompts for example with additional information or the repetition of instructions could yield even better results. Further, the output of GPT4 has to be parsed into declarative process models using for example *Declare* to allow complete usage. This is possible in an automatic manner given the consistent output format for all 104 sentences.

5 Assessing RPA Suitability of Process Tasks from Natural Language Descriptions

5.1 Motivation

RPA is a technology that aims to automate routine and repetitive tasks in business environments. To do so, software robots that work on the user interface of software systems are developed to perform these tasks the same way human actors would do, thus increasing operational efficiency [14].

Various process information can be used to identify tasks that are suitable for RPA. This includes textual process descriptions, which are commonly used to document processes [18]. The approach proposed in [10] identifies suitable tasks for RPA by measuring the degree of the automation of process tasks using supervised machine learning techniques from the textual descriptions of business processes. From this textual data, the approach classifies the process tasks into manual, automated, or user tasks. Manual tasks are the tasks performed by a human actor without any use of an information system, user tasks consist of humans interacting with an information system, and automated tasks are performed automatically on an information system without any human involvement. Tasks classified as user tasks are suitable RPA candidate tasks as they can be automated by replicating human interactions by means of RPA agents. This increases the efficiency of identifying suitable RPA tasks in comparison to a manual analysis that takes a long time and effort, especially if there exists a large number of such documents or a large number of processes to be analyzed [10].

5.2 Evaluation

Following the approach from Fig. 1, GPT4 is used to replicate the experiment of [10]. The task is described in the prompt by asking the LLM to classify process tasks into one of three output formats: manual, user, or automated task. Possible features that might affect the task classifications (e.g., verb feature, object feature, resource type (human/non-human), and IT domain) are included in the task description. The output format as well as an example of tasks' classification for tasks of a given process description is also provided. We use the same dataset as [10], consisting of 33 textual process descriptions obtained from [7]. These descriptions consist of 424 process tasks to be classified. See Table 4 for an example of an input process description and the output generated by GPT4.

Table 4. Exemplary Output of GPT4 for the RPA Classification Task

Task Input	GPT4 Output
register a claim performed by claims officer	User task
examine a claim performed by claims officer	Manual task
write a settlement recommendation performed by claims officer	Manual task
send the claim back to the claims performed by claims officer	User task

Table 5. Precision, Recall, and F1-Score for the RPA Task

			Manual	User	Automated	Overall
Output Robustness	Prompt 1 Try 1	Prec.	0.68	**0.88**	0.15	0.79
		Rec.	0.83	0.6	0.75	0.67
		F1	0.75	0.74	0.45	0.73
	Prompt 1 Try 2	Prec.	0.65	0.84	0.73	0.78
		Rec.	0.69	0.83	0.69	0.78
		F1	0.67	**0.84**	**0.71**	0.78
	Prompt 1 Try 3	Prec.	0.84	0.84	0.32	**0.82**
		Rec.	0.65	0.83	**0.93**	0.78
		F1	0.75	**0.84**	0.63	0.8
Input Robustness	No Examples	Prec.	**0.85**	0.77	0.34	0.78
		Rec.	0.42	**0.88**	0.88	0.74
		F1	0.64	0.83	0.61	0.76
	Other Author (1)	Prec.	0.44	0.71	0.29	0.61
		Rec.	0.42	0.74	0.13	0.62
		F1	0.43	0.73	0.21	0.62
	Other Author (2)	Prec.	0.5	0.87	0.36	0.74
		Rec.	0.82	0.55	0.8	0.64
		F1	0.66	0.71	0.58	0.69
Benchmark [10]		Prec.	0.81	0.8	**0.92**	0.81
		Rec.	**0.9**	0.7	0.52	**0.8**
		F1	**0.85**	0.75	0.66	**0.81**

We did three identical prompts, we use two other prompts by different authors with an example in each. We also did one prompt without an example. Table 5 displays precision (Prec.), recall (Rec.), and F1-score (F1) for each of the six prompts compared to the benchmark from [10]. For the overall results, we apply the same micro-averaging approach as the benchmark, i.e., the number of tasks belonging to a class was used to weigh the respective precision and recall values.

GPT4 outperforms the benchmark for 4 out of 6 prompts for user tasks. For the automated tasks, precision results are below the benchmark because many tasks were classified by GPT4 as automated although they are not. However, the recall for this class outperforms the benchmark in almost all the prompts. For the F1-score, it is similar to the benchmark for the classes except for the user class where the F1-score results were higher than the benchmark. Overall, as indicated by the F1-score, GPT4 performs similarly to the benchmark for all six prompts. We also saw the performance of GPT4 deteriorate over time. We suspect that this is caused by the limited context window of GPT4, combined with the large number of tasks to be classified (424). In such cases, reminding the LLM of the task description between inputs could yield better results.

6 Discussion

After illustrating that out-of-the-box GPT4 performs similarly or even better than specialized approaches for our three exemplary tasks, we now want to discuss the usage of LLMs in practice and provide guidelines for users.

Prompt Recommendations. In our experiments, we found that including different contents in the prompt increase the performance of GPT4. For example, the output should be clearly defined instigating the task. Further, for the text-to-LTL task, examples led to better results. We can therefore recommend specifying the output format and to try using examples if feasible. In general, different prompts should be used and compared to maximize the benefits of using GPT4.

Non-deterministic Output. In order to produce more natural-sounding text, generative LLMs typically have *temperature* parameter that adds some variability to the output. Because of this, responses given by GPT4 may change even if the input remains constant. At the same time, if the input is varied slightly (e.g., by phrasing the same instruction in a different way), the model may make significant alterations to its response. In our experiments, we attempted to account for this by establishing a certain level of input and output consistency. We found that, although results are overall relatively consistent, there is still considerable variation in how well each response reflects individual aspects of the provided text, for example, whether a particular task has been correctly identified and categorized. We, therefore, argue that future research into the behavior of LLMs and their reaction to different inputs is needed. In particular, the non-deterministic nature of LLM's output has implications for evaluation design: in our opinion, a basic sensitivity analysis as applied in this paper is always required in order to perform a meaningful evaluation of performance.

File Generation. When using it in practice, as illustrated with the three tasks, GPT4 does not generate files but rather text. Therefore, in order to use it in the first two exemplary tasks, further translation into formalized languages was necessary. This can be done via a compiler that generates *Declare* constraints or BPMN models based on the output. Nevertheless, it poses a limitation of current LLMs, especially considering output variability. It should be noted that this limitation is specific to present-day LLMs such as GPT4, which are not capable of file generation and may be overcome by future iterations of the models.

7 Conclusion

In this paper, we developed and applied an approach that utilizes the LLM GPT4 for diverse BPM tasks. The approach itself is simple and leverages the capabilities of GPT4 by instructing it to accomplish the task at hand. We selected three BPM tasks to illustrate that GPT4 is indeed able to accomplish them: mining imperative process models from the textual description, mining declarative process models from the textual description, and assessing RPA suitability of process tasks from textual descriptions. For all three tasks, GPT4 performs similarly to or better than the benchmark, i.e., specific applications for the respective task. We analyzed the input and output robustness of the approach and found that the output is relatively insensitive to different executions of the same prompt, even if different authors formulated them. Further, we found that some prompts should include examples to help the LLM. Future research could assess whether LLMs are also applicable to other tasks from different phases of the BPM lifecycle. All in all, this paper illustrates and evaluates three practical applications of GPT4 and provides implications for future research and usage.

References

1. Bellan, P., van der Aa, H., Dragoni, M., Ghidini, C., Ponzetto, S.P.: PET: an annotated dataset for process extraction from natural language text tasks. In: Cabanillas, C., Garmann-Johnsen, N.F., Koschmider, A. (eds.) BPM 2022. LNBIP, vol. 460, pp. 315–321. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-25383-6_23
2. Bellan, P., Dragoni, M., Ghidini, C.: A qualitative analysis of the state of the art in process extraction from text. In: DP@AI*IA (2020)
3. Bellan, P., Dragoni, M., Ghidini, C.: Extracting business process entities and relations from text using pre-trained language models and in-context learning. In: Almeida, J.P.A., Karastoyanova, D., Guizzardi, G., Montali, M., Maggi, F.M., Fonseca, C.M. (eds.) EDOC 2022. LNCS, vol. 13585, pp. 182–199. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17604-3_11
4. Burattin, A., Maggi, F.M., Sperduti, A.: Conformance checking based on multi-perspective declarative process models. Expert Syst. Appl. **65**, 194–211 (2016)
5. Busch, K., Rochlitzer, A., Sola, D., Leopold, H.: Just tell me: prompt engineering in business process management. Preprint arXiv:2304.07183 (2023)
6. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Introduction to business process management. In: Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A. (eds.) Fundamentals of Business Process Management, pp. 1–33. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-56509-4_1
7. Friedrich, F.: Automated generation of business process models from natural language input. Master thesis. https://frapu.de/pdf/friedrich2010.pdf
8. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 482–496. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21640-4_36
9. Klievtsova, N., Benzin, J.V., Kampik, T., Mangler, J., Rinderle-Ma, S.: Conversational process modelling: state of the art, applications, and implications in practice. Preprint arXiv:2304.11065 (2023)
10. Leopold, H., van der Aa, H., Reijers, H.A.: Identifying candidate tasks for robotic process automation in textual process descriptions. In: Gulden, J., Reinhartz-Berger, I., Schmidt, R., Guerreiro, S., Guédria, W., Bera, P. (eds.) BPMDS/EMMSAD -2018. LNBIP, vol. 318, pp. 67–81. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91704-7_5
11. Mustansir, A., Shahzad, K., Malik, M.K.: Towards automatic business process redesign: an NLP based approach to extract redesign suggestions. Autom. Softw. Eng. **29**, 1–24 (2022)

12. OpenAI: GPT-4 technical report. Preprint arXiv:2304.04309 (2023)
13. Rebmann, A., van der Aa, H.: Extracting semantic process information from the natural language in event logs. In: La Rosa, M., Sadiq, S., Teniente, E. (eds.) CAiSE 2021. LNCS, vol. 12751, pp. 57–74. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79382-1_4
14. Reddy, K.N., Harichandana, U., Alekhya, T., Rajesh, S.: A study of robotic process automation among artificial intelligence. Int. J. Sci. Res. **9**(2), 392–397 (2019)
15. Reijers, H.A., Limam, S., van der Aalst, W.: Product-based workflow design. JMIS **20**(1), 229–262 (2003)
16. Rizun, N., Revina, A., Meister, V.G.: Assessing business process complexity based on textual data: evidence from ITIL IT ticket processing. BPMJ **27**(7), 1966–1998 (2021)
17. Teubner, T., Flath, C.M., Weinhardt, C., van der Aalst, W., Hinz, O.: Welcome to the era of ChatGPT et al. the prospects of large language models. BISE **65**, 95–101 (2023). https://doi.org/10.1007/s12599-023-00795-x
18. van der Aa, H., Carmona, J., Leopold, H., Mendling, J., Padró, L.: Challenges and opportunities of applying natural language processing in business process management. In: COLING, pp. 2791–2801 (2018)
19. van der Aa, H., Di Ciccio, C., Leopold, H., Reijers, H.A.: Extracting declarative process models from natural language. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 365–382. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_23
20. Vidgof, M., Bachhofner, S., Mendling, J.: Large language models for business process management: opportunities and challenges. Preprint arXiv:2304.04309 (2023)

Collecting Activities and States in German Business Process Models

Ralf Laue(✉), Kristin Kutzner, and Martin Läuter

University of Applied Sciences Zwickau, Dr.-Friedrichs-Ring 2a, 08056 Zwickau, Germany
`{Ralf.Laue,Martin.Laeuter}@fh-zwickau.de`

Abstract. In this paper we describe a publicly available dataset containing 6,266 verb phrases which can be used to express activities and states in business process models in German language. This addresses the problem that often activities or states in labels of business process models are not expressed by a single verb but rather by a multiword expression.

Verb phrases that are considered semantically equivalent have been grouped into synsets. This helps to identify the actual meaning of a textual label.

The dataset has been compiled from a comprehensive analysis of 6,711 business process models with German labels and a study of already available collections of multiword expressions in the literature.

The resource can be used for algorithms that analyze business process models with respect to the semantics of their labels.

Keywords: multiword expression · light verb construction · support verb construction · business process model · task labels

1 Introduction

There is extensive research on analyzing business process models (BPM) for various purposes such as preventing and correcting modelling errors, comparing models to each other or querying model repositories. For all these purposes, two aspects need to be taken into account:

First the model logic as a control flow graph can be analyzed and represented in abstract form. But second, to a large degree, the meaning of a BPM is given by the *textual labels* assigned to the model's "boxes and arrows" which describe the tasks, events, decisions, etc. In practice, in most cases, unrestricted natural language is used for those labels which means that the labels are prone to ambiguities, missing information, redundancies and lack uniform style and consistent wording.

Despite (or because?) of these problems, in previous years several researchers worked on algorithms that reason about natural language labels in BPMs [1].

Ms. Kutzner is a former project member.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 466–477, 2024.
https://doi.org/10.1007/978-3-031-50974-2_35

Algorithms have been developed for various purposes such as querying model repositories, comparing BPMs to each other, improving modelling quality or detecting possible modelling errors. The majority of this work deals with task labels, i.e. with the description of the activities taking place when executing the modelled process. While the algorithms serve different purposes, many of them share the application of a tool chain starting with the following steps:

1. Tokenization (segmenting the textual label into words, numbers, abbreviations, etc.),
2. Part-of-speech tagging (classifying the tokens into grammatical categories such as nouns, verbs, adjectives, etc.),
3. Mapping synonyms to so-called synsets (using a lexical resource such as *WordNet* [7]).

In most cases these steps result in synsets for a verb as well as synsets for nouns and other parts of speech. Among those the most significant item for describing the meaning of the label (the actual activity described by the label) is the synset of the verb.

In this article, we consider the following task:

T-MEANING: Given a label of an activity or event in a BPM, find the semantic meaning of a verb or multiword expression containing a verb (see Fig. 1). If different phrases can be used to express the same meaning, T-MEANING should yield the same result for each of them.

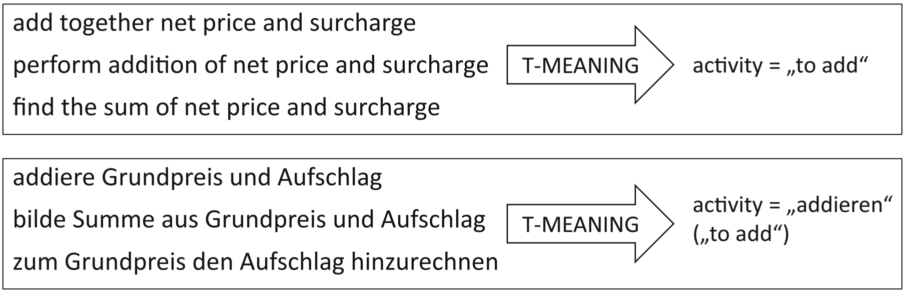

add together net price and surcharge
perform addition of net price and surcharge
find the sum of net price and surcharge
T-MEANING
activity = „to add“
addiere Grundpreis und Aufschlag
bilde Summe aus Grundpreis und Aufschlag
zum Grundpreis den Aufschlag hinzurechnen
T-MEANING
activity = „addieren“
(„to add“)

Fig. 1. Task T-MEANING solved for English and German labels

We want to solve the task T-MEANING irrespective of whether the activity is described by a single verb or by a multiword expression. From the example in Fig. 1 it can be seen that it would be less useful to name the activity of the second and third English label as "to perform" and "to find" (as it would be the result of the classical approach described above).

Solving T-MEANING can serve as a building block for solving more complex tasks, like the ones discussed in Sect. 5.

2 Multiword Expressions

We can see from the example in Fig. 1 that in order to solve T-MEANING we have to provide resources for dealing with two problems: First we have to deal with longer phrases (instead of just a single verb) that describe an activity or state and second we have to group verb phrases with the same meaning together in sets (so-called synsets).

The first issue is related to multiword expressions: German, English and most European languages don't always use just simple verbs to express a state or an activity. Often they use more complex verb phrases to convey the intended meaning – either for paraphrasing a verb, or just due to the fact that there is no verb available in that situation. When using a verb phrase, the meaning of the phrase is either an interplay of the phrase's members, or a new meaning of its own. English has "to make a decision" and "to get in contact", where the main verbs don't contribute most of the meaning, while German has *Bericht erstatten* (= to report), *Angebot einholen* (= to ask for offers) and *geboten sein* (= be imperative).

This phenomenon has been studied under various names such as "support verb construction", "light verb construction", "function verb construction" or (for German language) "Funktionsverbgefüge". The term "light/support verb" refers to the fact that the semantics of such a verb is "either void or reduced to a small set of semantic features that are relevant for very large subclasses of verbs" [14].

There is no generally accepted definition of the term "light verb construction" or the term "Funktionsverbgefüge" in German [11,22]. Existing definitions have in common that they refer to a collocation where a verb – the light verb – is a main verb that has lost most of its concrete lexical semantics. It is not mainly the verb but its collocate that describes the action. To avoid inaccuracy and misunderstandings, we will not use the term "light verb construction" in the remainder of this article. Instead we will refer to the phenomenon described above as a *multiword expression*. This allows us to include phrases such as *Kopie anfertigen* (to make a copy) which would usually not be regarded as light verb construction. However, while this is a normal verb phrase consisting of verb and object, the object explains the action more specifically than the verb, and the answer to T-MEANING should be *kopieren* (= to copy) instead of *anfertigen* (= to fabricate sth.).

For solving T-MEANING, a formal representation of the syntactic structure of a verb phrase is useful. Our approach for such a structure supporting the T-MEANING task for BPM in German language will be discussed in Sect. 4. In the following Sect. 3, we describe the procedure to compile this dataset.

3 Collecting and Analyzing Multiword Expressions

3.1 Collecting and Analyzing Multiword Expressions from the Literature

For compiling a list of multiword task descriptions that are relevant to the business domain we investigated existing collections of such constructions.

Bruker [2] compiled a list of more than 2,000 German multiword expressions for activities from an extensive corpus analysis that included the TIGER corpus (containing German newspaper text) and the DWDS corpus (containing various kinds of texts). Because both corpora are not domain-specific, this list contains a large number of phrases that are not expected to be found in a BPM label (such as "to give a kiss"). We scanned the phrases in the list for those which are relevant to the business domain and included them in our collection.

In addition we included all phrases from a collection of such constructs in the business domain published in [18] where 10 business reports from large German corporate groups had been analyzed. Furthermore all business-related verbs from [27], a compendium of German verbs, have been added.

In addition to this systematic work the first author of this paper collected multiword expressions that describe activities in a business context over a period of 9 years in a non-systematic, but continuous manner. This work was complemented by multiword expressions from lists available on the internet.[1]

3.2 Collecting BPMs with German Activity Labels

In order to identify multiword expressions occurring in BPMs with labels in German language we collected such models from various sources. If the models were available in print only, we created a computer-readable file. We included models using the modelling languages EPC (Event-Driven Process Chains) [25], BPMN and oEPC (Object-Oriented Event-Driven Process Chains) [21].

Altogether we examined 6,711 BPMs, in particular:

- 2,299 Event-Driven Process Chains from the following sources:
  - 604 models from the SAP R/3 reference model [12],
  - 393 models from the repository of the BPM Academic Initiative [13],
  - 349 models from 16 real-world projects,
  - 329 models from 26 textbooks, 52 models from 13 PhD theses, 210 models from 11 bachelor, 3 master and 43 diploma theses and 10 term papers, 83 models from 43 published academic texts,
  - 106 models from 17 various resources found on the internet (not belonging to any of the other categories),
  - 39 models from a German process sharing platform for public administration processes [6] and 38 models from the publicly available repository of the process modelling tool Oryx [3],
  - 22 models from 12 university lecture notes/exam sample solutions and 54 students' solutions to university exercises,
  - 14 models from 3 technical manuals and 6 models that come as examples with ARIS Toolset, a process modelling tool,
- 3,775 BPMN models from the following sources:

---

[1] in particular: https://www.grammatiktraining.de/nomenverb/grammatikliste-nomen-verb-verbindungen.html and https://program.idf.uni-heidelberg.de/fvg/liste.

  - 3537 models from the *BPMN for Research* repository[2],
  - 186 models from the BPMN repository of Swiss public administration[3],
  - 45 models from a textbook [16],
  - 7 models from a master thesis,
- 637 Object-Oriented Event-Driven Process Chains (oEPC) [21] from a modelling project at the Hesse University for Public Management and Security.

3.3 Extracting Verbs and Multiword Expressions from BPMs

From the collected BPMs we compiled a list of both single verbs and multiword expressions that describe activities or states in the models. The procedure of this compilation is described in this subsection.

Pre-processing and Part-of-Speech Tagging. We implemented Python scripts that used the Natural Language Toolkit (NLTK). As a first step we checked whether the collected BPMs contained data-extraction related noise and cleaned the text (e.g. by normalizing character encoding for umlauted vowels). Then the textual labels of tasks and events were extracted and split into tokens by using the NLTK tokenizer package. Based on the tokens we performed lemmatization and part-of-speech (POS) tagging using the Hanover Tagger [29].

Manual Inspection. Next we generated a list of words from the BPM corpus that have been tagged as verbs together with the text passages of BPM element labels in which they occurred. We manually inspected this list to identify and collect multiword expressions, i.e. all those cases where the identified verb alone would not be sufficient to describe the activity properly. This manual inspection was done independently by two authors of his paper. If the authors came to different conclusions on whether a phrase should be included, each case was discussed until consensus was reached.

3.4 Expanding the Range of Verbs

Due to the construction our word list in its current state bore the danger of being overly specific to the word choice in the collected models. In order to alleviate this, we used the publicly available *fastText* word vectors [9] for 300,000 word forms to manually extend our list with additional simple verbs. Said word vectors were constructed such that semantically similar terms can be found in short cosine measure distance to each other. Therefore we executed the following steps in order to collect additional verb forms that have not been found in the BPM but promise to be relevant for the BPM domain anyway due to their proximity to already collected verbs.

---

[2] https://github.com/camunda/bpmn-for-research/.

[3] https://www.ech-bpm.ch.

- Starting with a verb that is already in our collection, we considered all word vectors of inflected forms for this verb in the *fastText* resource. For each of those word vectors we selected the 20 nearest-neighbor word forms based on the cosine measure of their word vectors. For example, starting with *vergrößern* (= to enlarge) already being in the list, we find inflected forms such as *vergrößert* (= enlarged). In the neighborhood of that form we found *vergrössert* (the same word in Swiss-German spelling), *verkleinert* (= reduced in size), *erweitert* (= extended), *vergrößerte* (a conjugation of the initial word), *verringert* (= reduced in number), ...
- After joining the sets of nearest neighbors for all inflected forms of each lemma we obtained 2220 sets of related words.
- We marked verbs that are a) deemed possible to occur in BPM and b) semantically similar to an existing verb, e.g. *vergrößern* – *ausdehnen*, *ausweiten*, *steigern*, *verbreitern*. Those were regarded as candidates for being in the same synset (see Sect. 3.6).

Although this process didn't extend the dataset to more domains than before and added only simple verbs (no multiword expressions), it is now more robust with respect to verb choice.

3.5 Building Syntactic Verb Frames

Multiword expressions describing an activity – as mentioned above – take their meaning from combining a verb with objects, adverbs, prepositional phrases etc. In order to keep track of the relationships of those words we need to leave the word level and formalize the data. For this purpose we defined a markup with the following properties:

- Each entry is given by means of a verb frame with the verb infinitive occupying the head slot. The structure of head and labelled slots is called a syntactic verb frame, cf. *GermaNet* [8].
- We use the following labels, largely following GermaNet: *vfin* (finite verb), *NN* (subject), *AN* (accusative object), *DN* (dative object), *GN* (genitive object), *NG* (subject complement), *B* (adverb), *AR*, *DR* (reflexive pronoun, accusative and dative, resp.), and all prepositions with the corresponding cases they govern. *aus+D* is different from *aus+A*.
- Verbs are rarely standing for themselves, mostly forming a relationship with and between their arguments. For usually occupied roles of simple verbs we have to add their slots to the frame, naming it according to the parts-of-speech with which they are expressed. So most verbs have *NN* slots, and transitive verbs have *AN* slots.
- Multiword expressions have objects and other syntactic participants that are fixed. We represent these fixed participants by filling their slots with the respective word. Slots for usually occupied roles of simple verbs are called *unfilled.*

  Example: geben (= to give) as a simple verb has a *NN* (subject) slot in

addition to *AN* (accusative object) and *DN* (dative object). The multiword expression *Alarm geben* (= to alarm/alert) has a *NN*, but also an *AN* slot filled with *Alarm*. The transitive verb *anschalten* (= to switch on) has both a *NN* slot and an *AN* slot, both unfilled.
- In some cases, it is helpful to mark a slot mandatory to be filled. While we cannot know in advance *which* object is used in an utterance, we want a verb frame only to match *if* there is an object present. Analogously a slot can be marked *not* to be filled, or *forbidden*. For example, if there is an object present in a label, a verb frame with a forbidden object slot should not match. This way we make the verb frames disjoint and disambiguate between different meanings.
  Example: warten (= to service sth.) has a mandatory *AN* slot, while for *warten* (= to wait) the *AN* slot is marked as "forbidden".

All expressions describing an activity (both simple verbs and multiword expressions) were annotated in this data structure, effectively embedding them in a unified data type. The list of verb frames is the basis for the next step – grouping verb frames into synsets.

3.6 Grouping Syntactic Verb Frames into Synsets

In the T-MEANING task discussed in this paper we want to recover the meaning of a verb or verbal phrase in a BPM label. As illustrated in Fig. 1, this should be robust to the concrete choice of words and phrases, e.g. *starten* (= to start) and *beginnen* (= to begin) should be identified as words describing essentially the same activity.

Therefore we grouped the syntactic verb frames into sets of verb frames that are considered semantically equivalent, so-called synsets (Fig. 2). In most cases a verb frame belongs to only one synset, but there are also verbs that are members of more than one synset (e.g. *einführen* can have the meaning of "insert into", but also "introduce sth./so."). This also applies to verbs that can be understood either as a mental process or as the act of communicating the result of this process (e.g. *urteilen* (= to judge/assess, but also to render a judgment)) [30].

Like in other resources (such as [7,10]), we also put synsets into a hierarchy. While a manual process, we again decided by consensus. There are various ways to build a hierarchy reflecting the "is-a-kind-of" relation for synsets (see discussion in [7, p. 71–72]), there is not only one correct way to build such a hierarchy. We have chosen a structure that we hope is most helpful for reasoning about similarity between and within business process models. In this hierarchy, we use one layer (the orange nodes in Fig. 3) for domains such as "Offers", "Money Transfer", "Communication", "Movement", etc. Nodes (i.e. synsets) can have child nodes, which leads to a multi-level hierarchy. For example, "Money Transfer" has child nodes "to pay" and "to receive payment", and "to pay" in turn has one child node for "to make a down payment", a synset containing verb frames such as *anzahlen* or *eine Anzahlung leisten*.

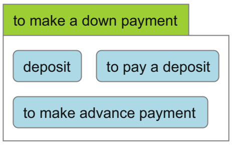


Fig. 2. Synsets (green) contain verb frames (blue) (Color figure online)

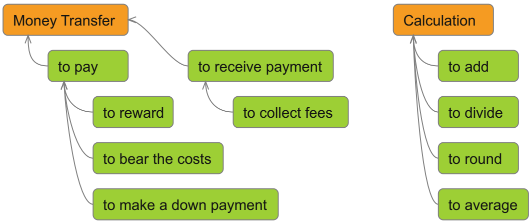


Fig. 3. Hierarchy of synsets (Color figure online)

A differentiation has to be made between verb frames denoting activities on the one hand and states on the other hand. However, using "to act" and "to be" as roots in the hierarchy turned out to have disadvantages (see discussion in [7, p. 70–72]). Instead, the information whether a verb frame is describing an activity or a state is given by an additional attribute in our list: Activities such as *unterschreiben* (= to sign) have the attribute TYPE = "act", while states such as *genügen* (= to suffice) have the attribute TYPE = "be". In addition, we decided to introduce another attribute value TYPE = "become" for verb frames describing a state change, e.g. *enden* (= to come to an end).

4 Resulting Data

Our dataset contains 6,266 verb frames, among them 2,942 multiword expressions. They are grouped into 1,782 synsets. For 1,184 of those synsets, we found at least one occurrence in a label in our model collection.[4]

The multiword expressions occurring most frequently in the BPM in our collection can be found in Table 1.

Table 1. Frequent Multiword Expressions

Angebot einholen	ask for offers
Versicherung abschließen	to contract insurance
Bestellung aufgeben	place an order
Antrag stellen	file an application
Scoring durchführen	perform scoring
Anfrage erhalten	receive a request
Bedarf liegt vor	a need exists
Geld erhalten	receive money
Auftrag vergeben	to award a contract

---

[4] The mapping of label texts to verb frames and synsets has been done by means of a computer program. This means that the actual number of occurrences could slightly differ from the given number, e.g. because of errors in the automatic mapping of inflected verbs to their infinitives.

5 Possible Fields of Application

Mendling et al. [19] name 25 challenges for semantic process modelling, and we think that the consideration of multiword expressions can improve the results for most of them. The following examples of possible application areas and the references are by no means complete:

- *Testing the compliance of labels with a grammatical guideline* (sometimes also the generation of suggestions for changing a label to a more desirable style) [17],
- *Classifying Actions* into verb classes and attaching graphical icons in order to improve the understanding of a BPM [20],
- *Finding possible errors or ambiguities in a model* [15],
- *Generating natural language text* from BPMs [17],
- *Merging models*, e.g. in the context of company mergers [24],
- *Calculating similarity measures* between BPM, e.g. for querying model repositories [5].

All but the first task not only benefit from the list of multiword expressions and verb frames, but also from the associated synsets.

In our experiments we realized that relying on *GermaNet* [10] – the German version of *WordNet* – would not be sufficient when dealing with multiword expressions. While certain multiword expressions are included in *GermaNet*, this is not the main focus of that project and expressions such as *auf die Tagesordnung setzen* (= to put sth. on the agenda) are missing. *VerbNet* [26] also deals with mapping syntactical verb frames to verb classes, but for the English language. It uses a relatively small number of classes ("borrow", "receive" and "select" share the same class, as do "sign (agreement)", "handpaint" and "annotate") and handles mostly every-day verbs. Furthermore they only cover compound verbs and *not* multiword phrases (e.g. "make a decision", "take for granted", "take a photo" etc. are not listed). So for solving T-MEANING we had to extend the scope substantially.

At the same time modern Machine Learning approaches surprisingly prove less effective. We used multi-language (incl. German) versions of BERT [4], SBERT [23] and LLaMA (a Large Language Model from Meta AI) [28] to provide embeddings for all labels in our repository of German BPMs. Then we looked at the nearest neighbors of the vector of each embedding. We found that BERT and SBERT yielded good environments for cosines larger than 0.9, but these environments consisted of only few labels, though much more labels with the same action (cf. T-MEANING) were available in our repository. LLaMA's embeddings showed that an LLM is trained to continue sentences, so the environments consisted of mainly labels that had the same syntactic sentence structure, dominating the semantics (e.g. action) by a huge factor. Solutions with classical dependency parsers building on our dataset can help cover these weak spots, provide the data source for fine tuning of Deep Learning models and be an alternative in scenarios where large neural networks cannot be installed due to resource constraints, e.g. in local installations.

6 Limitations

The collection of both verb frames and synsets is tailored towards the business domain, so it would have to be amended when used with other domains or even unspecific text. Targeting T-MEANING definitely has shaped the structure of the synset tree. While still usable for solving other tasks in the business domain, the data lacks useful annotations and relationships between synsets, such as "is-not", "is-reverse", "other-direction" for different kind of antonyms and "is-redo" for doing an action again. In order to handle the arguments of an action, a mapping from the verb frame slots to slots of synsets would be a necessary addition.

7 Publicly Available Dataset

The list of multiword expressions and synsets is available at https://github.com/bflowtoolbox/VerbframesDE.

It is our aim to further improve the described resource, e.g. by improving both coverage and specificity of synsets.

Of course, the same kind of study should be repeated for other languages, most importantly for English. Although our work was executed on a set of business process models in German language, the issue addressed in this paper is relevant for BPMs with labels in other languages as well.

Acknowledgements. Our research project was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project 445156547.

References

1. van der Aa, H., Carmona, J., Leopold, H., Mendling, J., Padró, L.: Challenges and opportunities of applying natural language processing in business process management. In: Proceedings of the 27th International Conference on Computational Linguistics 2018, pp. 2791–2801. Association for Computational Linguistics (2018)
2. Bruker, A.: Funktionsverbgefüge im Deutschen. Bachelor+Master Publ. (2013)
3. Decker, G., Overdick, H., Weske, M.: Oryx – an open modeling platform for the BPM community. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 382–385. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85758-7_29
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186. Association for Computational Linguistics, Minneapolis (2019)
5. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring similarity between semantic business process models. In: Conceptual Modelling 2007, pp. 71–80 (2007)
6. Eid-Sabbagh, R.-H., Kunze, M., Weske, M.: An open process model library. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 100, pp. 26–38. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28115-0_4

7. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press, Cambridge (1998)
8. GermaNet: Verb frames. https://uni-tuebingen.de/en/faculties/faculty-of-humanities/departments/modern-languages/department-of-linguistics/chairs/general-and-computational-linguistics/ressources/lexica/germanet/description/verbs/verb-frames/#c1081857
9. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018) (2018)
10. Hamp, B., Feldweg, H.: GermaNet - a lexical-semantic net for German. In: Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications (1997)
11. Harm, V.: Funktionsverbgefüge des Deutschen: Untersuchungen zu einer Kategorie zwischen Lexikon und Grammatik. De Gruyter (2021)
12. Keller, G., Detering, S.: Process-oriented modeling and analysis of business processes using the R/3 reference model. In: Bernus, P., Nemes, L. (eds.) Modelling and Methodologies for Enterprise Integration. ITIFIP, pp. 69–87. Springer, Boston, MA (1996). https://doi.org/10.1007/978-0-387-34983-1_5
13. Kunze, M., Berger, P., Weske, M.: BPM academic initiative - fostering empirical research. In: Demonstration Track of the 10th International Conference on Business Process Management 2012. CEUR Workshop Proceedings, vol. 940, pp. 1–5 (2012)
14. Langer, S.: A formal specification of support verb constructions. In: Langer, S., Schnorbusch, D. (eds.) Semantik im Lexikon, Tübinger Beiträge zur Linguistik, vol. 479, pp. 179–201. Gunter Narr Verlag, Tübingen (2004)
15. Laue, R., Koop, W., Gruhn, V.: Indicators for open issues in business process models. In: Daneva, M., Pastor, O. (eds.) REFSQ 2016. LNCS, vol. 9619, pp. 102–116. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30282-9_7
16. Laue, R., Koschmider, A., Fahland, D. (eds.): Prozessmanagement und Process-Mining - Grundlagen. De Gruyter Studium. De Gruyter (2020)
17. Leopold, H.: Natural Language in Business Process Models - Theoretical Foundations, Techniques, and Applications. LNBIP, vol. 168. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-04175-9
18. Marušić, B.: Funktionsverbgefüge in deutscher Konzernsprache. Ph.D. thesis, University of Osijek, Croatia (2015)
19. Mendling, J., Leopold, H., Pittke, F.: 25 challenges of semantic process modeling. Int. J. Inf. Syst. Softw. Eng. Big Companies (IJISEBC) **1**(1), 78–94 (2015)
20. Mendling, J., Recker, J., Reijers, H.A.: On the usage of labels and icons in business process modeling. Int. J. Inf. Syst. Model. Des. **1**(2), 40–58 (2010)
21. Nüttgens, M., Feld, T., Zimmermann, V.: Business process modeling with EPC and UML: transformation or integration? In: The Unified Modeling Language - Technical Aspects and Applications 1997, pp. 250–261. Physica, Heidelberg (1997)
22. van Pottelberge, J.: Funktionsverbgefüge und verwandte Erscheinungen. In: Burger, H., Dobrovolskij, D., Kühn, P., Norrick, N.R. (eds.) Phraseologie, vol. 1, pp. 436–444. De Gruyter Mouton (2008)
23. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2020)
24. Rosa, M.L., Dumas, M., Uba, R., Dijkman, R.M.: Business process model merging: an approach to business process consolidation. ACM Trans. Softw. Eng. Methodol. **22**(2), 11:1–11:42 (2013)

25. Scheer, A.W.: Business Process Engineering: Reference Models for Industrial Enterprises. Springer, Heidelberg (1994). https://doi.org/10.1007/978-3-642-79142-0
26. Schuler, K.K., Korhonen, A., Brown, S.W.: VerbNet overview, extensions, mappings and applications. In: Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, Boulder, Colorado, USA, 31 May–5 June 2009, Tutorial Abstracts, pp. 13–14. The Association for Computational Linguistics (2009)
27. Schumacher, H. (ed.): Verben in Feldern. Walter de Gruyter (1986)
28. Touvron, H., et al.: LLaMA: open and efficient foundation language models. Preprint arXiv:2302.13971 (2023)
29. Wartena, C.: A probabilistic morphology model for German lemmatization. In: 15th Conference on Natural Language Processing (2019)
30. Winkler, E.: Der Wortschatzausschnitt der deutschen Kommunikationsverben - eine empirische Bestandsaufnahme des Wortbestandes. In: Harras, G., Proost, K., Winkler, E. (eds.) Handbuch deutscher Kommunikationsverben. Teil 2: Lexikalische Strukturen, pp. 25–71. de Gruyter (2007)

1st International Workshop of Object-centric Processes from A to Z (OBJECTS 2023)

Object-centric Processes from A to Z 1st International Workshop (OBJECTS 2023)

The topic of object-centric processes has been gaining momentum in the last few years, with many works addressing foundational and practical problems on the interplay of processes and objects, where behaviour arises from the complex interplay among multiple business objects and their one-to-many/many-to-many relationships. Despite the surging number of results on the topic, many related problems have not yet been addressed. One such problem relates to correct modelling and analysis of such processes, where a suitable trade-off between expressiveness and feasibility of analytic techniques must be found. Another problem concerns how object-centric processes can be handled in Process Mining (PM), including novel, efficient PM techniques and suitable event data/log formats to operate over complex event data for such processes and fully unleash the insights hidden therein. The OBJECTS workshop provides a platform for researchers from the fields of Business Process Management and Process Mining who work on object-centric processes where they can share their ideas and current research addressing the aforementioned problems as well as discuss challenges and future directions of the field.

OBJECTS 2023 is the first edition of the workshop, and attracted 11 high-quality international submissions. Each paper was single-blind reviewed by at least three members of the Program Committee. Of all the submitted manuscripts, the top five were accepted for presentation, four of them as long contributions. These four papers are published in the proceedings.

Adams and van der Aalst presented a detailed analysis of the problems of convergence, divergence and deficiency that may arise when extending Process Mining towards the support of object-centricity. The authors also proposed concrete solutions for case extraction in order to avoid the above problems. Galanti and de Leoni provided an empirical study of four predictive analytic techniques on event logs coming from two real-life object-centric processes. The study focused on the prediction performance of such techniques and addressed questions related to their pragmatics. Calegari and Delgado studied the application of Model-Driven Engineering (MDE) to the Object-Centric Event Data (OCED) metamodel, and investigated the alignment of the Business Process and Organizational Data Integrated Metamodel (BPODIM) with OCED through a model transformation, discussing the associated challenges. Lomazova et al. proposed a generalization of object-centric Petri nets with a new mechanism of arc weight parameterization via linear combinations of type-dependent weight variables. The authors also demonstrated that the obtained class of Petri nets is expressively equivalent to a classical P/T-net.

The workshop concluded with the invited talk of Dirk Fahland titled “From Cases to Objects and Relations – and Beyond”. Dirk gave an overview of the core concepts of “object-centric” process modeling and Process Mining and discussed how subtle adaptations of earlier ideas led to this simple but powerful “sweet spot”. The talk also reviewed which new analysis use cases and insights are enabled by object-centric concepts and how they can be generalized in simple ways to consider process dynamics beyond data objects or “object-centric process executions”, and provided insights on research challenges around object-centric Process Mining, highlighting its value for Business Process Management.

We thank the authors for their interesting contributions and the members of the Program Committee for their invaluable help in the reviewing and discussion phase of the manuscripts. We hope that, by reading these papers, the reader will know more about the latest advances in research on object-centric processes.

September 2023

Marco Montali
Andrey Rivkin
Jan Martijn van der Werf

Organization

Workshop Chairs

Marco Montali	Free University of Bozen-Bolzano, Italy
Andrey Rivkin	Technical University of Denmark, Denmark
Jan Martijn van der Werf	Utrecht University, The Netherlands

Program Committee

Han van der Aa	University of Mannheim, Germany
Wil van der Aalst	RWTH Aachen University, Germany
Johannes De Smedt	KU Leuven, Belgium
Rik Eshuis	Eindhoven University of Technology, The Netherlands
Peter Fettke	German Research Center for Artificial Inteilligence (DFKI) and Saarland University, Germany
Irina Lomazova	HSE University, Russia
Xixi Lu	Utrecht University, The Netherlands
Jan Mendling	Humboldt-Universität zu Berlin, Germany
Giovanni Meroni	Technical University of Denmark, Denmark
Jorge Munoz-Gama	Pontificia Universidad Católica de Chile, Chile
Artem Polyvyanyy	University of Melbourne, Australia
Stefanie Rinderle-Ma	Technical University of Munich, Germany
Michael Rosemann	Queensland University of Technology, Australia
Natalia Sidorova	Eindhoven University of Technology, The Netherlands
Monique Snoeck	KU Leuven, Belgium
Pnina Soffer	University of Haifa, Israel
Gerd Wagner	Brandenburg University of Technology, Germany
Moe Thandar Wynn	Queensland University of Technology, Australia
Francesca Zerbato	University of St. Gallen, Switzerland

Typed Petri Nets with Variable Arc Weights

Irina A. Lomazova[1], Alexey A. Mitsyuk[1], and Andrey Rivkin[2](✉)

[1] Faculty of Computer Science, HSE University, Moscow, Russia
{ilomazova,amitsyuk}@hse.ru
[2] Department of Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark
ariv@dtu.dk

Abstract. Object-centric processes have become increasingly popular in the last years mainly due to the establishment of object-centric process mining. One of the most popular formalisms for describing lifecycles of objects and capturing relationships between them are object-centric Petri nets. An important feature of such nets is the ability to transfer an arbitrary number of same-typed objects upon transition firing by means of so-called variable arcs. In this work, we generalise the concept of variable arcs by introducing a fairly simple and versatile mechanism of arc weight parameterization via linear combinations of type-dependent weight variables, and incorporating it into the new formalism of typed Petri nets with variable arc weights. Moreover, we demonstrate that such extended variable arcs can be effectively eliminated, making the resulting net model expressively equivalent to a classical P/T-net. This result allows a natural transfer of analytical techniques available for P/T-nets to the formalisms like object-centric Petri nets.

1 Introduction

Traditional workflow nets often focus on a single case in isolation. However, in reality, the notion of a case is often more complex and may consist of multiple simultaneously "active" objects with complex inter-relations. This issue has been already recognized and addressed in various works many years ago. Among such, the seminal work on workflow patterns [4] with the related multiple instance patterns, and works on proclets [1,2], where interactions between multiple workflows are studied.

The ideas of object-centricity had been recently revisited in the context of object-centric process mining [3], where a new class of object-centric Petri nets was introduced. In a nutshell, such nets can be seen as a "weak variant" of colored Petri nets [10] in which tokens can be only compared by their types. Moreover, such nets are equipped with variable arcs—a special type of transfer arcs that can move an arbitrary number of same-typed objects upon transition firing. Variable arcs become immediately useful in the contexts when local one-to-many synchronizations must be modeled (e.g., putting multiple items from an order into a package).

In this paper, we propose a generalization of object-centric Petri nets, called *typed Petri nets with variable arc weights*, by allowing for special arc expressions in which variables, as opposed to colored Petri nets, are not instantiated with tokens but with arc weights. Like that, one can see such nets as those enabling arc weight parameterization. This parameterization is realized via linear combinations, which, in turn, allow us

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 483–495, 2024.
https://doi.org/10.1007/978-3-031-50974-2_36

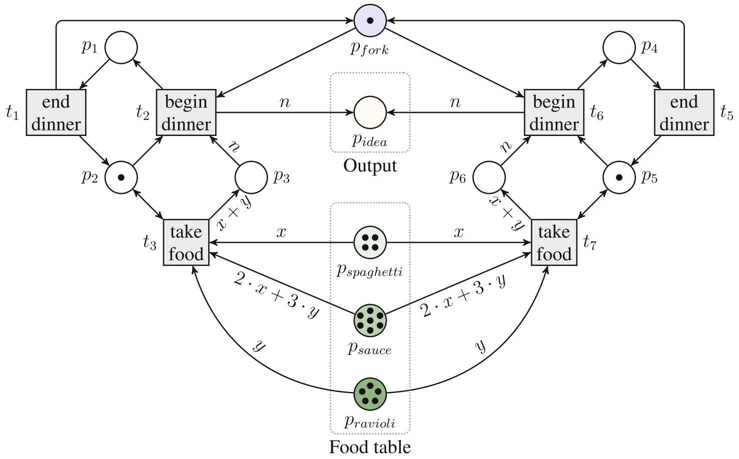


Fig. 1. Two hedonist philosophers eating spaghetti and ravioli

to model more complex dependencies between manipulated objects. For example, by writing $3 \cdot x + 3$ we require that items should be shopped in bundles of three and at least 3 items should be added into a shopping cart. We a bit diverge from the nomenclature used by W. van der Aalst and A. Berti in their seminal paper on object-centric Petri nets [3] (where variable arcs denote arcs allowing for a simultaneous transfer of a not known a priori number of tokens), and refer to our arcs with such expressions as arcs with variable weights. We formally show that arcs with variable weights can be effectively eliminated, resulting in weakly bisimilar typed Petri nets. We observe that such nets are as expressive as classical Petri nets, which naturally paves the way to further transfer of existing analytic results.

The rest of the paper is organized as follows. In Sect. 2, we provide two motivational examples, discussing the modelling power of typed Petri nets. Then, in Sect. 3, we formally define such nets and, in Sect. 5, demonstrate our main theoretical results. Finally, the paper concludes with a discussion on related work (Sect. 6) and future work (Sect. 7).

2 Two Motivating Examples

In this section, we present two examples showcasing the modeling power of typed Petri nets with variable arc weights.

The model in Fig. 1 is inspired by the well-known dining philosophers problem defined by E. Dijkstra and eventually formulated T. Hoare [9]. Here we extended the classical setting by allowing philosophers to choose the dishes to eat. To eat, a philosopher has to not only grab forks or chopsticks, but also take a certain number of servings of food from the common table.

Our hedonist philosophers eat spaghetti and ravioli with sauce. The amount of sauce depends on the chosen dish, namely two servings of sauce for a serving of spaghetti and three for a serving of ravioli. To dine, a philosopher adds a few servings of food to his/her plate. While dining a philosopher generates new ideas. The more food is consumed, the more ideas are produced – an idea for every serving of food.

It is easy to think about the aforementioned servings (as well as ideas and forks) as some sort of objects and distinguish such objects by types. In Fig. 1, this is done by assigning unique object types (represented as colors) to places. With variables on arcs, we specify correspondence between different types of food, and between amounts of food and ideas. Variables x, y and n are used as numeric parameters of transition firings. Thus, firing transition ‘take food’ for $x = 3, y = 2$ consumes three servings of spaghetti and two servings of ravioli and $2 \cdot 3 + 3 \cdot 2 = 12$ servings of sauce, producing $2 + 3 = 5$ dishes to eat in place p_3. Transitions like ‘take food’ that manipulate objects of different types are de facto synchronization transitions that may also produce new objects. For example, ‘take food’ produces in place p_3 objects that represent “complete dishes” such as spaghetti with a sauce.

The main purpose of the above example is to show how one can use variable arc weight expressions to model resource dependencies. Therefore, for ease of presentation, we model only two philosophers who eat with one fork. The example can easily be extended to model more complex communication scenarios between philosophers.

The model in Fig. 2 is based on the popular delivery example from [3], which illustrates object-oriented process modeling concepts. This is a generalized workflow net with input places $p_1, \ldots, p_4$ and output places $p_{16}, p_{18}, p_{19}, p_{15}$. In this example, the delivery process is as follows. Customers order goods, goods are packed and delivered by tracks, and orders are completed when paid by customers. Thus, there are objects of four types (highlighted in different shades of green): bills, customer orders, packages, and delivery tracks. Here also, variable arc weight expressions are used to specify correspondence between numbers of objects involved in the transition firing. Thus, for example, each truck has to be loaded with exactly 6 packages using transition ‘load’. Notice that here the synchronization happens between different types of objects, and the variables on adjacent arcs do not act according to the standard token-to-variable binding semantics (as it would be done in colored Petri nets [10]), but instead indicate *how many objects* will be passed through each of the arcs. We can also use the arc expressions not to mix up the objects upon their synchronization. For example, for ‘process payments’, we use y to account for orders and x to account for bills, since an order can be split in several bills, and a bill can include several orders.

These examples illustrate modeling facilities of variable arcs in object-centric settings. Next, we give the precise definition of typed P/T-nets with variable arc weights.

3 Preliminaries

Multisets, Petri Nets. Given a set B, a multiset m over B is the mapping of the form $m : B \to \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers inluding zero. In what follows, we use $\mathbb{N}_{>0}$ to denote $\mathbb{N}\setminus\{0\}$. Given an element $b \in B$, $m(b) \in \mathbb{N}$ denotes the number of times b appears in the multiset. We write b^n if $m(b) = n$. Given two multisets m_1 and m_2 over B: *(i)* $m_1 \subseteq m_2$ iff $m_1(b) \leq m_2(b)$ for each $b \in B$, also $m_1 \subset m_2$ iff

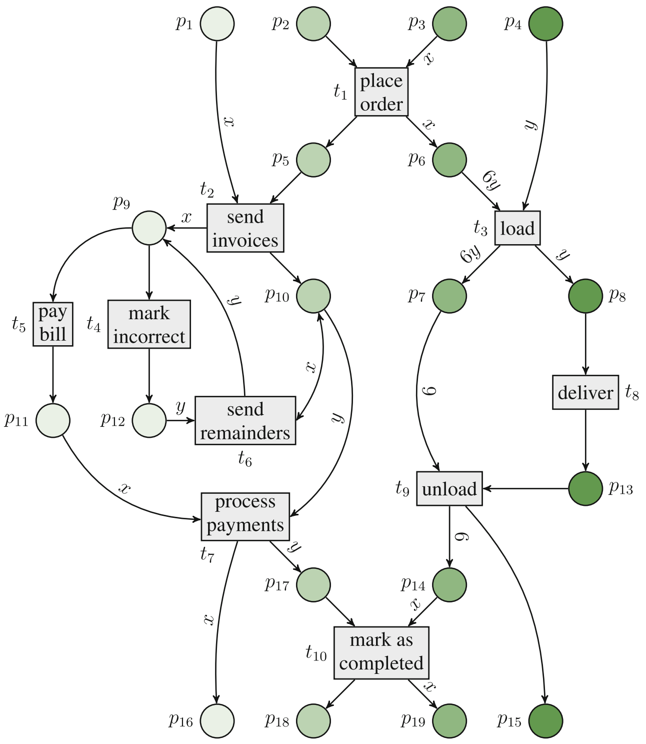


Fig. 2. Many trucks delivering multiple ordered and paid packages. Here we use the following object types: bill, order, package, truck. (Color figure online)

$m_1 \subseteq m_2$ and $m_1 \neq m_2$; *(ii)* $(m_1 + m_2)(b) = m_1(b) + m_2(b)$; *(iii)* if $m_1 \subseteq m_2$, $(m_2 - m_1)(b) = m_2(b) - m_1(b)$; *(iv)* $|m| = \sum_{b \in B} m(b)$.

Given two disjoint sets P and T of places and transitions respectively, multiset $F : (P \times T) \cup (T \times P) \to \mathbb{N}$ and injective function $\ell : T \to A \cup \{\tau\}$, where A is a finite set of *activities* (activity names) and τ is a special symbol for silent transitions, we call $N = (P, T, F, \ell, A)$ a *labeled place/transition net* (or a labeled P/T-net).

P/T nets are represented as bipartite directed graphs, where places are indicated by circles, transitions by rectangles, and arrows represent the flow relation F. The notions of marking, transition enablement and firing for classical P/T nets are defined as usual.

Transition Systems, Bisimulation Equivalence. Given a finite set of activities A together with a special silent label τ, a *(labeled) transition system* (LTS) is a tuple $\Gamma = (S, A, s_0, \to)$, where S is a set of *states*, s_0 is an *initial state* and $\to \subset (S \times (A \cup \{\tau\}) \times S)$ is a *transition relation*. In what follows, we write $s \xrightarrow{a} s'$ instead of $(s, a, s') \in \to$.

Given $a \in A$, $p \overset{a}{\dashrightarrow} q$ denotes a *weak transition relation* that is defined as follows:

– $p \overset{a}{\dashrightarrow} q$ iff $p(\xrightarrow{\tau})^* q_1 \xrightarrow{a} q_2 (\xrightarrow{\tau})^* q$ for some states q_1, q_2;

- $p \overset{\tau}{\dashrightarrow} q$ iff $p(\xrightarrow{\tau})^* q$.

Here, $(\xrightarrow{\tau})^*$ denotes the reflexive and transitive closure of $\xrightarrow{\tau}$.

Definition 1 (Weak bisimulation). *Let* $\Gamma_1 = (S_1, A, s_{01}, \rightarrow_1)$ *and* $\Gamma_2 = (S_2, A, s_{02}, \rightarrow_2)$ *be two labeled transition systems. Then relation* $R \subseteq (S_1 \times S_2)$ *is called a* weak bisimulation *between* Γ_1 *and* Γ_2 *iff for every pair* $(p, q) \in R$ *and* $a \in A \cup \{\tau\}$ *the following holds:*

(i) if $p \xrightarrow{a}_1 p'$*, then there exists* $q' \in S_2$ *such that* $q \overset{a}{\dashrightarrow}_2 q'$ *and* $(p', q') \in R$*;*
(ii) if $q \xrightarrow{a}_2 q'$*, then there exists* $p' \in S_1$ *such that* $p \overset{a}{\dashrightarrow}_1 p'$ *and* $(p', q') \in R$*.* ◁

A state $p \in S_1$ is weakly bisimilar to $q \in S_2$, written $p \approx q$, if there exists a weak bisimulation R between Γ_1 and Γ_2 such that $(p, q) \in R$. Finally, Γ_1 is weakly bisimilar to Γ_2, written $\Gamma_1 \approx \Gamma_2$, if $s_{01} \approx s_{02}$, i.e. their initial states are weakly bisimilar.

4 The Formalism

In this section, we define typed P/T-nets with variable arc weights (*TVAW-nets* for short), which allow transitions to have different modes of firing depending on the values of the variables in the arc expressions. It can be considered as a generalization of object-centric Petri nets (OC-nets), introduced in [3] to describe process models discovered using object-centric process mining techniques. As opposed to classical P/T-nets, the main features of OC-nets are typed places and variables arcs. The latter allow to transfer multiple tokens (the actual number of transferred tokens is defined upon transition firing) between two places of the same type. TVAW-nets extend OC-nets by replacing variable arcs with arcs labeled with linear expressions that semantically go beyond the limit of transferring a variable number of tokens between same-typed places only.

TVAW-nets can be thought of as a special version of colored Petri nets (CPNs), where each color type contains exactly one color. Thus, color types coincide with colors, and tokens of the same type are indistinguishable. Similar to CPNs, places in TVAW-nets are labeled with color types to indicate the types of tokens they can carry. Since the tokens in a place are indistinguishable, arc expressions are used to define number of tokens consumed/produced by transition firings.

Following [3], colors in TVAW-nets are called *object types*. Hereinafter, we use $\mathbb{O}$ to denote a set of object types. As in [3], TVAW-nets allow to use the concept of *variable arcs* – a special type of arcs for transferring a non-deterministic number of tokens from an input place to an output place by firing a transition. We generalize the notion of variable arcs so that now each arc is assigned a linear variable expression. Like that, transition firing becomes parameterized by variables and, hence, the mode of firing is determined by evaluation of variables. This makes it possible not only to transfer any number of tokens, but also to specify the number of consumed/produced tokens as a linear combination of parameters plus a given natural number, not less than a given number. Moreover, now it is possible to specify dependencies between numbers of consumed/produced tokens regardless of their type. This can be seen as a way of defining "local" (i.e., transition-level) cardinal constraints between manipulated objects.

Formally, given a set of variables $\mathcal{V}$, we denote by $\mathcal{E}$ the set of expressions of the form $a_1 \cdot \mu_1 + \cdots + a_k \cdot \mu_k + b$, where $a_1, \ldots, a_k, b \in \mathbb{N}$ and $\mu_1, \ldots, \mu_k \in \mathcal{V}$. We suppose that $\mathcal{E}$ includes also a special *empty* expression ϵ, which is used to make the labeling function total on all pairs of nodes. Labeling a pair of nodes with ϵ means there is no arc between these nodes. Alternatively we could consider partial labeling functions.

Expressions from $\mathcal{E}$ are interpreted over natural numbers in the standard way. When all coefficients $a_1, \ldots, a_k$ are equal to zero, the value of $\varepsilon = a_1 \cdot \mu_1 + \cdots + a_k \cdot \mu_k + b$ coincides with $b \in \mathbb{N}$. Thus, natural numbers are a special case of expressions in $\mathcal{E}$.

Definition 2 (TVAW-net). *A* typed P/T-net with variable arc weights $\mathcal{N}$ *is a tuple* $(\mathbb{O}, P, T, F, \mathtt{type}, \ell, A)$*, where:*

(i) $\mathbb{O}$ *is the set of object types;*
(ii) P *and* T *are finite sets of places and transitions, s.t.* $P \cap T = \emptyset$*;*
(iii) $\mathtt{type} : P \to \mathbb{O}$ *is a place typing function;*
(iv) $F : (P \times T) \cup (T \times P) \to \mathcal{E}$*;*
(v) *for every* $p \in P$, $t \in T$, *if output arc expression* $\varepsilon = F(t, p)$ *contains variable* μ, *then there is* at least one *input for t place* $p' \in P$ *s.t.* $F(p', t)$ *contains* μ *as well;*
(vi) $\ell : T \to A \cup \{\tau\}$ *is an injective function mapping transitions to process activities (from the set of activities* A*), where* τ *denotes a silent (invisible) activity.* $\triangleleft$

The meaning of arcs with variable weights (*VW-arcs* for short) in TVAW-nets remains in line with [3]: in particular, they can be used to model the transfer of several objects from one place to another. Condition *(v)* in the above definition is a weakened version of the consistent variability condition from [3], where if a transition t has two variable arcs (or, in the case of TVAW-nets, two arcs labeled with μ) connecting it to places of some type d, then t must not have other places of type d adjacent to it. We only require that if variable μ occurs in an arc expression for an outgoing arc for t, then μ should occur in at least one incoming for t arc expression. Note that, unlike the definition in [3], we allow a variable amount of tokens to be consumed and not transferred. Also, arc expressions can include multiple variables and thus define relatively complex linear relationships between the numbers of consumed and produced tokens.

Since the tokens residing in the same place cannot be distinguished, the TVAW-net marking is defined similarly to the marking for classical P/T-nets. More specifically, a *marking* of $\mathcal{N} = (\mathbb{O}, P, T, F, \mathtt{type}, \ell, A)$, is a function $M : P \to \mathbb{N}$. When $M(p) = n$ and $n > 0$ for some $p \in P$, we say that place p contains n objects of type $\mathtt{type}(p)$ in state M. We write $\langle \mathcal{N}, M \rangle$ to denote TVAW-net $\mathcal{N}$ marked with M and use symbol M_0 to define the initial marking of the net. For ease of notation, we also denote with $[p_1^{i_1}, \ldots, \mathrm{p}_n^{i_n}]$ a concrete multiset representing a marking in which each place p_k contains i_k tokens, and with $\mathbb{M}^P$ denote the set of all markings defined on top of P.

Let us now specify the net dynamics. Let $\mathcal{N} = (\mathbb{O}, P, T, F, \mathtt{type}, \ell, A)$ be a TVAW-net. As customary, given $x \in P \cup T$, we use ${}^\bullet x := \{y \mid F(y, x) \neq 0\}$ to denote the *preset* of x and $x^\bullet := \{y \mid F(x, y) \neq 0\}$ to denote the *postset* of x. For ease of notation, $T_\mathcal{V}$ denotes the set of all transitions $t \in T$ with adjacent VW-arcs. For expression $\varepsilon \in \mathcal{E}$, by $\mathit{Vars}(\varepsilon)$ we denote the set of all variables occurring in ε, and for transition $t \in T$, by $\mathit{Vars}(t)$ – the set of all variables occurring in adjacent to t arc expressions.

Since a transition can have multiple incoming and outgoing VW-arcs with arc expressions depending on different variables, we would like to be able to explicitly identify how many tokens from each place adjacent to the VW-arc are consumed/produced by a concrete transition firing. To this end, we introduce a *binding* function $\alpha : \mathcal{V} \rightarrow (\mathbb{N}\backslash\{0\})$ specifying the *mode of firing*. For each variable $\mu \in \mathcal{V}$, binding α defines values of variables in arc expressions to determine the number of tokens involved in the transition firing. For an arc expression $\varepsilon = a_1 \cdot \mu_1 + \cdots + a_k \cdot \mu_k + b$, we define its value under binding α as $\varepsilon(\alpha) = a_1 \cdot \alpha(\mu_1) + \cdots + a_k \cdot \alpha(\mu_k) + b$. Obviously, when ε is a natural number, its value does not depend on α.

We then say that transition t is *enabled* in marking M under binding α, written $M[t, \alpha\rangle$, iff, for every $p \in {}^\bullet t$, $M(p) \geq F(p,t)(\alpha)$. When t is enabled in marking M under binding α, it may *fire*, yielding new marking M' that is defined for every $p \in P$ as $M'(p) = M(p) - F(p,t)(\alpha) + F(t,p)(\alpha)$. We denote this as $M[t, \alpha\rangle M'$ and assume that the definition is inductively extended to sequences $\sigma \in (T \times (\mathcal{V} \rightarrow (\mathbb{N}\backslash\{0\})))^*$ of transition firings. We say that M' is *reachable* from M if there exists $\sigma \in (T \times (\mathcal{V} \rightarrow (\mathbb{N}\backslash\{0\})))^*$, s.t. $M[\sigma\rangle M'$. For a TVAW-net $\mathcal{N}$, we write $\mathcal{R}(\mathcal{N}, M)$ to denote the set of all markings of $\mathcal{N}$ reachable from its marking M. For TVAW-nets without arcs with variable weights (or when a particular transition is not in $T_{\mathcal{V}}$), we omit the binding in all the related notions.

The execution semantics of a TVAW-net can be captured with a possibly infinite-state LTS that accounts for all possible executions starting from the initial marking. Formally, TVAW-net $\mathcal{N} = (\mathbb{O}, P, T, F, \mathtt{type}, \ell, A)$ with initial marking M_0 *induces* a labeled transition system $\Gamma_{\mathcal{N}} = (S, A, s_0, \rightarrow)$, where:

- $S = \mathcal{R}(\mathcal{N}, M_0)$ and $s_0 = M_0$;
- for $M, M' \in S$ it holds that: $M \xrightarrow{a} M'$ iff $M[t, \alpha\rangle M'$, for some $t \in T$ s.t. $\ell(t) = a$ and some binding α.

5 Elimination of Arcs with Variable Weights

Now, after we have defined the syntax and semantics of TVAW-nets, we would like to observe that every TVAW-net $\mathcal{N}$ can be replaced with a behaviorally equivalent TVAW-net $\mathcal{N}'$ that does not contain arcs with variable weights. To this end, we provide a variable arc elimination algorithm and show that its output is always weakly bisimilar to its input. The weak bisimilarity is conditioned by the fact that in order to correctly represent the behavior of $\mathcal{N}$, $\mathcal{N}'$ includes additional intermediate transition firings that are, however, invisible and therefore not relevant to the comparison of net behavior. Finally, we say that two marked TVAW-nets $\langle\mathcal{N}, M_0\rangle$ and $\langle\mathcal{N}', M_0'\rangle$ are *weakly bisimilar* (and denote it as $\langle\mathcal{N}, M_0\rangle \approx \langle\mathcal{N}', M_0'\rangle$) if for transition systems $\Gamma_{\mathcal{N}}$ and $\Gamma_{\mathcal{N}'}$ they respectively induce, it holds that $\Gamma_{\mathcal{N}} \approx \Gamma_{\mathcal{N}'}$.

To access coefficients of a given expression $\varepsilon = a_1 \cdot \mu_1 + \cdots + a_k \cdot \mu_k + b$, we define its variable projection as follows:

$$\varepsilon|_{\mu} = \begin{cases} a_i, & \text{for } \mu \in \mathit{Vars}(\varepsilon) \text{ and } \mu = \mu_i \\ 0, & \text{for } \mu \notin \mathit{Vars}(\varepsilon) \end{cases}$$

Whenever $F(x, y) = 0$, for $x, y \in P \cup T$, it means that components x and y of the net graph are simply not connected by any arc.

Definition 3 (Arc elimination). *Let* $\langle \mathcal{N}, M_0 \rangle$ *be a marked TVAW-net, where* $\mathcal{N} = (\mathbb{O}, P, T, F, \mathtt{type}, \ell, A)$ *and* $T_\mathcal{V} \neq \emptyset$*. Define a TVAW-net* $\langle \mathcal{N}', M_0' \rangle$*, where* $\mathcal{N}' = (\mathbb{O}' \cup \{\bot\}, P', T', F', \mathtt{type}', \ell', A)$*, as follows:*

(i) $\bot \notin \mathbb{O}$ *is a fresh object type,* $\mathbb{O}' = \mathbb{O} \cup \{\bot\}$*;*

(ii) Construct a new place lock such that:
  - $F'(lock, t) = 1$*, for every* $t \in T_\mathcal{V}$*;*
  - $F'(lock, t') = F'(t', lock) = 1$*, for every* $t' \in T \backslash T_\mathcal{V}$*.*

(iii) For each $t \in T_\mathcal{V}$ *and* $\mu \in$ *Vars*(t)*, construct a new place* $lock_t$*, and new transitions* $unlock_t$ *and* $add_{t,\mu}$ *s.t.:*
  - $F'(t, lock_t) = 1$*,* $F'(lock_t, unlock_t) = 1$ *and* $F'(lock_t, add_{t,\mu}) = F'(add_{t,\mu}, lock_t) = 1$*;*
  - *for each* $p \in {}^\bullet t$ *and* $q \in t^\bullet$*,* $F'(p, add_{t,\mu}) = F(p,t)|_\mu$*,* $F'(add_{t,\mu}, q) = F(t,q)|_\mu$*.*

(iv) For each $t \in T_\mathcal{V}$*, let* $Vars(t) = \{\mu_i | 1 \leq i \leq k\}$*. Then, for each* $p \in {}^\bullet t$ *and* $q \in t^\bullet$*,* $F'(p,t) = \sum_{j=1}^{k} F(p,t)|_{\mu_j} + F(p,t)|_\mathbb{N}$ *and* $F'(t,q) = \sum_{j=1}^{k} F(t,q)|_{\mu_j} + F(t,q)|_\mathbb{N}$*.*

(v) $P' = P \cup \{lock\} \cup \bigcup_{t \in T_\mathcal{V}} \{lock_t\}$*.*

(vi) $T' = T \cup \bigcup_{t \in T_\mathcal{V}} (\{unlock_t\} \cup \{add_{t,\mu} \mid \mu \in Vars(t)\})$*.*

(vii) for $p \in P$*,* $F'(p,t) = F(p,t)|_\mathbb{N}$*,* $F'(t,q) = F(t,q)|_\mathbb{N}$*, if* $t \in T_\mathcal{V}$*, and* $F'(p,t) = F(p,t)$*,* $F'(t,q) = F(t,q)$*, if* $t \in T \backslash T_\mathcal{V}$*.*

(viii) $\mathtt{type}'(p) = \mathtt{type}(p)$*, if* $p \in P$*, and* $\mathtt{type}'(p) = \bot$*, otherwise.*

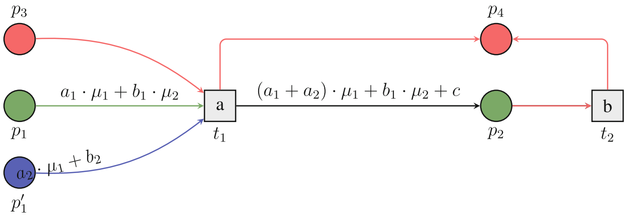
(ix) $\ell'(t) = \ell(t)$*, if* $t \in T$*, and* $\ell(t) = \tau$*, otherwise.*

(x) $M_0'(p) = M_0(p)$*, if* $p \in P$*,* $M_0(lock) = 1$ *and* $M_0(p) = 0$*, otherwise.*

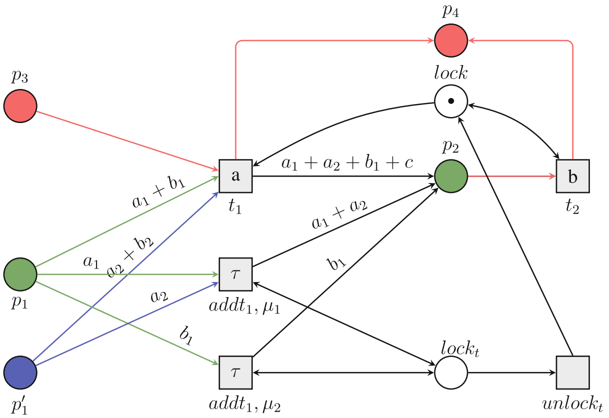
We say that a TVAW-net $\langle \mathcal{N}', M_0' \rangle$ *is obtained from* $\langle \mathcal{N}, M_0 \rangle$ *by VW-arcs elimination.* $\lhd$

Informally, the above elimination procedure can be described as follows. Given a generic TVAW-net $\mathcal{N}$, it constructs a behaviorally equivalent TVAW-net $\mathcal{N}'$ without VW-arcs. The latter is obtained by first copying the original net structure for all the net components that are not related via VW-arcs, and then by creating sub-nets representing non-deterministic loops that allow to model a sort of a "lossy" consumption and production of tokens for each transition t that has adjacent VW-arcs. Such loop is organized for each involved variable. To prevent other transitions to interleave within this process, the global lock place $lock$ is used. The loop for variable μ requires extra transitions $add_{t,\mu}$, and relies on a local lock place $lock_t$. Moreover, the loops work under the assumption that at least one token is transferred via related VW-arcs (this assumption is in line with the definition of TVAW-net semantics).

We demonstrate how this construction works in more detail on a simple net in Fig. 3a. By applying the elimination procedure to the net, we obtain a new TVAW-net (see Fig. 3b) that has no VW-arcs and that behaves exactly like the input one. Notice that the original net in Fig. 3a has one variable transition t_1. After applying the elimination procedure, this transition is replaced with a sub-net that works as follows. First, using


p_3
p_4
$a_1 \cdot \mu_1 + b_1 \cdot \mu_2$
a
$(a_1 + a_2) \cdot \mu_1 + b_1 \cdot \mu_2 + c$
b
p_1
t_1
p_2
t_2
$a_2 \cdot \mu_1 + b_2$
p_1'

(a) A simple TVAW-net with VW-arcs



(b) The same net but without variable arcs

Fig. 3. Demonstration of the VW-arc elimination procedure from Definition 3 applied to a TVAW-net $\mathcal{N}$ (3a) with a VW-arc-free TVAW-net $\mathcal{N}'$ (3b) as output. Object types are provided as colors.

t_1, the net enters into a critical section that is guarded by the special lock place $lock$. Notice that this lock is global to the whole net and is used to guard firing of every transition from the original net. By firing t_1, the net also performs one complete "transfer" of tokens from ${}^\bullet t_1$ to p_2, which amounts to consuming $(F(p_1, t_1)|_{\mu_1} + F(p_1, t_1)|_{\mu_2})$–tokens from p_1 and $(F(p_1', t_1)|_{\mu_1} + F(p_1', t_1)|_{\mathbb{N}})$–tokens from p_1' respectively, and placing $(F(t_1, p_2)|_{\mu_1} + F(t_1, p_2)|_{\mu_2} + F(t_1, p_2)|_{\mathbb{N}})$–tokens into p_2. Like that, the net is guaranteed to produce at least one token in p_2. The net then may continue transferring tokens from the same places using transitions add_{t_1,μ_1} and add_{t_1,μ_2} (these transitions are guarded by a local lock $lock_{t_1}$). Notice that the token transfer can only happen if p_1 (resp., p_1') contains at least as many tokens as can be used to correctly bind the expression $F(p_1, t_1)$ (resp., $F(p_1', t_1)$).

The token transfer is performed non-deterministically by firing add_{t_1,μ_1} and add_{t_1,μ_2} in any order, and concludes by firing transition $unlock_{t_1}$, which releases the global lock by removing the token from $lock_{t_1}$ and placing a token back to $lock$. This example leads us to the following statement.

Theorem 1. *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked TVAW-net, where $\mathcal{N} = (\mathbb{O}, P, T, F, \mathtt{type}, \ell, A)$. Then application of the elimination procedure (Definition 3) to $\langle \mathcal{N}, M_0 \rangle$ produces a marked TVAW-net $\langle \mathcal{N}', M_0' \rangle$ such that $\langle \mathcal{N}, M_0 \rangle \approx \langle \mathcal{N}', M_0' \rangle$.* $\triangleleft$

Proof. By applying the elimination procedure to $\langle\mathcal{N}, M_0\rangle$, we obtain $\langle\mathcal{N}', M_0'\rangle$, where $\mathcal{N}'$ contains no VW-arcs.

Let $R \subseteq \mathcal{R}(\mathcal{N}, M_0) \times \mathcal{R}(\mathcal{N}', M_0')$ be a binary relation defined on reachable markings of $\mathcal{N}$ and $\mathcal{N}'$ such that $(M, M') \in R$ iff either of the following conditions holds:

- $M(p) = M'(p)$ for all $p \in P$, $M'(\mathit{lock}) = 1$, and $M'(\mathit{lock}_t) = 0$ for all $t \in T_\mathcal{V}$;
- for some $t \in T_\mathcal{V}$, $M(q) = M'(q)$ for all $q \in P \backslash ({}^\bullet t \cup t^\bullet)$, $M'(\mathit{lock}) = 0$ and $M'(\mathit{lock}_t) = 1$, and there exist natural numbers $n_1, \ldots, n_k$, where $k = |\mathit{Vars}(t)|$, such that each $n_i \in \mathbb{N}_{>0}$ and, for each $p \in {}^\bullet t \cup t^\bullet$, it holds that $M'(p) = M(p) - \big(\sum_{i=1}^k n_i \cdot F(p,t)|_{\mu_i} + F(p,t)|_\mathbb{N}\big) + \big(\sum_{i=1}^k n_i \cdot F(t,p)|_{\mu_i} + F(t,p)|_\mathbb{N}\big)$.

To show that $\langle\mathcal{N}, M_0\rangle$ and $\langle\mathcal{N}', M_0'\rangle$ are bisimilar, we need to prove that R is a weak bisimulation between $\varGamma_\mathcal{N} = (\mathcal{R}(\mathcal{N}, M_0), A, s_0, \rightarrow)$ and $\varGamma_{\mathcal{N}'} = (\mathcal{R}(\mathcal{N}', M_0'), A, s_0', \rightarrow')$ by testing both conditions from Definition 1.

(i) Let $(M_1, M_2) \in R$ and $M_1 \xrightarrow{\ell(t)} M_1'$, i.e. $M_1[t, \alpha\rangle M_1'$ with $t \in T$ and $\alpha : \mathcal{V} \rightarrow \mathbb{N}$. Now consider the following two cases.
  - Let $t \in T \backslash T_\mathcal{V}$. Then, there exists $M_2' \in \mathcal{R}(\mathcal{N}', M_0')$ s.t. $M_2[t\rangle M_2'$, $M_2'(p) = M_1'(p)$ for all $p \in P$, $M_2'(\mathit{lock}) = 1$ and $M_2'(\mathit{lock}_t) = 0$ for all $t \in T_\mathcal{V}$. Thus, $M_2 \xrightarrow{\ell(t)} M_2'$ and $(M_1', M_2') \in R$.
  - If $t \in T_\mathcal{V}$, then by the firing rule, for $p \in {}^\bullet t \cup t^\bullet$, $M_1'(p) = M_1(p) - F(p,t)(\alpha) + F(t,p)(\alpha)$, and $M_1'(q) = M_1(q)$, for $q \in P \backslash ({}^\bullet t \cup t^\bullet)$. By definition of R, either $M_2(w) = M_1(w)$ for all $w \in P$, or $M_2(q) = M_1(q)$ for all $q \in P \backslash ({}^\bullet t \cup t^\bullet)$, and for $p \in {}^\bullet t \cup t^\bullet$, $M_2(p) = M_1(p) - \big(\sum_{i=1}^k n_i \cdot F(p,t)|_{\mu_i} + F(p,t)|_\mathbb{N}\big) + \big(\sum_{i=1}^k n_i \cdot F(t,p)|_{\mu_i} + F(t,p)|_\mathbb{N}\big)$. Let V be a set of some variables from $\mathit{Vars}(t)$ (that is, $V \subseteq \mathit{Vars}(t)$). Then, in the latter case, $(\alpha(\mu_j) - 1)$ silent firings of each add_{t,μ_j} for $\mu_j \in V$(remember that, by construction in Definition 3, t performs the first mandatory transfer of tokens from ${}^\bullet t$ to $t^\bullet$) followed by the firing of unlock_t produce M_2' s.t. $M_2'(p) = M_1'(p)$ for all $p \in P$, $M_2'(\mathit{lock}) = 1$ and $M_2'(\mathit{lock}_t) = 0$. The case, when $M_2(\mathit{lock}) = 0$ and $M_2(\mathit{lock}_t) = 1$, is handled similarly.

(ii) Let now $(M_1, M_2) \in R$ and $M_2[t\rangle M_2'$, with $t \in T'$. Here we consider three cases.
  - If $t \in T \backslash T_\mathcal{V}$, then, similarly to **(i)**, with the same transition t, $M_1[t\rangle M_1'$, where $M_1'(p) = M_2'(p)$, for all $p \in P$, and for $q \in P' \backslash P$, $M_2'(q) = M_1'(q)$.
  - If $t \in T_\mathcal{V}$, then by construction of $\mathcal{N}'$ and definition of R, we have for $p \in {}^\bullet t \cup t^\bullet$, $M_2(p) = M_1(p) - \big(\sum_{i=1}^k n_i \cdot F(p,t)|_{\mu_i} + F(p,t)|_\mathbb{N}\big) + \big(\sum_{i=1}^k n_i \cdot F(t,p)|_{\mu_i} + F(t,p)|_\mathbb{N}\big)$, where $k = |\mathit{Vars}(t)|$ and each $n_k \in \mathbb{N}_{>0}$. Then setting $\alpha(\mu_i) = n_i$ for $1 \leq i \leq k$, we get $M_1[t, \alpha\rangle M_1'$, where $M_1'(w) = M_2'(w)$ for all $w \in P$, $M_2'(\mathit{lock}) = 1$ and $M_2'(\mathit{lock}_t) = 0$.
  - If $t \in \{\mathit{add}_{t,\mu} \mid \mu \in \mathit{Varst}\}$, we follow the same reasoning as in the previous case by considering sequences of firings of (some of the) $\mathit{add}_{t,\mu}$ transitions. The case when $t = \mathit{unlock}_t$ is handled similarly. ⊣

Corollary 2. *For each TVAW-net, a weakly bisimilar P/T-net can be effectively constructed.* ◁

Proof. Notice that the procedure from Definition 3 is polynomial, and thus, given a TVAW-net $\mathcal{N}$, its VW-arc-free variant $\mathcal{N}'$ can be effectively computed. According to Theorem 1, such N' is weakly bisimilar to $\mathcal{N}$. Since place typing in a TVAW-nets does not affect the behavior of the net, it can be omitted, resulting in a regular P/T-net. $\dashv$

The above result implies that analysis/verification for any TVAW-net can be effectively reduced to analysis/verification for TVAW-nets without VW-arcs coinciding with classical P/T-nets. Thus, the wide variety of techniques developed for P/T-nets are also applicable to TVAW-nets and, in particular, to OC-nets [3].

6 Related Work

The idea of adding parameters to Petri nets to make models more compact and readable is not quite new. Different kinds of Petri nets with different types of parameterization have been studied in the literature.

In Colored Petri nets (CPNs) [10], places and tokens are typed, and arcs carry expressions with variables that can be considered as parameters [6]. Transitions can fire in different modes, and a mode of firing is defined by evaluation of variables in adjacent arc expressions. Note that in CPNs, variables are assigned tokens as values (and such tokens carry data that can be potentially compared), and values of arc expressions are multisets of tokens consumed, or produced by the transition firing. In contrast to this, in TVAW-nets, tokens are indistinguishable, variables are assigned natural numbers, and arc expression values are also numeric weights. As the consequence of this difference, TVAW-nets are expressively equivalent to conventional P/T-nets, while CPNs (without finite color domains) are Turing-powerful.

In [7], the authors considered Parametric Petri nets (PPNs) with parameterized markings and arc weights to represent several systems by one model. In these nets, parametric values can be assigned to arcs, but not expressions. [5] and [12] similarly introduced and studied Petri nets with parameterized markings.

[14] studied nets with enabling and inhibitor arcs, with guard expression assigned to them. Guards are represented as sets of functions with parameters. These functions are evaluated to determine which transitions are enabled. However, in such nets tokens can only be consumed and produced via regular arcs.

In ω Petri nets (ωPNs) proposed in [8], some arcs can be labelled with a special ω-symbol. An ω-arc can consume/produce an arbitrary number of tokens from/to its input/output place when the corresponding transition fires. This concept is somewhat similar to λ-labelled variable arcs in Object-Centric nets (OC-nets) from [3]. The main difference, however, lies in the following: when an OC-net transition fires, an arbitrary number of tokens is not just consumed/produced as in ωPNs, but transferred via λ-arcs from one input place to one output place, and such places must be of the same type.

In typed P/T-nets with variable arc weights introduced in this paper, transition firing can be better fine-tuned. Namely, parameters/variables in arc expressions allow us to relate tokens consumed and produced in different places using linear dependencies.

7 Conclusions

In this work, we proposed a formalism of Petri nets with typed places and generalized variable arcs, called here a typed P/T-net with variable arc weights (TVAW-nets for short). This formalism can be seen as a generalization of the one of object-cenrtic nets studied in [3]. We demonstrate that, behaviourally, the nets with generalized variable arcs are behaviourally equivalent to nets without such arcs, and, eventually, classical P/T-nets. This seemingly straightforward result paves the road towards better understanding of soundness and its decidability for TVAW-nets. In particular, we may define the object-centric variant of soundness similarly to [13] proviso one crucial difference: TVAW-nets cannot compare identifiers of typed tokens they carry. As it has been discussed in [11], the inability of object-centric nets to compare identifiers allows us to define similar notions of object-centric soundness where object lifecycels have to be structurally extracted from and joined with the original net models, and show that such soundness property is actually decidable. In our future work, we plan to build upon the ideas from [11], formally define a variant of object-centric soundness for TVAW-nets and demonstrate its decidability.

Acknowledgment. The work of Irina A. Lomazova and Alexey A. Mitsyuk was supported by the Basic Research Program of the HSE University.

References

1. van der Aalst, W.M.P., Barthelmess, P., Ellis, C.A., Wainer, J.: Workflow modeling using proclets. In: Scheuermann, P., Etzion, O. (eds.) CoopIS 2000. LNCS, vol. 1901, pp. 198–209. Springer, Heidelberg (2000). https://doi.org/10.1007/10722620_20
2. van der Aalst, W.M.P., Barthelmess, P., Ellis, C.A., Wainer, J.: Proclets: a framework for lightweight interacting workflow processes. Int. J. Cooperative Inf. Syst. **10**(4), 443–481 (2001)
3. van der Aalst, W.M.P., Berti, A.: Discovering object-centric Petri nets. Fundam. Informaticae **175**(1–4), 1–40 (2020). https://doi.org/10.3233/FI-2020-1946
4. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow patterns. Distrib. Parallel Databases **14**(1), 5–51 (2003). https://doi.org/10.1023/A:1022883727209
5. Chiola, G., Donatelli, S., Franceschinis, G.: On parametric P/T nets and their modelling power. In: Proceeding of ICATPN, pp. 206–227 (1991)
6. Christensen, S., Mortensen, K.H.: Parametrisation of coloured Petri nets. DAIMI report series (521) (1997)
7. David, N., Jard, C., Lime, D., Roux, O.H.: Discrete parameters in Petri nets. In: Devillers, R., Valmari, A. (eds.) PETRI NETS 2015. LNCS, vol. 9115, pp. 137–156. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19488-2_7
8. Geeraerts, G., Heußner, A., Praveen, M., Raskin, J.: ω-Petri nets: algorithms and complexity. Fundam. Informaticae **137**(1), 29–60 (2015)
9. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall (1985)
10. Jensen, K., Kristensen, L.M.: Coloured Petri Nets - Modelling and Validation of Concurrent Systems. Springer, Heidelberg (2009). https://doi.org/10.1007/b95112
11. Lomazova, I.A., Mitsyuk, A.A., Rivkin, A.: Soundness in object-centric workflow Petri nets. CoRR abs/2112.14994 (2021)

12. Marsan, M.A., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G.: Modelling with generalized stochastic Petri nets. SIGMETRICS Perform. Eval. Rev. **26**(2), 2 (1998)
13. van der Werf, J.M.E.M., Rivkin, A., Polyvyanyy, A., Montali, M.: Data and process resonance - identifier soundness for models of information systems. In: Bernardinello, L., Petrucci, L. (eds.) PETRI NETS 2022. LNCS, vol. 13288, pp. 369–392. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06653-5_19
14. Zhu, R., Ban, Q., Cong, X.: Modules of Petri nets and new Petri net structure: arcs with a weighted function set. Adv. Mech. Eng. **11**(3), 1–12 (2019)

Addressing Convergence, Divergence, and Deficiency Issues

Jan Niklas Adams(✉) and Wil M. P. van der Aalst

Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany
{niklas.adams,wvdaalst}@pads.rwth-aachen.de

Abstract. The application of process mining algorithms to event logs requires the extraction of cases, describing end-to-end runs through the process. When extracting cases for object-centric event data, this extraction is often subject to convergence, divergence, and deficiency issues. Recently, connected-components extraction was proposed, extracting graph-based cases, called *process executions*, from the graph of event precedence constraints. This paper shows that only case extraction based on connected-components is free of convergence, divergence, and deficiency issues. This proof has several implications for future research in object-centric process mining. First, if a downstream process mining task is negatively affected by these quality issues, connected-components extraction is the only way to mitigate these. Second, additional requirements that would conflict with connected-components extraction would render the mitigation of quality issues infeasible, making trade-offs between quality issues necessary. Third, as traditional event logs are a special case of object-centric event logs and connected-components extraction is equivalent to the traditional case concept for a traditional event log, new extraction techniques, as well as object-centric adaptations of algorithms, should be backward-compatible.

Keywords: Object-Centric Process Mining · Event Data · Flattening

1 Introduction

In most information systems, events are documented in relation to multiple entities, or objects [8]. Process mining algorithms require input in the form of cases, which are sets of events with precedence constraints. When each event is connected to precisely one object, case extraction is trivial as every object defines its own case [1]. However, when multiple objects are linked to a single event, extraction becomes problematic. For example, extracting one case per object results in event duplication when an event is associated with two objects.

Previous research has identified three quality problems when extracting cases from event data with multiple objects per event (object-centric event data): convergence, deficiency, and divergence [2,8,11]. These quality problems can lead to challenges in downstream process mining tasks due to cases containing inaccurate data.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 496–507, 2024.
https://doi.org/10.1007/978-3-031-50974-2_37

Adams et al. have proposed connected-components process execution extraction [5]. This extraction merges all events and precedence constraints connected through common objects into graph-based cases, called *process executions*. In this paper, we show that connected-components extraction is free of convergence, divergence, and deficiency issues and that all extraction techniques not based on connected components do not meet the requirements of being convergence, divergence, and deficiency-free. We situate these findings with respect to the fields of object-centric and traditional process mining and discuss the implications of our proof for the field of process mining.

2 Event Data

A sequence of length $n \in \mathbb{N}$ is a function $\sigma : \{1, \ldots, n\} \rightarrow X$. We denote a sequence with $\sigma = \langle x_1, \ldots, x_n \rangle$. Sequences can be concatenated, denoted by $\langle x_1, \ldots, x_n \rangle \cdot \langle y_1, \ldots, y_m \rangle = \langle x_1, \ldots, x_n, y_1, \ldots, y_m \rangle$. A sequence $\delta_{sub} = \langle y_1, \ldots, y_m \rangle$ is a subsequence of $\delta_{sup} = \langle x_1, \ldots, x_n \rangle$ if the complete δ_{sub} can be mapped to consecutive indices of δ_{sup}, i.e., $\exists_{i \in \{0, \ldots, n-m\}} \, \forall_{j \in \{1, \ldots, m\}} x_{i+j} = y_j$. We denote this by $\delta_{sub} \in \delta_{sup}$. The powerset of a set X defines the set of all possible sets and is denoted by $\mathcal{P}(X)$. The power set without the empty set is denoted by $\mathcal{P}^+(X)$. A directed graph is a tuple $G = (E, K)$ of nodes E and edges $K \subseteq E \times E$. A path between two nodes $e, e' \in E$ describes a sequences of nodes that are connected by edges $e \sim_G e' = \langle e_1, \ldots, e_n \rangle$ such that $e_1 = e \wedge e_n = e' \wedge \forall_{1 \leq i < n} (e_i, e_{i+1}) \in K$. We abbreviate the existence of a path with $e \sim_G e'$. If there is no path between two nodes $e, e' \in E$ then $e \nsim_G e'$. We explicitly define the path from a node to itself as $e \sim_G e = \langle e \rangle$.

Lemma 1 (Connecting Edge). *Let $G = (E, K)$ be a graph and $e, e' \in E$ be two nodes. There is a set of events $E' \subseteq E$ with e' being part of the set and e not being part of the set: $e \in E \setminus E' \wedge e' \in E'$. If there exists a path between e and e' then there is an edge that connects the two sets $E \setminus E'$ and E': $e \sim_G e' \Rightarrow \exists_{(e_1, e_2) \in K} \, e_1 \in E \setminus E' \wedge e_1 \notin E' \wedge e_2 \in E' \wedge e_2 \notin E \setminus E'$.*

Proof. The path between $e, e' \in E$ is $e \sim_G e' = \langle e_1, \ldots, e_n \rangle$ such that $e_1 = e \wedge e_n = e' \wedge \forall_{1 \leq i < n} (e_i, e_{i+1}) \in K$. Each node of E is either part of E' or $E \setminus E'$. We prove by contradiction: If there would not exist an edge that connects both sets, all elements in the path would need to be of the same set: $\neg \exists_{\langle e_1, e_2 \rangle \in e \sim_G e'} \, e_1 \in E \setminus E'' \wedge e_2 \in E'' \Rightarrow \{e_1, \ldots e_n\} \subseteq E \setminus E' \vee \{e_1, \ldots e_n\} \subseteq E'$. However, $e_1 \in E \setminus E'$ and $e_n \in E'$, i.e., there is at least on element of both sets in $e \sim_G e'$. Therefore, $\exists_{(e_1, e_2) \in K} \, e_1 \in E \setminus E' \wedge e_1 \notin E' \wedge e_2 \in E' \wedge e_2 \notin E \setminus E'$.

Lemma 2 (Transitivity). *Let $G = (E, K)$ be a graph and $e, e' \in E$ be two nodes with $e \nsim_G e'$ with $E_1 = \{e'' \in E \mid e \sim_G e'' \vee e'' \sim_G e\}$ and $E_2 = \{e'' \in E \mid e' \sim_G e'' \vee e'' \sim_G e'\}$. The reachable nodes from e do not overlap with the reachable nodes from e': $E_1 \cap E_2 = \emptyset$.*

Proof. We prove this by contradiction. Assume there is an event $e'' \in E$ such that $e'' \in E_1 \wedge e'' \in E_2$. By construction of E_1 and E_2, it holds that $e \sim_G e''$

or $e'' \sim_G e$ and $e' \sim_G e''$ or $e'' \sim_G e'$. Therefore, there exists a path $e \sim_G e'$ by concatenating these two paths which conflicts the initial statement.

An event log consists of events. Each event is identified through an element from the universe of event identifiers $\mathcal{E}$. An event describes the execution of an activity at a given time for affected objects. The universe of activities is denoted by $\mathcal{A}$, the universe of timestamps is denoted by $\mathcal{T}$, and the universe of objects is denoted by $\mathcal{O}$. Each object is of a type from the universe of types $\mathcal{OT}$. This type is given by the typing mapping $\pi_{type} : \mathcal{O} \rightarrow \mathcal{OT}$.

Definition 1 (Event Log). *An event log is a tuple* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *consisting of*

- *events* $E \subseteq \mathcal{E}$, *objects* $O \subseteq \mathcal{O}$, *object types* $OT = \{\pi_{type}(o) \mid o \in O\}$,
- *event-object associations* $\pi_{obj} : E \rightarrow \mathcal{P}(O)$,
- *event-activity mappings* $\pi_{act} : E \rightarrow \mathcal{A}$,
- *event-timestamp mappings* $\pi_{time} : E \rightarrow \mathcal{T}$.

Each object can be associated with multiple events. The sequence in which events related to the same object occur establishes the precedence constraints of the event log. By merging together all precedence constraints for all objects we derive a global view of the events and how they are connected via objects' precedence constraints. As one event can be related to multiple objects, this global view is a graph.

Definition 2 (Event-Object Graph). *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log. We define the events and precedence constraints of an event log in one single object: the event-object graph* $EO_L = (E, P)$. *The nodes of the event-object graph are the events of the event log, the edges are the precedence constraints defined by the objects* $P = \{(e, e') \in E \times E \mid e \neq e' \land \exists_{o \in O}\ o \in \pi_{obj}(e) \land o \in \pi_{obj}(e') \land \neg\exists_{e'' \in E}\ o \in \pi_{obj}(e'') \land \pi_{time}(e) < \pi_{time}(e'') < \pi_{time}(e')\}$.

To analyze a process, we extract cases from the event log that describe end-to-end runs through the process. In traditional process mining, this is trivial as a case identifier directly identifies an end-to-end run through the process. In the more general case (i.e., object-centric event logs), events do not refer to one case identifier but to multiple case identifiers, i.e., objects. Case extraction refers to the notion of assigning events and their precedence constraints to different cases.

Definition 3 (Case Extraction). *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log with event-object-graph* $EO_L = (E, K)$. *To analyze the process, cases are extracted. An extraction technique* $ext(L) \subseteq \mathcal{P}^{+}(E)$ *extracts cases as sets of events, where events and precedence constraints are defined by cases* $C_{ext}(L) = \{(E', K') \mid E' \in ext(L) \land K' = E' \times E' \cap K\}$.

The result of a case extraction are, in general, graphs. In traditional process mining, cases are assumed to be sequences, however, sequences are also special cases of graphs [10]. Different case extraction methods have been proposed and discussed. Van der Aalst introduces the extraction method of flattening on a single object type, i.e., taking the objects of one object type and using their event sequences as cases for traditional process mining [2], which has also been discussed in earlier publications addressing the data extraction from information systems with object-centricity [9]. Adams et al. propose two methods, one that uses connected components of the event-object graph and one that uses connected subgraphs of a leading object type [5], also representing the cases as graphs instead of sequences. Furthermore, Calvanese et al. describe an extraction technique that would be equivalent to the connected-components extraction of Adams et al. and squashing the resulting graph into a sequence [7]. We discuss the different extraction techniques in Sect. 6.

3 Quality Issues

Extracted cases can be subject to different quality problems. These quality problems can affect downstream process analysis by providing misleading statistics, incorrect process models, and missing information. The three quality problems are convergence, deficiency, and divergence [2].

Definition 4. *(Convergence) Let* $L{=}(E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log with event-object graph* $EO_L{=}(E, K)$ *and cases* $C_{ext}(L){=}\{(E_1, K_1), \ldots (E_n, K_n)\}$. $C_{ext}(L)$ *is convergence-free iff* $\forall_{e \in E}\ \neg\exists_{(E_i,K_i),(E_j,K_j) \in C_{ext}(L)}$ $(E_i, K_i) \neq (E_j, K_j) \wedge e \in E_i \wedge e \in E_j$.

If an event is contained in two cases there is a convergence problem. Duplicated events lead to issues in downstream-process mining, such as increased activity counts that trigger problems in other areas like feature engineering in predictive process monitoring.

Definition 5. *(Deficiency) Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log with event-object graph* $EO_L = (E, K)$ *and extracted cases* $C_{ext}(L) = \{(E_1, K_1), \ldots (E_n, K_n)\}$. *The cases are deficiency-free iff* $\forall_{e \in E} \exists_{(E_i,K_i) \in C_{ext}(L)}\ e \in E_i$.

If an event is contained in no case there is a deficiency problem. Missing events lead to missing information in downstream process mining as events that could contain important hints about activities, bottlenecks, or features are not considered.

Definition 6. *(Divergence) Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log with event-object graph* $EO_L = (E, K)$ *and extracted cases* $C_{ext}(L) = \{(E_1, K_1), \ldots (E_n, K_n)\}$. *The extracted cases are divergence-free iff* $\forall_{(e,e') \in E \times E}\ (e, e') \in K \Leftrightarrow \exists_{(E_i,K_i) \in C}\ (e, e') \in K_i$.

If the precedence constraints contained in the cases do not match the precedence constraints of the event log there is a divergence problem. This means, that the directly-follows relationships that are contained in the event log do not reflect the ones that are represented by the cases. By our definition of case extraction, the cases cannot contain precedence constraints that are not in the event log. Removed constraints will lead to issues with downstream process mining tasks such as discovery, as the resulting model will be based on incorrect precedence constraints.

4 Process Executions from Connected Components

In this paper, we investigate the properties of connected-components process execution extraction introduced by Adams et al. [5] with respect to quality issues. Process executions are graph-based cases that are built by using the weakly connected components of the event-object graph, i.e., merging all interdependent objects and their precedence constraints into one case.

Definition 7 (Connected-Components Extraction). *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log with event-object graph* $EO_L = (E, K)$*. The connected-component extraction extracts graph-based cases (called: process executions) as connected components of the event-object graph, i.e.,* $ext\_cc(L) = \{E' \in \mathcal{P}^{+}(E) \mid e, e' \in E' \Leftrightarrow e \sim_{EO} e'\}$*.*

All events that are connected to each other via a path in the event-object graph are grouped into one process execution. All precedence constraints for which both events are in this group are added to the process execution.

5 Only Connected Components Mitigate Quality Issues

In this section, we prove that only execution extraction based on connected components is free of convergence, deficiency, and divergence issues. To do so, we prove that connected-components extraction does not have any quality problems and, subsequently, prove that all techniques not employing connected components have quality issues.

5.1 Connected Components Mitigate Quality Issues

Theorem 1. *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an object-centric event log and let* $C_{ext\_cc}(L) = \{(E_1, K_1), \dots (E_n, K_n)\}$ *be extracted process executions with connected-components extraction. Then,* $C_{ext\_cc}(L)$ *does not suffer from convergence, divergence, or deficiency problems.*

We prove this theorem by proving that connected-components extraction is subject to neither convergence, deficiency, or divergence. For each of those, we show that extracted cases with the corresponding quality issue cannot stem from connected-components extraction.

Lemma 3 (Convergence Free). *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log and let* $C = C_{ext}(L) = \{(E_1, K_1), \dots (E_n, K_n)\}$ *be extracted cases using* $ext(L)$ *such that* $\exists_{e \in E} \exists_{(E_i,K_i),(E_j,K_j) \in C} \; (E_i, K_i) \neq (E_j, K_j) \wedge e \in E_i \wedge e \in E_j$ *(a convergence problem is present). Then* $C \neq C_{ext\_cc}(L)$*, i.e.,* C *cannot stem from a connected-components extraction.*

Proof. We prove our lemma by contradiction: Suppose we have an event that is part of two cases, i.e., a convergence problem $\exists_{e \in E} \exists_{(E_i,K_i),(E_j,K_j) \in C} \; (E_i, K_i) \neq (E_j, K_j) \wedge e \in E_i \wedge e \in E_j$ and both cases would stem from connected-components extraction $(E_i, K_i), (E_j, K_j) \in ext_{cc}(L)$. For both cases, their events would be constructed by adding all events reachable from event e, according to the connected-components extraction, i.e., $E_i = \{e' \in E \mid e \sim_{EO} e'\}$ and $E_j = \{e' \in E \mid e \sim_{EO} e'\}$, i.e., $E_i = E_j$. Thus, $(E_i, K_i) = (E_j, K_j)$ which conflicts with the definition of convergence. Therefore, an extraction with convergence issues cannot stem from connected-components extraction.

Lemma 4 (Deficiency Free). *Let $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ be an event log and let $C = C_{ext}(L) = \{(E_1, K_1), \dots (E_n, K_n)\}$ be extracted cases using $ext(L)$ such that $\exists_{e \in E}\ \neg\exists_{(E_i,K_i) \in C}\ e \in E_i$ (a deficiency problem is present). Then $C \neq C_{ext\_cc}(L)$, i.e., C cannot stem from a connected-components extraction.*

Proof. We prove this lemma by contradiction: We assume that $\{(E_1, K_1), \dots, (E_n, K_n)\} \subseteq C_{ext\_cc}(L)$ are cases retrieved from connected-components extraction and there is an event that is not part of a case $\exists_{e \in E}\ \neg\exists_{(E_i,K_i) \in C_{ext\_cc}(L)}\ e{\in}E_i$ Therefore, the event is not part of any case $\forall_{(E_i,K_i) \in C_{ext\_cc}(L)}\ e \notin E_i$. Due to the equivalence relation of connected-components extraction, it must hold that e has n opath to any other event $\forall_{(E_i,K_i) \in ext_{cc}(L)}\ \forall_{e' \in E_i} e \nsim_{EO} e'$. However, e has a path to itself $e \sim_{EO} e$, therefore, e must appear in one of the cases extracted by connected components and cases suffering from the deficiency problem cannot stem from connected-components extraction.

Corollary 1. *Let $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ be an event log and $\{(E_1, K_1), \dots (E_n, K_n)\} = C_{ext\_cc}(L)$ be cases form connected-components extraction. Since they do not have convergence or divergence issues $E{=}E_1 \cup \dots \cup E_n$.*

If no event is missing, the set of events included in the cases corresponds exactly to the events in the event log.

Lemma 5 (Divergence Free). *Let $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ be an event log and let $C = C_{ext}(L) = \{(E_1, K_1), \dots (E_n, K_n)\}$ be extracted cases using $ext(L)$ such that $\exists_{(e,e') \in E \times E}\ (e, e') \in K \not\Leftrightarrow \exists_{(E_i,K_i) \in C}\ (e, e') \in K_i$ (a divergence problem is present). Then $C \neq C_{ext\_cc}(L)$, i.e., C cannot stem from a connected components extraction.*

Proof. We prove that an extraction with a divergence problem cannot stem from connected-components extraction by proving both directions of the equivalence relation hold for connected components: First ($\Rightarrow$), a precedence constraint present in the event log must be in the cases and, second ($\Leftarrow$), a precedence constraint present in the cases must be in the event log. For the first, we assume that $\{(E_1, K_1), \dots (E_n, K_n)\} \in ext_{cc}(L)$ and show a contradiction.

1) $\Rightarrow$ Suppose there is a precedence constraint in the event log, i.e., the event-object graph, but not in the cases $\exists_{(e,e') \in E \times E}(e, e') \in K \not\Rightarrow \exists_{(E_i,K_i) \in C}\ (e, e') \in K_i$. This precedence constraint connects two nodes e and e'. Since all events are covered in connected-components extraction $E =\ E_1 \cup \ldots \cup E_n$ the event e needs to be in one case $\exists E_i \in \{E_1, \dots, E_n\}$ such that $e \in E_i$. Since e has a path to e', $e \sim_{EO} e' = \langle e, e' \rangle \neq \perp$, it holds that e' must also be in the same case $e' \in E_i$. However, since the precedence constraints of the case are defined through the precedence constraints of the events in the event log $K_i = E_i \times E_i \cap K$ and $e, e' \in E_i$ and $(e, e') \in K$ it holds that the precedence constraint must be contained in the event log $(e, e') \in K_i$. This is a contradiction. Therefore, an extraction where a precedence constraint is present in the event log but not in the cases cannot stem from connected-components extraction.

2) By definition, the precedence constraints of the extracted cases are a subset of the ones in the event log. Therefore, this direction holds.

5.2 Only Connected Components Mitigate Quality Problems

As we have shown that connected-components extraction does not suffer from convergence, deficiency, or divergence issues, we want to examine other extraction techniques in this section. We prove that all techniques not building process executions from connected-components have quality issues.

Theorem 2. *Let* $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ *be an event log and let* $C = C_{ext}(L) = \{(E_1, K_1), \dots (E_n, K_n)\}$ *be extracted cases with another extraction than connected-component extraction. Then,* C *is either composed of multiple connected components or suffers from a convergence, divergence, or deficiency problem.*

We prove this theorem by deconstructing the definition of connected-components extraction to list all ways in which other extraction techniques can differ from it. These two ways are events connected by a path ending up in different cases and events without a path ending up in the same case. For both these ways, we show that it always introduces quality problems or boils down to an extraction where multiple connected-components are grouped to one process execution.

Proof. Let $L = (E, O, OT, \pi_{obj}, \pi_{act}, \pi_{time})$ be an event log and let $C = C_{ext}(L) = \{(E_1, K_1), \dots (E_n, K_n)\}$ be extracted cases with another extraction technique than connected-component, i.e., $\exists_{(E_i,K_i)\in C}\ e, e' \in E_i \not\Leftrightarrow e \sim_{EO} e'$. We break this down into the possible conditions that would fulfill a non-connected-components extraction. First, it may include two events in a case that are not connected by a path $\exists_{(E_i,K_i)\in C}\ e, e' \in E_i \wedge e \nsim_{EO} e'$, or, second, it may exclude an event with a path to another event from the same case $\exists_{(E_i,K_i)\in C}\ \{e, e'\} \nsubseteq E_i \wedge e \sim_{EO} e'$. For both possible conditions, we will individually prove that there is a convergence, divergence, or deficiency problem.

1) $\exists_{(E_i,K_i)\in C}\ e, e' \in E_i \wedge e \nsim_{EO} e'$ then C has a convergence, divergence, or deficiency issue.
   When two events are contained in the same case without a path $\exists_{(E_i,K_i)\in C}\ e, e' \subseteq E_i \wedge e \nsim_{EO} e'$ we apply the transitivity lemma (cf. Lemma 2) to show that these two events are connected to a disjoint set of other events, $E_1 = \{e'' \in E' \mid e \sim_{EO} e'' \vee e'' \sim_{EO} e\}$ and $E_2 = \{e'' \in E' \mid e' \sim_{EO} e'' \vee e'' \sim_{EO} e'\}$ with $E_1 \cap E_2 = \emptyset$. For these two event sets, we define a tautology comprising a statement and its negation, implying that either the statement or its negation must be satisfied. $\forall_{E'' \in \{E_1, E_2\}}$

   a) $\neg\exists_{e \in E \setminus E''} \forall_{e' \in E''}\ e \sim_{EO} e' \vee e' \sim_{EO} e$ or
   b) $\exists_{e \in E \setminus E''} \forall_{e' \in E''}\ e \sim_{EO} e' \vee e' \sim_{EO} e$.

   The first part of the tautology states that there is no event outside this set that has a path to this set's elements. The second part states that an event outside a set element has a path to that set's elements. For both parts of the tautology, we show there is a convergence, deficiency, or divergence problem or it boils down to connected-components extraction if it is fulfilled:

   a) E'' constructs a connected component and does not induce any quality problem (cf. Theorem 1).
   b) Here we can directly apply the connecting-edge lemma (cf. Lemma 1). $\exists_{e \in E \setminus E''} \exists_{e' \in E''}\ e \sim_{EO} e' \Rightarrow \exists_{(e_1,e_2) \in (E \setminus E'' \times E'')}\ (e_1, e_2) \in K$. With $e_1 \sim_{EO} e_2 \wedge e_2 \in E''$ it holds that $e_1 \notin E'$, as e_1 would be part of E'' if it would be in E' since it is connected to nodes of E''. Therefore, $(e_1, e_2) \notin E' \times E'$ and following that $(e_1, e_2) \notin E' \times E' \cap K = K'$. Therefore, a divergence problem is present unless both events are part of another case, which would introduce a convergence problem.

2) $\exists_{(E_i,K_i) \in C}\ e \in E_i \wedge e' \notin E_i \wedge e \sim_{EO} e'$ then C has a convergence, divergence, or deficiency issue.

   When considering two events connected by a path but not included in the same case, we extend the formula by adding a tautology with a statement and its negation $\exists_{(E_i,K_i) \in C}\ e \notin E_i \wedge e' \in E_i \wedge e \sim_{EO} e' \wedge$

   a) $(\neg\exists_{E'' \in (E_1,\ldots,E_n)}\ e \in E'' \vee$
   b) $\exists_{E'' \in (E_1,\ldots,E_n)}\ e \in E'')$

   The first part states that the event connected by a path but not in the case is also not in the other case. The second part states that there is another case that contains the event. Either one of the two needs to be fulfilled, therefore, we will show that both lead to convergence, divergence, or deficiency.

   a) This corresponds to the definition of a deficiency problem, i.e., an event is in no case.
   b) It holds that $e \in E'' \wedge e \notin E'$. We can generalize this such that $e \in E \setminus E'$ and apply Lemma 1. $e \in E \setminus E' \wedge e' \in E' \wedge e \sim_{EO} e' \Rightarrow \exists_{(e_1,e_2) \in E \setminus E' \times E'}\ (e_1, e_2) \in K$. Since $e_1 \notin E'$ it also holds that $(e_1, e_2) \notin E' \times E'$ and $(e_1, e_2) \notin E' \times E' \cap K$. Therefore, the edge (precedence constraint) cannot be included in the case (E', K'). To avoid a divergence problem, it would need to be included in another case. However, then both events of the edge would need to be included in the case, leading to a convergence issue since e appears in both cases. Therefore, either a convergence or divergence issue is present.

6 Discussion, Limitations, and Implications

In this section, we discuss the relationship of our proof to traditional process mining, draw limitations of our proof for practice, and derive the implications for future process mining research.

Table 1. Different case extraction techniques and their properties. Divergence issues that are only introduced by deficiency issues are depicted with (✓).

Extraction Technique	Convergence-Free	Divergence-Free	Deficiency-Free
Single-Type Flattening [2]			
Composite-Type Flattening [4,7]	✓		✓
Leading-Type Extraction [5]		(✓)	
Maximal-Type-Set Extraction	✓	(✓)	
Connected-Components Extraction [5]	✓	✓	✓

6.1 Practical Limitations

Our paper has shown that case extraction can only conform to the conditions of convergence-freeness, divergence-freeness, and deficiency-freeness when using connected-components extraction. While these are important and foundational criteria for the correctness of event data, there are two major factors to take into consideration: First of all, for some downstream process mining tasks, the presence of quality problems might not affect the quality of the results. For example, when computing the cycle time of different objects, e.g., orders that are placed and then delivered, it would not matter if some events that are shared between orders are multiplied, as this does not affect the computation of cycle times. Second, one might have additional conditions when extracting cases, rendering the problem of fulfilling these conditions and mitigating quality problems infeasible. For example, requiring exactly one sales order object per case would conflict with connected-components extraction if one connected component contains two sales orders. *This means, that mitigating convergence, divergence, and deficiency is often infeasible in practice.* For such cases, a trade-off between different quality issues has to be made. We provide a collection of current alternative case extraction techniques along with associated quality issues in Table 1. These techniques could also be employed if their underlying quality problems would not affect the results of the employed analysis.

Techniques enforcing sequential, traditional cases are called flattening. All other listed techniques produce graph-based process executions. As enforcing sequentiality tempers with precedence constraints, flattening techniques can never guarantee divergence-freeness. Deficiency implies divergence, as missing events lead to missing precedence constraints. Therefore, we depict a technique that is only subject to divergence issues introduced by deficiency issues with a checkmark in parentheses.

As discussed by van der Aalst [2], single-type flattening is subject to any of the three quality problems. When choosing a single object type and considering the event sequence of each object as a case, events might get duplicated, missing, and precedence constraints get left out. However, single-type flattening can still offer valuable insights into the subprocess of a single object type, especially when considering that the whole traditional process mining pipeline can be applied.

Composite-type flattening describes an extraction technique that collects connected objects and merges their events into an event sequence [7]. This is equivalent to compressing the results of connected-component extraction by Adams et al. [4] into a sequence. As it builds on connected-components, it is free of convergence and deficiency issues. Therefore, it can produce valuable insights when precedence constraints are not important. However, divergence issues might be very consequential for some tasks like discovery.

Leading-type extraction extracts subgraphs of the connected components that are associated with objects of a leading type and their closest related objects of other types [5]. As the closest objects of one leading object can also be the closest objects of another object (i.e., two sales orders having the same delivery object), this extraction is subject to convergence. Some objects can be left out, e.g., if they are not connected to any leading object, introducing deficiency and, therefore, also divergence problems. These quality problems are further illustrated in [6]. However, the precedence constraints within the process execution are correctly represented. Furthermore, this extraction technique allows to incorporate related objects, in contrast to single-type flattening.

Maximal-type-set extraction is a new extraction technique suggestion based on the results of this paper. Consider a setting where a user has an additional condition that the extracted cases should satisfy. An extraction technique could eliminate a minimal set of object types such that the connected-components extraction satisfies the condition. Under this consideration, maximal-type-set extraction would only be subject to deficiency and deficiency-induced divergence and could be an alternative to single-type flattening since it can incorporate information about related objects.

6.2 Traditional Event Logs and Cases

Traditional event logs, where each event is associated with precisely one object and all objects are of the same type, are a special case of object-centric event logs. When constructing the event-object graph for such logs, each object creates its own weakly-connected component, as no events are shared between objects. Consequently, connected-components extraction aligns with the traditional case notion, resulting in one event sequence per object. This underscores two points: First, connected-components extraction is backward compatible with traditional process mining, i.e., if the input is a traditional event log, the process executions correspond to cases. Second, if the information system's underlying events adhere to the assumption of exactly one object per event, traditional process mining does not encounter convergence, deficiency, or divergence problems. This assumption holds true for some information systems, like ticketing or case management systems [12]. However, for the vast majority of information systems this does not hold true, i.e., quality problems are to be expected when extracting traditional cases, i.e., sequences.

6.3 Implications

The most important implication of our paper is that quality issues are unavoidable if there are conflicting requirements for connected components. Depending on the planned process analysis, this could have significant effects on the quality of the results. Based on this, the second implication of our work is that many real-life applications of object-centric process mining will have to make a trade-off between different quality issues using different extraction techniques. As we currently can only make qualitative statements (cf. Table 1), a quantitative evaluation of quality issues for different extraction techniques is necessary. Furthermore, an overview of which process analysis tasks are negatively affected by which quality issues is necessary. Based on the trade-off spectrum between different quality issues, new extraction techniques that inhibit unoccupied parts of the spectrum can be proposed. These developments will bring transparency and capabilities to dealing with quality issues in object-centric process mining.

We derive secondary implications from aligning connected-components extraction with the traditional case notion: New extraction techniques should provide the traditional case concept when applied to traditional event logs and adaptations of traditional process mining algorithms to the object-centric setting should be backward compatible, i.e., if the object-centric event log is a traditional event log, the results of the algorithm should be consistent with the traditional process mining algorithm. This ensures the consistency of the process mining field when moving towards object-centricity. Examples of such backward-compatible process mining algorithms are the discovery of object-centric Petri nets [3], which yields a standard Petri net when fed with a traditional event log, object-centric variants [5], which produce standard variants when fed with a traditional event log, or object-centric features [4] which would return standard features when fed with a traditional event log. This ensures the consistency of the process mining field.

7 Conclusion

This paper addressed the issue of quality problems in process mining when extracting cases. We proved that only case extraction based on connected components is free of convergence, deficiency, or divergence issues. For process analysis tasks that are negatively affected by quality issues, connected components can yield the only case extraction that mitigates the negative impacts of convergence, divergence, and deficiency. If there are other conditions that the extracted cases should fulfill and these conditions conflict with connected-components extraction, mitigating quality problems is infeasible. In that situation, case extraction is a trade-off between different quality issues. Furthermore, we have discussed that connected-components extraction aligns with the traditional case notion of traditional event logs. This necessitates object-centric adaptations of traditional process mining algorithms to be backward compatible, ensuring the consistency of the process mining field.

For future work, there should be an in-depth analysis of the impact of different quality issues on different process analysis tasks, as well as a quantification of quality issues for different case extraction techniques. These contributions would deliver transparency to anyone applying case extraction to conduct process analysis tasks by understanding the presence of quality issues, their effect on the quality of results, and the trade-offs that can be achieved by utilizing different case extraction techniques.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, 2nd edn. Springer, Cham (2016). https://doi.org/10.1007/978-3-662-49851-4
2. van der Aalst, W.M.P.: Object-centric process mining: dealing with divergence and convergence in event data. In: Ölveczky, P.C., Salaün, G. (eds.) SEFM 2019. LNCS, vol. 11724, pp. 3–25. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30446-1_1
3. van der Aalst, W.M.P., Berti, A.: Discovering object-centric Petri nets. Fundam. Informaticae **175**(1–4), 1–40 (2020). https://doi.org/10.3233/FI-2020-1946
4. Adams, J.N., Park, G., Levich, S., Schuster, D., van der Aalst, W.M.P.: A framework for extracting and encoding features from object-centric event data. In: Troya, J., Medjahed, B., Piattini, M., Yao, L., Fernandez, P., Ruiz-Cortes, A. (eds.) Service-Oriented Computing. Lecture Notes in Computer Science, vol. 13740, pp. 36–53. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20984-0_3
5. Adams, J.N., Schuster, D., Schmitz, S., Schuh, G., van der Aalst, W.M.P.: Defining cases and variants for object-centric event data. In: ICPM, pp. 128–135. IEEE (2022). https://doi.org/10.1109/ICPM57379.2022.9980730
6. Adams, J.N., van Zelst, S.J., Rose, T., van der Aalst, W.M.P.: Explainable concept drift in process mining. Inf. Syst. **114**, 102177 (2023). https://doi.org/10.1016/j.is.2023.102177
7. Calvanese, D., Montali, M., Syamsiyah, A., van der Aalst, W.M.P.: Ontology-driven extraction of event logs from relational databases. In: Reichert, M., Reijers, H.A. (eds.) BPM 2015. LNBIP, vol. 256, pp. 140–153. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42887-1_12
8. Fahland, D.: Process mining over multiple behavioral dimensions with event knowledge graphs. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. Lecture Notes in Business Information Processing, vol. 448, pp. 274–319. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_9
9. Gerke, K., Mendling, J., Tarmyshov, K.: Case construction for mining supply chain processes. In: Abramowicz, W. (ed.) BIS 2009. LNBIP, vol. 21, pp. 181–192. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01190-0_16
10. Leemans, S.J.J., van Zelst, S.J., Lu, X.: Partial-order-based process mining: a survey and outlook. Knowl. Inf. Syst. **65**(1), 1–29 (2023). https://doi.org/10.1007/s10115-022-01777-3
11. Lu, X., Nagelkerke, M., van de Wiel, D., Fahland, D.: Discovering interacting artifacts from ERP systems. IEEE Trans. Serv. Comput. **8**(6), 861–873 (2015). https://doi.org/10.1109/TSC.2015.2474358
12. Weerdt, J.D., Wynn, M.T.: Foundations of process event data. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. Lecture Notes in Business Information Processing, vol. 448, pp. 193–211. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_6

A Model-Driven Engineering Perspective for the Object-Centric Event Data (OCED) Metamodel

Daniel Calegari(✉) and Andrea Delgado

Instituto de Computación, Facultad de Ingeniería, Universidad de la República,
11300 Montevideo, Uruguay
`{dcalegar,adelgado}@fing.edu.uy`

Abstract. Object-centric process mining has been proposed as a practical solution for dealing with the multiple views from which a process can be analyzed regarding its relation with organizational data (i.e., objects). The Object-Centric Event Data (OCED) metamodel was recently proposed as a data exchange standard for object-centric process mining. As far as we know, the metamodel has yet to be studied from a Model-Driven Engineering (MDE) perspective. This paper provides an Ecore-based representation of the OCED metamodel and explores its capabilities from an MDE perspective. We also study how the Business Process and Organizational Data Integrated Metamodel (BPODIM), i.e., a proposal for integrating process and organizational data, can be aligned with OCED providing fruitful information for OCED improvement.

Keywords: Object-centric · OCED · model-driven engineering

1 Introduction

Process Mining (PM) [1] is a powerful approach employed by organizations to gain insights into the actual execution of their business processes (BPs), thereby uncovering valuable information for identifying opportunities for improvement. It relies on event logs from information systems supporting process execution. Traditional techniques assume events are related to a particular case, i.e., a process instance. Nevertheless, real environments induce different case notions leading to different views of the same process, e.g., from the perspective of different data object types. In this context, traditional techniques must address several challenges [2], such as representing multiple cases within a log and its analysis from a multi-view perspective.

Object-centric process mining has been proposed to address these challenges, relaxing the assumption that an event is solely associated with a single case and allowing for events to be related to different objects of various types. It captures the relationships between objects (organizational data) and process data. Consequently, new techniques and tools have been developed for analyzing

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 508–520, 2024.
https://doi.org/10.1007/978-3-031-50974-2_38

object-centric event logs and process models [3,4], along with the proposal of standards for storing object-centric event logs [12].

The IEEE Task Force on Process Mining has established the Object-Centric Event Data (OCED)[1] standard as an eXtensible Event Stream (XES) standard [14] successor [21]. It is an abstract metamodel from which reference implementations must be proposed. As far as we know, the metamodel has yet to be studied from a Model-Driven Engineering (MDE) [15] perspective.

MDE emphasizes the specification of models conforming to metamodels and the use of transformations between these models for various objectives, e.g., model refinement and code generation. Since it provides a mature set of technical concepts and implemented technologies, adopting an MDE perspective for the OCED proposal could provide several benefits. For example, it may help develop more expressive metamodels and concrete representation languages/formats for process/event logs. By analyzing the transformability between modeling languages (or metamodels), MDE can also help to analyze the expressiveness of a modeling language and connect these languages to be used in a broader context.

As an example of the latter, in previous work [9], we addressed the challenge of integrating processes and organizational data by adopting an MDE approach. We introduced the Business Process and Organizational Data Integrated Metamodel (BPODIM), which enables the collection of heterogeneous data through generic APIs [10], and its integration to provide a unified view for PM. We envisioned model transformations from BPODIM models to multiple targets, such as an XES event log. Considering the current existence of the OCED metamodel, it seems reasonable to align BPODIM with the object-centric view, providing a complete approach from data extraction to object-centric log creation.

This paper explores an MDE perspective for the OCED metamodel based on representing the OCED metamodel using the Eclipse Modeling Framework (EMF)[2] and analyzing how it could be exploited. We also study how the BPODIM proposal can be aligned with the OCED metamodel by defining a transformation from BPODIM to OCED and discussing alignment challenges.

The rest of the paper is organized as follows. Section 2 presents the OCED metamodel and discusses the opportunities that arise when focusing on an MDE perspective. Then, Sect. 3 briefly summarizes the BPODIM proposal from [9,10], and in Sect. 4, we present how it can be aligned with OCED and discuss OCED expressiveness in a broader context. In Sect. 5, we summarize related work, and in Sect. 6, we provide some conclusions.

2 The OCED Metamodel from a MDE Perspective

We have defined a reference implementation [7] of the complete Object-Centric Event Data (OCED) metamodel using an MDE perspective. As depicted in Fig. 1, the Ecore-based definition of OCED directly represents the original object-centric concepts. The left-side part of the metamodel describes events,

---

[1] OCED Standard. https://www.tf-pm.org/resources/oced-standard.

[2] Eclipse Modeling Framework (EMF). https://www.eclipse.org/modeling/emf/.

associated attributes, and the time construct, while the right-side part describes the objects, associated attributes, and object relations. Events are connected to objects, attributes, and relations through qualified associations that are represented as pivot elements (e.g., `event-object`) to store its qualifier. The metamodel was partitioned in two to express the two levels that the OCED call for action proposes: a base metamodel (the one depicted in Fig. 1) and a full metamodel (in [7]) expressing the qualified associations between the event, object attributes, and relations. The full metamodel is defined as an extension of the base metamodel using the EMF standard extension mechanism.

The metamodel could be improved, e.g., the event type could be modeled as an attribute of the event class since it is a single attribute concept with a [1..1] to [1..*] relation with the event. However, we decided to provide the most direct representation of the original metamodel, which allows us to adapt to proposed changes and extensions, e.g., a hierarchy of event types.

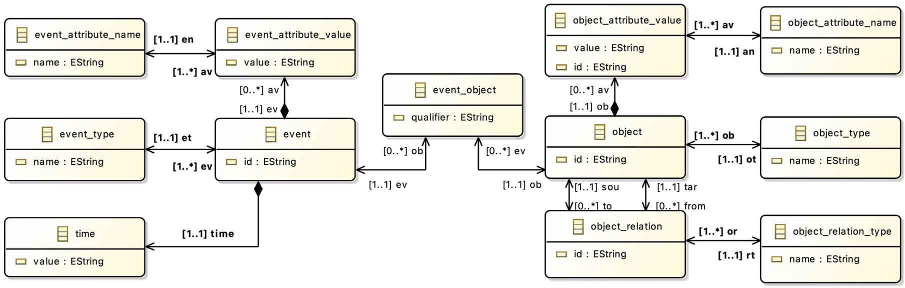


Fig. 1. OCED base metamodel

Since MDE provides a mature set of technical concepts and implemented technologies, adopting an MDE perspective for the OCED proposal opens new opportunities for its use in a broader context. We now identify some of these opportunities and provide examples supporting this view. These examples can be accessed in [7].

Figure 2 provides a general view of the examples we provide: (a) OCED metamodel and models definition; (b) generation of an in-memory OCED model from Java code; (c) ATL[3] model-to-model (M2M) transformation from BPODIM to OCED; (d) Acceleo[4] model-to-text (M2T) transformation for serializing OCED models into two formats: XES extended event log for organizational data [5], and OCEL JSON [12]; (e) Acceleo M2T transformation for generating a type-dependent database (SQL script) from an OCED model. For space reasons and aligned with the objectives of this work, we only describe in detail the M2M transformation from BPODIM to OCED in Sect. 4.

---

[3] ATL Transformation Language. https://www.eclipse.org/atl/.

[4] Acceleo. https://www.eclipse.org/acceleo/.

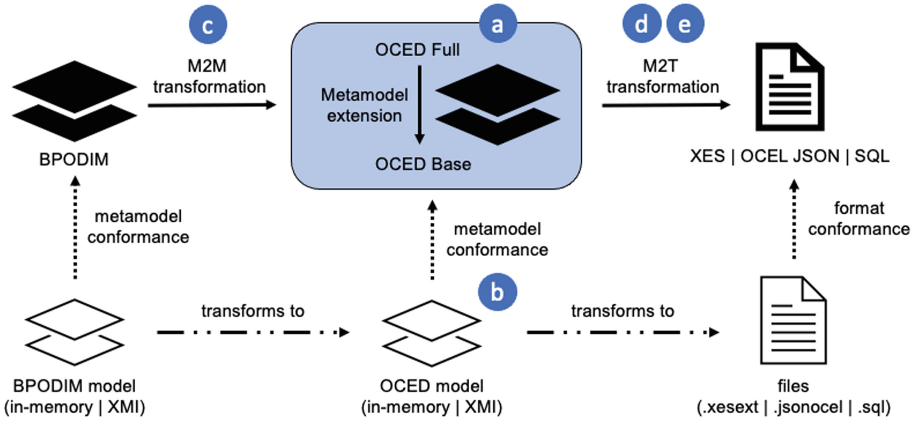


Fig. 2. Examples of using the OCED metamodel

From a **technical perspective**, many tools allow for managing metamodels and performing related model-based activities. For example:

1. The EMF toolset provides a standard *XMI serialization* for models and metamodels, and a tool for checking *conformance* between a model and a metamodel.
2. EMF provides a *reflective API* for manipulating in-memory models. In large models, it could be possible to incorporate a more *scalable modeling ecosystem* based on NoSQL databases, e.g., [8].
3. EMF provides a *Java-based representation* of the metamodel to process an in-memory model integrated with an industrial programming language.

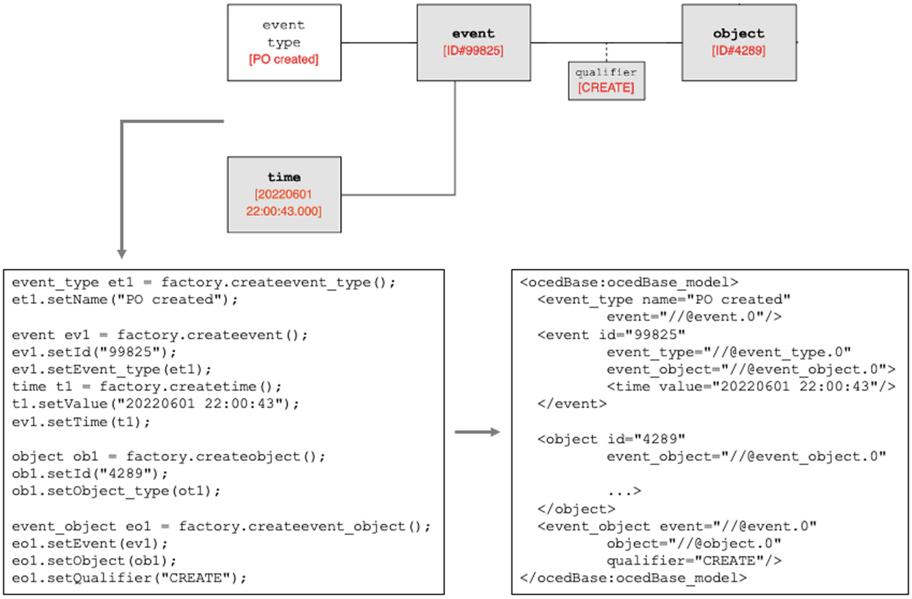
In [7], we exemplify an OCED model definition from code and its storage using the XMI format. An excerpt is depicted in Fig. 3, in which an event `#99825` and an object `#4289` are related through a qualified association `CREATE`.

From a **metamodeling perspective**, there are some opportunities, such as:

1. Leveraging the *extension mechanism*, as we did for expressing the OCED metamodel at two different levels. It allows integrating with other initiatives and extending its capabilities. For example, extending the metamodel for handling static and dynamic attributes for events and objects is possible in a similar way the DOCEL proposal does [13].
2. Defining a metamodel of filters capable of specifying the sequence of filters that must be applied to an OCED model as part of a PM project.
3. Using tools for *comparing and merging models* (e.g., EMF Compare), as well as for *analysis* of the complexity of a model and improving its understanding (e.g., MetaModelAgent[5])
4. Defining a *domain-specific language* for log definition and use the Object Constraint Language (OCL) for expressing *invariants* on models.

---

[5] MetaModelAgent. www.metamodelagent.com.



```
event_type et1 = factory.createevent_type();
et1.setName("PO created");

event ev1 = factory.createevent();
ev1.setId("99825");
ev1.setEvent_type(et1);
time t1 = factory.createtime();
t1.setValue("20220601 22:00:43");
ev1.setTime(t1);

object ob1 = factory.createobject();
ob1.setId("4289");
ob1.setObject_type(ot1);

event_object eo1 = factory.createevent_object();
eo1.setEvent(ev1);
eo1.setObject(ob1);
eo1.setQualifier("CREATE");
```

```
<ocedBase:ocedBase_model>
  <event_type name="PO created"
          event="//@event.0"/>
  <event id="99825"
          event_type="//@event_type.0"
          event_object="//@event_object.0">
          <time value="20220601 22:00:43"/>
  </event>

  <object id="4289"
          event_object="//@event_object.0"

          ...>
  </object>
  <event_object event="//@event.0"
          object="//@object.0"
          qualifier="CREATE"/>
</ocedBase:ocedBase_model>
```

Fig. 3. OCED model excerpt (code + XMI)

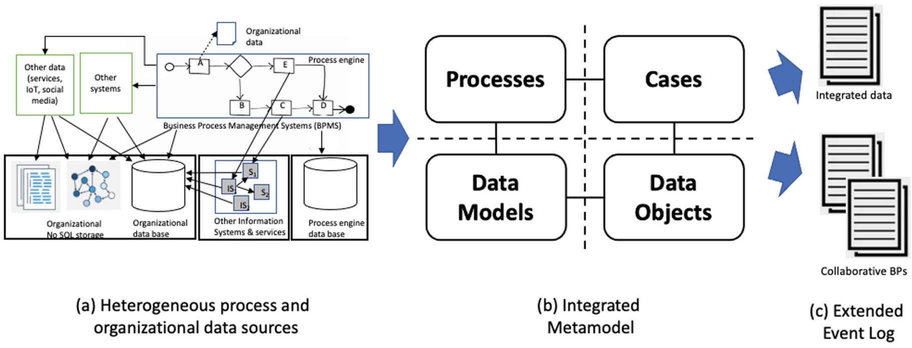
We can also identify many opportunities from a **model transformations perspective**. Following the classification of model transformations in [16]:

1. *Re-expression*, i.e., mapping a model into its equivalent in another language at the same level of abstraction. For example, transforming BPODIM into OCED models (described in Sect. 4) and serializing OCED models in XES extended event log [5] and OCEL JSON [12] formats. As described in Sect. 4, re-expression presents challenges since, although being at the same level of abstraction, representation languages could present semantic differences that prevent full synchronization. These examples can be found in [7].
2. *Refinement*, i.e., mapping a more abstract model into a more specific version. For example, building a type-dependent repository from OCED-specific attributes. We provide an example of this that can be found in [7]. The transformation essentially takes event/object types to generate domain-specific relational tables (e.g., table `PO_created` storing events with such type). It allows querying the model using domain knowledge.
3. *Abstraction*, i.e., providing an abstraction of a model (the opposite of refinement). For example, querying the model and calculating quality metrics of integrated process and organizational data, such as those from the Business Process and Organizational Data Quality Model (BPODQM) [5]. The type-dependent repository mentioned before can be used as a built-in solution for the information systems from which the events/objects come. In such case, the repository is an abstraction from which an OCED model could be extracted, e.g., from Java code as the example already mentioned [7].

4. *Quality improvement/restructuring*, i.e., reorganizing a model to achieve some quality goal, remaining at the same abstraction level. For example, filtering/repairing an OCEL event log.

3 A Unified Vision of Processes and Organizational Data

In previous work [9,10], we have proposed a general approach for dealing with heterogeneous process data and organizational (application) data settings, as depicted in Fig. 4. The proposal is part of a process mining methodology [11] guiding the work within such a complex data scenario.



(a) Heterogeneous process and organizational data sources

(b) Integrated Metamodel

(c) Extended Event Log

Fig. 4. General approach for heterogeneous process and organizational data

The first step (a) depicts the complex scenario of process enactment with heterogeneous data sources. Events come from many sources: traditional information systems, BPMS platforms and their process engine databases, and relational and NoSQL databases, among others. An Extraction, Transformation, and Load (ETL) process [10] consistently gathers data from all sources and loads it into a model conforming to the Business Process and Organizational Data Integrated Metamodel[6] shown in step (b) [10]. Finally, step (c) shows that once the data is integrated, it is possible to exploit this model-driven nature by, for example, defining a model transformation generating an extended XES event log [10]. With the extended event logs, we can apply process mining to BPs execution, providing an integrated view for a complete, evidence-based BP analysis. Following a similar idea, we can define a model transformation to connect BPODIM with the OCED standard and exploit OCED's model-driven nature in different ways, as will be presented in the following sections.

The metamodel, depicted in Fig. 5, defines four quadrants. The left top quadrant presents concepts for process definition (orchestrations and choreographies), their corresponding elements (e.g., tasks), user roles, and variables defined in

---

[6] BPODIM. https://gitlab.fing.edu.uy/open-coal/bpodim.

design time. The right top quadrant shows related instances. The left bottom quadrant presents concepts for organizational data definition (the data model) with entities and their corresponding attributes, and the right bottom quadrant gives related instances (data objects) containing specific values that evolve through time. Details of the complete approach can be seen in [9,10].

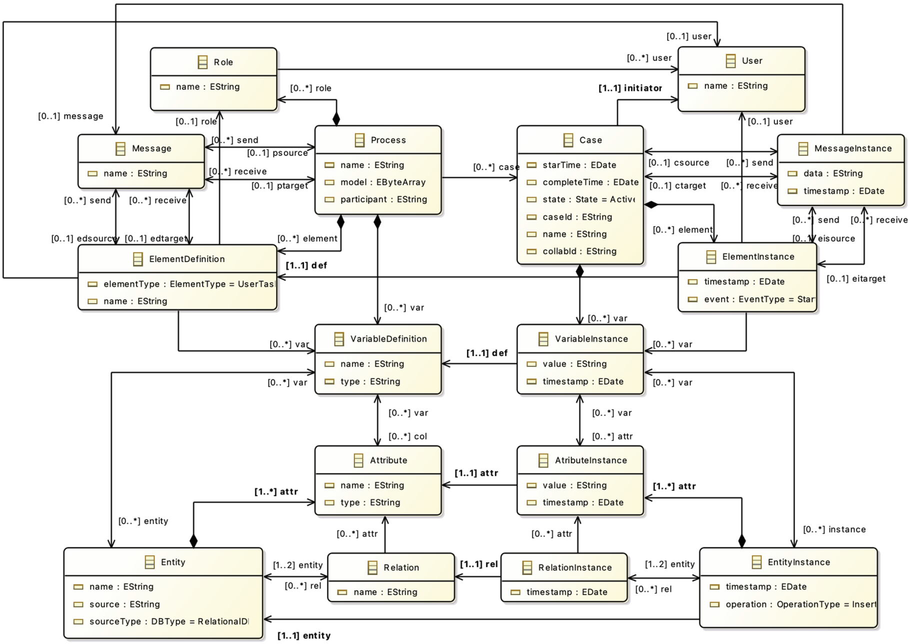


Fig. 5. BPODIM metamodel

4 Aligning BPODIM with OCED

We define an ATL model transformation from BPODIM to OCED [7], based on identifying mappings (Table 1) between elements. This alignment provides a complete approach from data extraction to object-centric log creation.

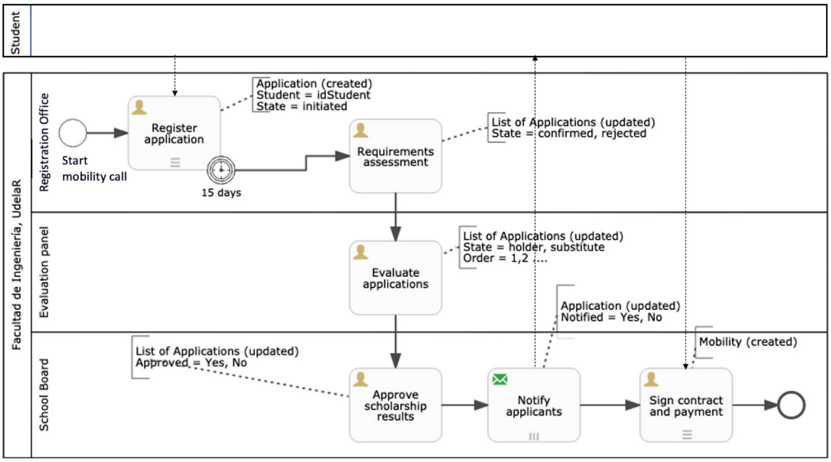
From the process perspective of BPODIM (process and case quadrants), we map `ElementInstance` to an OCED `Event` since both concepts refer to events occurring in the BP. From the data perspective of BPODIM (data model and data object quadrants), we map an `EntityInstance` (i.e., a data object) to an OCED `Object`. We also map every relation between entities (`RelationInstance`) to an OCED `Object_Relation`. Finally, we map every pair of connected element and entity instances with an OCED `Event_Object` connecting the corresponding mapped elements, using the entity instance's operation type (e.g., insert, delete) as the qualifier. We also defined correspondences between concepts from the process perspective of BPODIM, that can be optionally used to maintain references to process information. They are valuable for recomposing a flat event log.

Table 1. Mapping between BPODIM and OCED metamodels

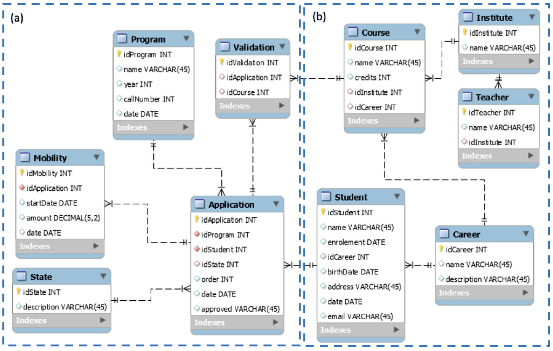
BPODIM	**OCED**
Mapping elements to events	
`ElementInstance (ei)`	`Event (ev)`
`ei.ElementDefinition.name`	`ev.Event_Type`
`ei.timestamp`	`ev.Time`
`ei.VariableInstance (vi)`	`ev.Event_Attribute_Value (eav)`
`vi.VariableDefinition.name`	`eav.Event_Attribute_Name`
Mapping entities to objects	
`EntityInstance (eni)`	`Object (obj)`
`eni.Entity.name`	`obj.Object_Type`
`eni.AttributeInstance (ai)`	`obj.Object_Attribute_Value (oav)`
`ai.Attribute.name`	`oav.Object_Attribute_Type`
`eni.RelationInstance (ri)`	`obj.Object_Relation (or)`
`ri.Relation.name`	`or.Object_Relation_Type`
Connecting events with objects	
If an `ElementInstance (ei)` is connected with an `EntityInstance (eni)` through their variable/attribute instances.	Their corresponding event/object are connected, i.e., `Event_Object(ev,obj)` with `ei->ev`, `eni->obj`, and `qualifier` is `ei.operation`.
Optional event attributes (usually present in flat event logs)	
`ei.event`	`ev.Event_Attribute_Value`
`ei.Case.caseId`	`ev.Event_Attribute_Value`
`ei.Case.Process.name`	`ev.Event_Attribute_Value`
`ei.ElementDefinition.elementType`	`ev.Event_Attribute_Value`
`ei.User.name`	`ev.Event_Attribute_Value`

4.1 BPODIM to OCED Example

To illustrate the M2M transformation, we use the "Student Mobility" BP, based on a real BP from our university [10], in which students apply for mobility scholarships to attend courses offered by partner universities. Figure 6 presents its definition and corresponding organizational data model composed of a process-specific model (a) referencing existing global entities to the organization (b).



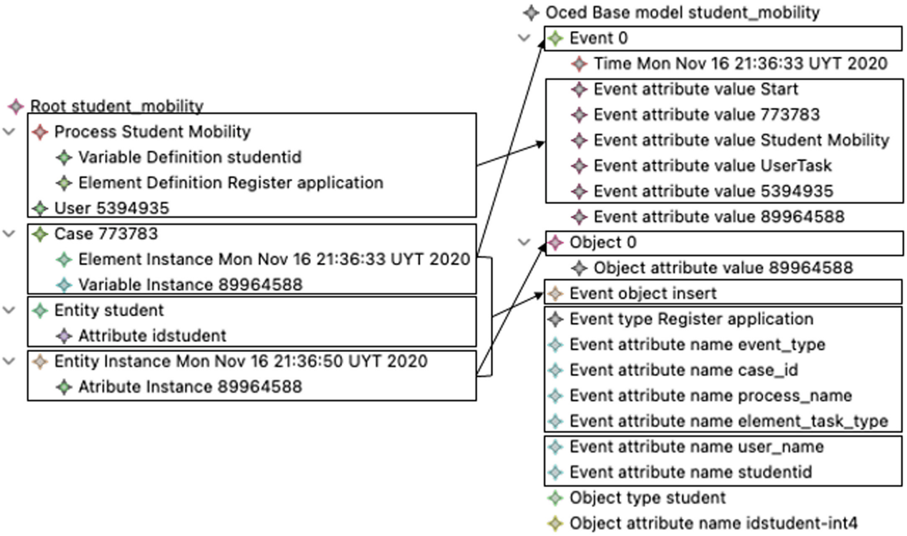
(a) Students Mobility BP



(b) Extended data model for the Students Mobility BP

Fig. 6. Students Mobility BP and data model

Figure 7 presents an excerpt of a BPODIM model and the generated OCED model. There is a case with an element instance corresponding to the `Register application` task with a variable `studentid` registering the student that fills the application. There is also a student entity with an attribute of type `idstudent` and an instance of such entity with an attribute storing the student ID. The element instance is mapped to the event with the corresponding timestamp and an attribute value for the student id `89964588`. General attribute values also correspond to the process information, e.g., the user that performs the event. Moreover, the entity instance is mapped to an object of the corresponding type, with an attribute `idstudent-int4` mapped from the attribute instance with type `idstudent`. The element and entity instances are connected through their variable and attribute instances (the student ID). Thus, there is a relation between the corresponding `event 0` and `object 0` with the `insert` qualifier that is the type of operation related to the entity instance.



- Root student_mobility
  - Process Student Mobility
    - Variable Definition studentid
    - Element Definition Register application
  - User 5394935
  - Case 773783
    - Element Instance Mon Nov 16 21:36:33 UYT 2020
    - Variable Instance 89964588
  - Entity student
    - Attribute idstudent
  - Entity Instance Mon Nov 16 21:36:50 UYT 2020
    - Atribute Instance 89964588

- Oced Base model student_mobility
  - Event 0
    - Time Mon Nov 16 21:36:33 UYT 2020
    - Event attribute value Start
    - Event attribute value 773783
    - Event attribute value Student Mobility
    - Event attribute value UserTask
    - Event attribute value 5394935
    - Event attribute value 89964588
  - Object 0
    - Object attribute value 89964588
  - Event object insert
  - Event type Register application
  - Event attribute name event_type
  - Event attribute name case_id
  - Event attribute name process_name
  - Event attribute name element_task_type
  - Event attribute name user_name
  - Event attribute name studentid
  - Object type student
  - Object attribute name idstudent-int4

Fig. 7. BPODIM to OCED generation for events and objects definition

4.2 Discussion

The biggest challenge is information loss. Although both metamodels could be considered at the same abstraction level, OCED is more focused and thus simplifies how the data is described. On the contrary, BPODIM offers more capabilities to integrate process and organizational data. However, its full capabilities are only used in some cases, e.g., sometimes the process description data is unavailable. Although it is possible to define a backward transformation (from OCED to BPODIM) if a full BPODIM model is transformed to OCED, the backward transformation will not be able to rebuild the former model.

The BPODIM metamodel has two quadrants for defining data connected with their corresponding instances. It allows representing the direct association between, e.g., an attribute instance and its definition, which defines a general name (e.g., the column name of a relational entity) and a type (e.g., string, int). Within the OCED metamodel, an (`Object_Attribute_Name`) is not directly related to the object type (`Object_Type`) corresponding to the entity definition (although it can be derived). Moreover, instances of variables and attributes are defined with their corresponding types (e.g., string, int), which do not have a direct representation in OCED. Some of these aspects can be represented using a workaround, such as the one we used for the optional attributes in Table 1.

BPODIM also allows representing other scenarios. For example, cases could have variables unrelated to element instances, and it is possible to have multiple instances of the same element representing the lifecycle of an event (e.g., start, complete). As in BPODIM, it is not possible to directly relate events to each other. These aspects could be addressed with appropriate metamodel extensions.

5 Related Work

The unified vision of processes and organizational data was exploited from many perspectives. Some works address the problem from a data perspective, e.g., in [6], the authors analyze the exploitation of database events as a source of information for event logs. Moreover, some works address the integration of both sources of information in a consistent and unified format, e.g., [18] and [20]. These works have some aspects in common with our proposal [9,10].

Some authors worked on improving the connection between existing technologies and the new object-centric vision. In [17], the authors propose an approach to extract object-centric logs from database tables using the eXtensible Object-Centric (XOC) log format. In [13], the authors express that existing event log formats (at a time) do not support dynamic object attributes (attributes with changing values) and introduce the Data-aware OCEL (DOCEL) format to support it, and an algorithm to translate XES logs into them. In [19], the authors propose an approach to automatically identify object-related information in flat event data and transform this data into OCEL. These works use formats that precede the OCED proposal and are compatible with an MDE perspective by defining corresponding metamodels and model transformations.

Metamodels were also used before in the context of process mining, e.g., for defining the XES standard [14] and the OCEL proposal [12]. However, its use was reduced to an abstract way of expressing the structure of something that will later be expressed using a concrete file format (e.g., JSON and XML). As far as we know, no works exploit the metamodeling nature from the MDE perspective.

6 Conclusions

This paper explored an MDE perspective for the OCED metamodel. We analyzed its MDE capabilities regarding technical, metamodeling, and model transformations aspects, providing examples of potential uses. It opens new opportunities for its use in a broader context related to process mining and software development. In particular, we studied the alignment between the BPODIM proposal for integrating process and organizational data with the OCED proposal. We defined a model transformation between them, allowing us to analyze alignment challenges, particularly information loss, and envision a complete approach from data extraction to object-centric log creation. These challenges provide valuable knowledge for improving the OCED proposal, which could be addressed with appropriate metamodeling extensions or workarounds.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, 2nd edn. Springer, Cham (2016)

2. Aalst, W.M.P.: Object-centric process mining: dealing with divergence and convergence in event data. In: Ölveczky, P.C., Salaün, G. (eds.) SEFM 2019. LNCS, vol. 11724, pp. 3–25. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30446-1_1
3. Adams, J.N., van der Aalst, W.M.P.: Ocπ: object-centric process insights. In: Bernardinello, L., Petrucci, L. (eds.) Application and Theory of Petri Nets and Concurrency. Lecture Notes in Computer Science, vol. 13288, pp. 139–150. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06653-5_8
4. Berti, A., van der Aalst, W.M.P.: OC-PM: analyzing object-centric event logs and process models. Int. J. Softw. Tools Technol. Transf. **25**(1), 1–17 (2023)
5. Betancor, F., Pérez, F., Marotta, A., Delgado, A.: Business process and organizational data quality model (BPODQM) for integrated process and data mining. In: Paiva, A.C.R., Cavalli, A.R., Ventura Martins, P., Pérez-Castillo, R. (eds.) QUATIC 2021. CCIS, vol. 1439, pp. 431–445. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85347-1_31
6. Claes, J., Poels, G.: Merging event logs for process mining: a rule based merging method and rule suggestion algorithm. Expert Syst. Appl. **41**(16), 7291–7306 (2014)
7. COAL: OCED: A meta model. https://gitlab.fing.edu.uy/open-coal/oced/
8. Daniel, G., Sunyé, G., Cabot, J.: Scalable queries and model transformations with the Mogwaï Tool. In: Rensink, A., Sánchez Cuadrado, J. (eds.) ICMT 2018. LNCS, vol. 10888, pp. 175–183. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93317-7_9
9. Delgado, A., Calegari, D.: Towards a unified vision of business process and organizational data. In: XLVI Latin American Computing Conference (CLEI), pp. 108–117. IEEE (2020)
10. Delgado, A., Calegari, D.: Process and organizational data integration from BPMS and Relational/NOSQL sources for process mining. In: Proceedings of the 17th International Conference on Software Technologies, ICSOFT, pp. 557–566. SCITEPRESS (2022)
11. Delgado, A., Calegari, D., Marotta, A., González, L., Tansini, L.: A methodology for organizational data science towards evidence-based process improvement. In: Fill, H.G., van Sinderen, M., Maciaszek, L.A. (eds.) Software Technologies. Communications in Computer and Information Science, vol. 1622, pp. 41–66. Springer, Cham (2021). https://doi.org/10.1007/978-3-031-11513-4_3
12. Ghahfarokhi, A.F., Park, G., Berti, A., van der Aalst, W.M.P.: OCEL: a standard for object-centric event logs. In: Bellatreche, L., et al. (eds.) ADBIS 2021. CCIS, vol. 1450, pp. 169–175. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85082-1_16
13. Goossens, A., Smedt, J.D., Vanthienen, J., van der Aalst, W.M.P.: Enhancing data-awareness of object-centric event logs. In: Montali, M., Senderovich, A., Weidlich, M. (eds.) Process Mining Workshops. Lecture Notes in Business Information Processing, vol. 468, pp. 18–30. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-27815-0_2
14. IEEE: IEEE standard for extensible event stream (XES) for achieving interoperability in event logs and event streams. IEEE Std. 1849–2016, pp. 1–50 (2016)
15. Kent, S.: Model driven engineering. In: Butler, M., Petre, L., Sere, K. (eds.) IFM 2002. LNCS, vol. 2335, pp. 286–298. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47884-1_16
16. Lano, K., Rahimi, S.K.: Model transformation specification and design. Adv. Comput. **85**, 123–163 (2012)

17. Li, G., de Murillas, E.G.L., de Carvalho, R.M., van der Aalst, W.M.P.: Extracting object-centric event logs to support process mining on databases. In: Mendling, J., Mouratidis, H. (eds.) CAiSE 2018. LNBIP, vol. 317, pp. 182–199. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92901-9_16
18. de Murillas, E.G.L., Reijers, H.A., van der Aalst, W.M.P.: Connecting databases with process mining: a meta model and toolset. SoSym **18**(2), 1209–1247 (2019)
19. Rebmann, A., Rehse, J., van der Aa, H.: Uncovering object-centric data in classical event logs for the automated transformation from XES to OCEL. In: Di Ciccio, C., Dijkman, R., del Rio Ortega, A., Rinderle-Ma, S. (eds.) Business Process Management. Lecture Notes in Computer Science, vol. 13420, pp. 379–396. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16103-2_25
20. Tsoury, A., Soffer, P., Reinhartz-Berger, I.: A conceptual framework for supporting deep exploration of business process behavior. In: Trujillo, J.C., et al. (eds.) ER 2018. LNCS, vol. 11157, pp. 58–71. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_6
21. Wynn, M.T., et al.: Rethinking the input for process mining: insights from the XES survey and workshop. In: Munoz-Gama, J., Lu, X. (eds.) ICPM 2021. LNBIP, vol. 433, pp. 3–16. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98581-3_1

Predictive Analytics for Object-Centric Processes: Do Graph Neural Networks Really Help?

Riccardo Galanti[1,2](✉) and Massimiliano de Leoni[2]

[1] IBM, Reggio Emilia, Italy
riccardo.galanti@ibm.com
[2] University of Padua, Padua, Italy
deleoni@math.unipd.it

Abstract. The object-centric process paradigm is increasingly gaining popularity in academia and industry. According to this paradigm, the process delineates through the parallel execution of different execution flows, each referring to a different object involved in the process. Object interaction is present, and takes place through bridging events where these parallel executions synchronize and exchange data. However, the complex intricacy of instances of such processes relating to each other via many-to-many associations makes a direct application of predictive process analytics approaches designed for single-id event logs impossible. This paper reports on the experience of comparing the predictions of two techniques based on gradient boosting or the Long Short-Term Memory (LSTM) network against two based on graph neural networks. The four techniques were empirically evaluated on event logs related to two real object-centric processes, and more than 20 different KPI definitions. The results show that graph-based neural networks generally perform worse than techniques based on Gradient Boosting. Considering that graph-based neural networks have training times that are 8-10 times larger, the conclusion is that their use does not seem to be justified.

Keywords: Predictive Analytics · Object-centric Processes · Machine Learning · Graph Neural Networks · Gradient Boosting

1 Introduction

Predictive process analytics is the branch of process mining that aims to predict the eventual outcome of process executions. Traditional predictive process analytics techniques rely on the assumption that process instances are composed of single flows of execution. However, recent industrial experience is showing that the assumption of a single execution flow is unfortunately often not met in practice. This led to the introduction of the paradigm of object-centric processes, which has recently been gaining more and more attention because it can naturally model inter-organizational processes more naturally [1]. Any process execution materializes itself as a set of instances that run concurrently, each representing the life cycle of one different object that contributes to the process execution (e.g., the order and the delivery object). These object life cycles

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 521–533, 2024.
https://doi.org/10.1007/978-3-031-50974-2_39

run independently and synchronize through bridging events to exchange data required for further processing.

The problem of predictive analytics remains relevant in the context of object-centric processes, as well. Typically, the process outcome is measured using a Key Performance Indicator (KPI), which depends on and is accordingly configured for the specific process being analyzed. The existing techniques for predictive process analytics cannot be directly applied in the context of object-centric processes because they heavily rely on the concept of a single identifier associated with a single execution flow.

This paper compares the predictive quality of two techniques leveraging on graph-based neural networks with one technique based on LSTM network and with one based on gradient boosting on decision trees. Graph-based neural network can naturally represent the complex many-to-many interaction between objects of object-centric processes. Conversely, LSTM networks and gradient boosting require the manual engineering of features that maintain a meaningful abstraction of the object interactions.

Graph-based neural networks have the significant disadvantage of a very large training time. The research question addressed in this paper is the following: *is the very large training time justified by a significant improvement of the prediction accuracy?* To answer this question, we conducted experiments with two object-centric processes and 21 different KPIs of interest, using the four predictive-analytics techniques mentioned above.

The results show that Gradient Boosting usually has the highest accuracy, compared with both LSTM and the two types of graph-based neural networks used in the experiments. At the same time, the training time is 8-10 times shorter. In sum, it seems that there is no advantage to use graph-based neural networks: a meaningful, manual engineering of features that encode the object interaction allows gradient boosting to reach higher prediction accuracy.

2 Preliminaries

2.1 Object-Centric Event Logs

Object-centric processes are executed with the support of one or more information systems. It is possible to extract the history of past executions from information systems into a transactional data set organized as object-centric event logs.

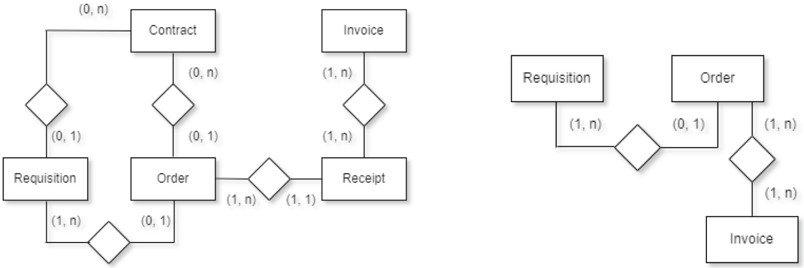
Space limitations prevent us from giving a full formalization, which can be found in [4]. Here, we limit ourselves to give the intuition through the example in Table 1, which shows an excerpt of an object-centric event log of an Italian utility-provider company. It consists of a set E of event identifiers (see column *ID*), each associated with an activity name, a timestamp of occurrence, a set of object identifiers of different types associated with the event (columns from *Contract* to *Invoice*), and a set of event attributes with their associated values (columns from *Order_Price* to *Rec_Quantity*).

In particular, five object types can be observed, each with a different life cycle: *Contract*, *Requisition*, *Order*, *Receipt*, and *Invoice*. As an example, the event with identifier $e1$ is associated with object $c1$ of type *Contract*, whereas, e.g., event $e20$ is associated with object $i2$ of type *Invoice*, and the three objects $r2$, $r3$, and $r4$ of type *Receipt*.

The object-centric event log can map the relationships between object types. For instance, the contracts $c1$ and $c2$ are associated with the requisitions $rq1$ and $rq2$,

Table 1. Example of an object-centric event log. Each row is an event, and the blank spaces represent attributes' missing values.

ID	Activity	Timestamp	Contract	Requisition	Order	Receipt	Invoice	User	Order Price	Order Month	Order Group	Rec Quantity
e1	Contract Line Creation	2017-07-11 9:00	c1					CO01				
e2	Contract Material Group Changed	2017-07-14 11:00	c1					CO01				
e3	Purchase Requisition Line Created	2017-07-15 12:00	c1	rq1				A456				
e4	Contract Line Creation	2017-07-15 14:00	c2					CO01				
e5	Purchase Requisition Line Created	2017-07-15 17:00	c2	rq2				A457				
e6	Purchase Order Line Creation	2017-07-16 15:00	c1		o1			A458	100	7	100_L50	
e7	Contract Line Creation	2017-07-16 16:00	c3					CO01				
e8	Purchase Order Line Creation	2017-07-17 15:00		rq1	o2			A458	200	8	100_L51	
e9	Purchase Order Line Creation	2017-07-18 15:00		rq2	o3			A458	300	8	100_L52	
e10	Goods Line Registered	2017-07-22 15:00			o1	r1		A456	100	7	100_L50	10
e11	Invoice Receipt	2017-07-22 16:00					i1	A125				
e12	Requisition Group Changed	2017-07-22 19:00		rq1				A456				
e13	Purchase Order Line Creation	2017-07-23 9:00		rq1	o4			A458	600	8	100_L51	
e14	Purchase Order Line Creation	2017-07-23 12:00	c3		o5			A458	600	8	100_L51	
e15	Goods Line Registered	2017-07-23 15:00			o2	r2		A456	100	8	100_L50	10
e16	Invoice Registered	2017-07-29 11:00				r1,r2	i1	A125				10
e17	Invoice Cleared	2017-07-30 12:00					i1	A125				
e18	Goods Line Registered	2017-07-31 15:00			o4	r3		A456	600	8	100_L51	10
e19	Goods Line Registered	2017-08-09 15:00			o5	r4		A456	600	8	100_L51	10
e20	Invoice Registered	2017-08-10 11:00				r2,r3,r4	i2	A125				10
e21	Invoice Cleared	2017-08-15 14:00					i2	A125				
e22	Goods Line Registered	2017-08-16 15:00			o3	r5		A456	300	8	100_L52	5
e23	Requisition Supplier Changed	2017-08-16 17:00		rq2				A456				
e24	Invoice Registered	2017-08-18 11:00				r5	i3	A125				5
e25	Invoice Cleared	2017-08-20 14:00					i3	A125				



(a) Energy event log.

(b) IT event log.

Fig. 1. ER-diagram representing the cardinality between the different object types in the two considered object-centric event logs. For each object type, the cardinality with the subsequent or the previous object type is represented as (*min_cardinality*, *max_cardinality*).

respectively (cf. events $e3$ and $e5$). In general, a contract can be associated with multiple requisitions, while each requisition is at most associated with one contract. However, this can not be seen in the event log excerpt because of its small size.

The life-cycle process of an object of type *Contract* refers to the stipulation of a contract with a customer, possibly followed by a *Requisition*, which is an optional object activated with its life cycle when the order needs a purchase requisition. The *Order* life-cycle process consists of several activities representing mainly quantity, price, or date modifications of the order, eventually approved by the head of the department. The *Receipt* life-cycle process is then related to receiving the goods or services requested, followed by the *Invoice* life-cycle process, which includes everything related to payments. Some events are associated with a single object identifier. In contrast, others are associated with multiple object identifiers (i.e., so-called bridge events), enabling the synchronization and data exchange between the object's life-cycle processes. Figure 1 illustrates how objects are related to each other for synchronization and data exchanges for the aforementioned utility-provider company (see Fig. 1a) and for our second object-centric event log related to an American technology company (see Fig. 1b). Note that relationships can be of many-to-many or many-to-one nature.

2.2 Single-Id Event Logs and Predictive Process Analytics

The traditional predictive process analytics assumes that a trace is naturally composed by a sequence of events, namely a trace $\sigma \in \mathcal{E}^*$ where $\mathcal{E}$ is the universe of events. An event $e \in \mathcal{E}$ records the execution of an activity $\pi_{act}(e)$ that occurred at time $\pi_{time}(e)$. Events also assign values to attributes: $\pi_{vmap}(e)$ returns a function f such that $f(a)$ indicates that e assigns value $f(a)$ to attribute a.

Process predictive analytics aims to predict the KPI value of traces $\sigma \in \mathcal{E}^*$. The definition of KPI depends on the process domain, and hereafter it is abstracted as a function:

Definition 1 (KPI Function). *Let $\mathcal{W}_K$ be the set of possible KPI values. A KPI is a function $\mathcal{T}_L : \mathcal{E}^* \times \mathbb{N} \not\rightarrow \mathcal{W}_K$ such that, given a trace $\sigma \in \mathcal{E}^*$ and an integer index $i \leq |\sigma|$, $\mathcal{T}_L(\sigma, i)$ returns the KPI value of σ after the occurrence of the first i events.*[1]

Note that our KPI definition assumes it to be computed a posteriori when the execution is completed and leaves a complete trail as a certain trace σ. In many cases, the KPI value is updated after each activity execution, namely after the occurrence of a subsequence event. We can then define the prediction problem on single-id, traditional event logs:

Definition 2 (Prediction Problem on Single-id Event Logs). *Let $\mathcal{T}_L$ be a KPI function. Let $\sigma = \langle e_1, \ldots, e_k \rangle$ be the trace of a running case, which eventually will complete as $\sigma_T = \langle e_1, \ldots, e_k, e_{k+1} \ldots, e_n \rangle$. The prediction problem can be formulated as predicting the value of $\mathcal{T}(\sigma_T, i)$ for all $k < i \leq n$.*

In the Process Mining literature, this problem has been faced with different Machine Learning models [9]. The training set is composed of pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ encodes the independent variables (also known as **features**) with their values, and $\mathcal{Y}$ is the dependent variable with its value (i.e., the value to predict). Predictive process analytics requires a KPI definition $\overline{\mathcal{T}}$ as input (cf. Definition 1). Let $\mathcal{W}_K = img(\overline{\mathcal{T}})$ be the domain of possible KPI values (i.e., the image or co-domain of $\overline{\mathcal{T}}$: $\mathcal{Y} = \mathcal{W}_K$). Afterward, each prediction technique requires the definition of the domain $\mathcal{X}$ and a **trace-to-instance encoding function** $\rho : \mathcal{E}^* \rightarrow \mathcal{X}$, which maps each trace σ or prefix of it to an element $\rho(\sigma) \in \mathcal{X}$.

The prediction model is then trained off-line based on a data set $\mathcal{D}$ that is created from an event log L as follows: Each prefix σ of each trace $\sigma_T \in L$ generates one

---

[1] Given a sequence X, $|X|$ indicates the length of X.

distinct item in $\mathcal{D}$, consisting of a pair $(x, y) \in (\mathcal{X} \times \mathcal{Y})$, where $x = \rho(\sigma)$ and $y = \overline{\mathcal{T}}(\sigma_T, |\sigma|)$. Once the data set item of every trace prefix is created, the model is trained. The resulting prediction model (known as predictor) can be abstracted as an oracle function $\Phi_{\mathcal{D}} : \mathcal{X} \to \mathcal{Y}$.

3 Techniques for Object-Centric Process Predictive Analytics

The application of object-centric predictive analytics techniques requires to build a set of graph instances. Initially, a single graph is built, where the nodes are the events of the object-centric event log. The arcs are then added: an arc is added between a node/event e' and a node/event e'' if e' and e'' have at least one object in common. For instance, there is an arc between the events e_1 and e_2 in Table 1 because they share the object with identifier $c1$. Arcs are directed: the arc goes out e' and enters e'' if the timestamp of e' is smaller than the timestamp of e''.[2]

This graph is in fact not strongly connected: not every object is connected to every other object. The graph is thus partitioned into its strongly-connected components: each component becomes a graph instance.

As an example, let us consider the object-centric event log in Table 1. Figure 2 represents the two graph instances, namely the two strongly-connected components that have been found. For instance, the nodes in the first connected component represented by the event identifiers $e1$, $e2$, $e3$, and $e6$ are connected to each other as the object identifier $c1$ is in common.

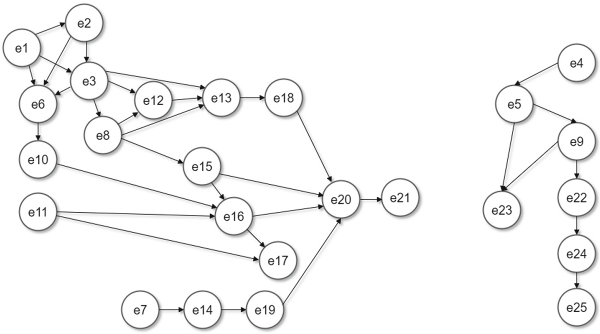


Fig. 2. The event graph extracted from the object-centric event log in Table 1. Each strongly-connected component becomes a graph instance, used to train predictors.

In this paper, we consider four techniques for object-centric process predictive analytics, which require a previous construction of the set of graph instances: GCN (Graph Convolutional Network), GGNN (Gated Graph Neural Network), LSTM, and Catboost. While Catboost and LSTM require the flattening of the graph instances, GGNN and GCN can directly take the set of graph instances as input during the training and test phase. The choice fell for Catboost and LSTM because they are seen to generally outperform other methods for predictive analytics [5,11,12]. The remainder introduces how the graphs are encoded when Catboost and LSTM models are used as well as via two types of graph-based neural networks.

---

[2] The special case of an arc between two events with the same timestamp can be dealt separately: in this case, the arc is bi-directional.

Catboost and LSTM Models. *LSTM* is a type of Recurrent Neural Network that uses gates to control the information flow over time. *Catboost* performs Gradient Boosting on Decision Trees [3]. Catboost performs at each iteration t a random permutation of the features and creates a tree based on it. The usage for predictive process analytics has firstly been reported in [5].

Both LSTM and Catboost require flattening: each graph instance is converted into a trace of a single-id event log (cf. Sect. 2.2). Given a graph instance g, the corresponding trace contains every event of g (recall that the g's nodes are events). To retain aggregated information about the interaction, each event, node of the graph instance, is extended with attributes (i.e., features) that summarizes the interaction. This is extensively discussed in [6], and here we limit to give an intuition. For each object type o and event e, we include an attribute that stores the number of events of type o to which e is connected in g. Also, for each numerical attribute a of every event connected to e, we compute the average value of a in the events connected to e.

The result is ultimately a single-id event log, and the prediction problem is as formulated in Definition 2 (cf. Sect. 2.2). The difference between LSTM and Catboost is related to a different definition of trace-to-instance encoding function $\rho : \mathcal{E}^* \rightarrow \mathcal{X}$.

In the Catboost learning domain, $\mathcal{X}$ is a vector that contains one dimension for each process activity, and one dimension for each log attribute. Given a trace σ defined over a set A of activities with a set V of event attributes, the encoding function is as follows:[3]

$$\rho_{Cat}(\langle e_1, \ldots, e_n \rangle) = \bigoplus_{a \in A} |\{e \in \sigma.\ \pi_{act}(e) = a\}| \oplus \zeta(e_n)$$

where $\oplus$ denotes the concatenation of two tuples and $\zeta(e)$ is the vector encoding of $e \in \sigma$:

$$\zeta(e) = \bigoplus_{v \in V} [\pi_{vmap}(e)(v)] \tag{1}$$

In the case of LSTM, $\mathcal{X}$ consists of sequences of vectors with n dimensions, where n is the number of event-log attributes: $\mathcal{X} = (\mathbb{R}^n)^*$.[4] Function ρ is then defined as $\rho_{LSTM}(\langle e_1, \ldots, e_m \rangle) = [\zeta(e_1), \ldots, \zeta(e_m)]$ where function ζ_σ is as introduced in Definition 1. Further details of the encoding for Catboost and LSTM are provided in [5].

Graph-Based Neural Network Models. *GCN* is designed to work on graph data [7]. Each log trace is represented as a graph, and we opted to represent the input graph as proposed in the work of [14]. Here, events of a prefix are represented as graph nodes.

GGNN integrates a Gated Recurrent Unit (GRU) cell that explicitly considers the temporal aspect of sequences [8]. We decided to represent the input graph as proposed in the work of [14]. Here, events of a prefix are represented as graph nodes, and edges are used to express relationships between the events of a prefix.

---

[3] To keep the explanation simple, we assume that the enumerations of all attributes $v \in V$ and all activities $a \in A$ are always returned consistently as if there is a total order among the variables and among activities (e.g., the alphabetical order).

[4] In literature, LSTMs are often trained based on matrices. However, a sequence of m vectors in $\mathbb{R}^n$ can be seen, in fact, as a matrix in $\mathbb{R}^{n \times m}$. We use here the data set representation as vectors to simplify the formalization.

In the case of the GCN, $\mathcal{X}$ is represented by a two-element tuple (AM, V); AM is an adjacency matrix storing which nodes (that in our case represent events) of the graph are connected by an edge and lies in $\mathbb{R}^{|V| \times |V|}$, while V is a node matrix storing features that describe the graph's nodes and lays in $\mathbb{R}^{|V| \times q}$, where q is the number of node features. Please notice that, for each event, i.e. node, of the GCN and GGNN, the node features have been encoded in the same way as LSTM.

In the case of the GGNN, $\mathcal{X}$ represents a three-element tuple (AM, V, EM). EM is an edge matrix that is added in order to store features that describe the edges of the graph and lays in $\mathbb{R}^{|V| \times p}$, where p refers to the number of features describing the edge. In particular, the edge between two events e' and e'' encodes the object in common between those of e' and e''. Edges are also characterized by a type: (1) *Repeat* (activity of a target event is equal to an activity of a source event), (2) *Backward* (activity of a target event was observed in a previous event of the current prefix), and (3) *Forward* (activity of a target event was not observed in previous events of the current prefix).

4 Evaluation Setup

The evaluation is based on the two object-centric processes described in Sect. 2.[5] The first object-centric process was executed by a well-known Italian utility-provider company, one of the major energy companies in Europe. The company focuses on the production/extraction of electricity and gas and on their distribution in different parts of the world. As mentioned in Sect. 2, this object-centric process runs through the intertwining of five different processes (i.e., object types): the *Requisition*, the *Order*, the *Invoice*, the *Contract*, and the *Receipt*.

The second object-centric process was executed by a well-known American technology company, one of the major companies worldwide. As mentioned in Sect. 2, this object-centric process runs through the intertwining of three different processes (i.e., object types): the *Requisition*, the *Order*, and the *Invoice*.

In a preprocessing phase, we removed attributes with missing values in more than 80% of the cases or attributes with the same values in all cases, and one of each pair of duplicate attributes (e.g., we removed the plant name, which is unique, and kept the plant identifier. Third, the large dimension of both companies is also reflected in the cardinality of some categorical attributes. For instance, for the utility-provider company, the codes of the materials shipped worldwide (*order_material_code*) are stored in an attribute that counts up to 4179 different values. We applied the 80-20 rule to reduce the cardinality of the attributes with thousands of different values [10]. Specifically, we kept the most frequent attribute values that covered 80% of the cases and labeled the remaining values as `other`. We considered several KPIs in our evaluation, grouped into three categories:

Elapsed Time between the first occurrence of the considered object type and the last occurrence of a selected target activity. The KPIs in this category are measured with respect to the first occurrence of an event that includes an object of the given time. For instance, regarding the utility-provider company, the first target

---

[5] The presence of NDAs prevent the authors from publicly sharing the datasets.

activity of interest is *SES Line Registered*. It indicates that the service requested by the customer is provided. However, as the customer can require several services, it is of interest to know when all the services requested are provided. The second target activity, *SES Line Released*, indicates that a further step is performed, which is the confirmation from the manager that everything is received correctly. Another interesting activity to be monitored is *Invoice Receipt*, which indicates that the invoice is correctly charged to the customer; conversely, *Invoice Cleared* indicates that the invoice is paid. Also for the technology company, two interesting activities to be monitored were *Invoice Receipt* and *Invoice Reconciled* that, similarly to *Invoice Cleared*, indicate that the invoice is paid. The third target activity of interest is *Invoice Submit*, which indicates that the Invoice is registered into the system. The last interesting activity to be monitored is *Invoice Approved*, which indicates that the invoice that is submitted for registration is approved by a manager.

Pay Delay estimation. It refers to the number of days exceeding the planned payment date, starting from the contract's creation to the last occurrence of *Invoice Cleared*.

Occurrence of activity/occurrence of attribute with a particular value. It refers to whether a certain activity or condition (e.g., a late payment) will occur in the future. This category is boolean, with true indicating the occurrence, and false the absence. First, the company is interested to know in advance whether there would be changes to the payment method (represented by the activity *Invoice Pay Method Changed*). When this activity happens, there are usually delays in payments. The company is also interested in predicting whether there will be problems with the order (represented by the activity *Purchase Order Blocked*) since this situation can bring additional delays caused by the reworks needed to fix the problem. Finally, it is interesting to know whether there will be delays with the payments (represented by the attribute *Pay Type* assuming value Late).

Table 2 enumerates the KPIs on which we performed experiments. In particular, Table 2a focuses different numerical KPIs, while Table 2b reports on boolean KPIs related to the occurrence of two activities and to the occurrence of one attribute taking on a certain value: for this table, we report the percentage of the graph instances for which the activity or the pair attribute-value is observed.

Each KPI definition used in our evaluation is of the form *from object type t to the last activity a*, to indicate that the KPI value is undefined for prefixes that ends *(i)* before the first event that incorporates objects of type t, or *(ii)* after the last event that refers to a. This means that, both in the train and test phases, we do not consider the prefixes for which the KPI values are undefined.

For the LSTM implementation we relied on the Keras framework and, as learning algorithm, we adopted ADAM with Nesterov Momentum (NAdam). For the GCN and the GGNN implementation, we used the Tensorflow framework, leveraging ADAM as learning algorithm. For all the three models, we selected 200 training epochs with a patience of 25. For the Catboost implementation, instead, we leveraged the open source library available at https://catboost.ai/.

The splitting in training and test set is performed on the set of graph instances that are obtained from the event log: two third of them are used for training, and one third for testing. During training, a hyperparameter optimization was performed, in which

Table 2. Descriptive statistics of selected KPIs.

(a) Numerical KPIs. Process 1 and 2 refer to the energy company and to the IT company, respectively. Values are reported in days.

Process	ID	KPI	Average	Standard Deviation
1	1	Elapsed Time from Contract to the last SES Line Registered	278.22	230.96
1	2	Elapsed Time from Contract to the last SES Line Released	279.42	230.94
1	3	Elapsed Time from Contract to the last Invoice Receipt	237.11	218.61
1	4	Elapsed Time from Contract to the last Invoice Cleared	287.14	229.69
1	5	Pay Delay estimation from Contract to the last Invoice Cleared	11.41	53.57
1	6	Elapsed Time from Order to the last Invoice Receipt	28.34	38.20
1	7	Elapsed Time from Order to the last Invoice Cleared	45.89	51.89
1	8	Elapsed Time from Requisition to the last Invoice Receipt	61.58	49.44
1	9	Elapsed Time from Requisition to the last Invoice Cleared	115.68	60.18
1	10	Elapsed Time from Requisition to the last SES Line Released	50.76	52.44
1	11	Elapsed Time from Requisition to the last SES Line Registered	49.79	52.25
2	12	Elapsed Time from Requisition to the last Invoice Reconciled	28.95	54.94
2	13	Elapsed Time from Requisition to the last Invoice Receipt	26.57	53.06
2	14	Elapsed Time from Requisition to the last Invoice Submit	27.65	54.31
2	15	Elapsed Time from Requisition to the last Invoice Approved	27.99	54.74
2	16	Elapsed Time from Order to the last Invoice Reconciled	37.65	71.45
2	17	Elapsed Time from Order to the last Invoice Submit	36.29	70.93
2	18	Elapsed Time from Order to the last Invoice Approved	36.74	71.40

(b) The percentage of graph instances in which the activity occurred, or the attribute is present with that value. KPIs are all related to the process for the energy company.

KPI	Percentage
Occurrence of Activity Purchase Order Blocked (from Contract to the last Invoice Cleared)	27%
Occurrence of Activity Pay Method Changed (from Contract to the last Invoice Cleared)	26%
Occurrence of Attribute Pay Type Assuming Value Late (from Contract to the last Invoice Cleared)	61%

we used the last 20% of the training set as a validation set. For LSTM, in particular, we validated the number of LSTM neurons used for each layer (which varied between 100 and 250), and the number of layers (1, 2 and 4), with a 20% dropout for each layer. We found that the best architecture consisted in most cases of two LSTM layers with 100 neurons each and a 20% dropout for each layer. For Catboost, conversely, we validated the number of trees used (which varied between 1500, 3000 and 4000) and the depth of each single tree (3, 6 and 10). We found that the best architecture consisted in most cases of 1,500 trees, each with a depth of 10.

Regarding the GCN implementation, the architecture was inspired by the work of [13], which included a GCN layer with one channel, followed by a global average pooling layer, a dropout layer with a dropout rate of 50%, two Dense layers with 256 neurons and $tanh$ activation, and a second dropout layer with the same dropout rate. In particular, we validated the number of channels in the GCN layer (which varied between 1 and 2), the number of final dense layers (we considered keeping the two layers or removing them) and the number of neurons for each dense layer (which varied between 100 and 250). However, we found the original architecture to be the most effective one.

Conversely, the architecture of the GGNN, was inspired by the work of [14]; the original architecture included a Gated Graph layer with four GRU cell iterations and $tanh$ activation, followed by a Global Attention layer with 100 output channels and three Dense layers with a dropout rate of 50% each and with 300, 200 and 100 neurons, respectively. In particular, we validated the number of GRU cell iterations in the Gated Graph layer (which varied between 1, 2, and 4), the number of final Dense layers (which varied between 1, 2, 3 or no layers at all) and the number of neurons for each Dense layer (which varied between 100 and 250). We found that the best architecture consisted in most cases of two GRU cell iterations and no final Dense layers at all.

We calculated the Mean Absolute Error (MAE) for the 18 numerical KPIs as values were reasonably well balanced. By contrast, we calculated the F1-Score for the last three KPIs. These KPIs are categorical and relate to activities the energy company wants to prevent. Finally, we report the training time required to train every prediction model for each KPI of interest.

5 Evaluation Results

Table 3 summarizes the results of our comparison. *We first observe that Catboost achieves the highest predictive accuracy among the four prediction models in almost all the considered KPIs, except for one case, where GCN obtains slightly better predictive accuracy. Also, Catboost models are learned significantly faster: summing up the training time of Catboost and GCN in Table 3 for every process and KPI, Catboost requires overall 7 h and 49 m, while GCN 59 h and 19 m (870%).*

We further compare the obtained results with the statistics of the selected KPIs in Table 2; in particular, we noticed that the event logs obtained for the numerical KPIs 3 to 7 and for the categorical KPIs 19 to 21 are those that contain more events. In these settings, the predictive accuracy of LSTM is closer to that of Catboost, and considerably outperforms those of GGNN and GCN. However, for KPIs 12 to 18 related to the second case study, while GGNN always performs better than GCN and there is not a clear winner between GGNN and LSTM, Catboost systematically outperforms other methods. Linked to the point above, LSTM can naturally learn from sequences of events, thus learning from the interaction among process objects. By contrast, GGNN and GCN tend to focus on adjacency matrices of nodes (i.e., events) in proximity, being less capable of reason on events that are indirectly connected. However, while the LSTM does not always outperform GGNN, Catboost systematically performs better than the other models because of the aggregated features, designed to capture the object-interaction [6]. Conversely, for the numerical KPIs 8 to 11, which are characterized by fewer events, the GCN outperforms LSTM and shows a predictive accuracy relatively comparable to Catboost. From this, we can conclude that the GCN can occasionally have slightly better performances in the presence of the limited amount of data. On the other hand, if enough data is provided, Catboost systematically outperforms graph-based approaches, which conversely struggle to learn more complicated interaction patterns, and also LSTM, which is known to require large amount of data.

Table 3. Predictive accuracy for KPIs and prediction models. An horizontal line split the Numerical and Boolean KPIs, with the former measured in terms of Mean Absolute Error (MAE) and the latter as F1-Score. Training times are reported in brackets.

Log	ID	KPI	GCN	GGNN	LSTM	Catboost
1	1	Elapsed Time from Contract to the last SES Line Registered	42.85 (1 h 5 m)	41.17 (1 h 45 m)	45.22 (3 h 34 m)	**31.9 (11 m)**
1	2	Elapsed Time from Contract to the last SES Line Released	42.31 (1 h 13 m)	47.99 (3 h 12 m)	50.77 (4 h 9 m)	**33.09 (21 m)**
1	3	Elapsed Time from Contract to the last Invoice Receipt	42.64 (2 h 59 m)	41.19 (8 h 7 m)	37.72 (5 h 23 m)	**30.78 (42 m)**
1	4	Elapsed Time from Contract to the last Invoice Cleared	47.86 (6 h 12 m)	44.37 (9 h 17 m)	39.05 (12 h 6 m)	**34.4 (49 m)**
1	5	Pay Delay estimation from Contract to the last Invoice Cleared	17.09 (2 h 12 m)	18.47 (10 h 4 m)	14.05 (8 h 53 m)	**12.36 (11 m)**
1	6	Elapsed Time from Order to the last Invoice Receipt	27.51 (2 h 29 m)	36.44 (5 h 45 m)	26.71 (2 h 47 m)	**19.15 (21 m)**
1	7	Elapsed Time from Order to the last Invoice Cleared	29.95 (7 h 56 m)	37.04 (11 h 50 m)	23.22 (4 h 19 m)	**20.08 (1 h 21 m)**
1	8	Elapsed Time from Requisition to the last Invoice Receipt	34.39 (41 m)	45.62 (33 m)	41.36 (56 m)	**31.08 (5 m)**
1	9	Elapsed Time from Requisition to the last Invoice Cleared	**35.33 (16 m)**	67.04 (27 m)	40.96 (2 h 15 m)	36.71 (17 m)
1	10	Elapsed Time from Requisition to the last SES Line Released	32.23 (5 m)	75.21 (20 m)	55.4 (49 m)	**26.96 (3 m)**
1	11	Elapsed Time from Requisition to the last SES Line Registered	31.52 (5 m)	81.62 (14 m)	48.97 (48 m)	**31.2 (6 m)**
2	12	Elapsed Time from Requisition to the last Invoice Reconciled	45.86 (2 h 56 m)	42.12 (4 h 47 m)	40.5 (1 h 16 m)	**27.5 (13 m)**
2	13	Elapsed Time from Requisition to the last Invoice Receipt	48.16 (3 h 10 m)	44.31 (4 h 17 m)	49.06 (1 h 9 m)	**29.44 (12 m)**
2	14	Elapsed Time from Requisition to the last Invoice Submit	47.88 (1 h 45 m)	43.02 (5 h 12 m)	43 (1 h 3 m)	**27.82 (13 m)**
2	15	Elapsed Time from Requisition to the last Invoice Approved	47.81 (3 h 54 m)	43.87 (3 h 50 m)	42.53 (1 h 2 m)	**28.75 (8 m)**
2	16	Elapsed Time from Order to the last Invoice Reconciled	51.3 (2 h 50 m)	46.6 (8 h)	50.60 (1 h 28 m)	**25.48 (9 m)**
2	17	Elapsed Time from Order to the last Invoice Submit	53.13 (4 h 51 m)	46.89 (10 h 31 m)	45.23 (42 m)	**24.02 (14 m)**
2	18	Elapsed Time from Order to the last Invoice Approved	51.61 (5 h 14 m)	47.14 (5 h 3 m)	47.49 (42 m)	**25.26 (14 m)**
1	19	Occurrence of Activity Purchase Order Blocked	0.33 (4 h 45 m)	0.37 (9 h 12 m)	0.51 (6 h 4 m)	**0.60 (20 m)**
1	20	Occurrence of Activity Invoice Pay Method Changed	0.38 (3 h 35 m)	0.50 (13 h)	0.64 (6 h 22 m)	**0.74 (20 m)**
1	21	Occurrence of Attribute Pay Type Late	0.73 (1 h 6 m)	0.75 (5 h 38 m)	**0.82 (7 h 12 m)**	**0.82 (19 m)**

When the KPI is related to the (non) occurrence of a process' activity (e.g., *Occurrence of Activity Purchase Order Blocked*) that is seldom observed (see KPIs 19 and 20), we observed that Catboost models significantly surpass graph-based neural networks, which are also outperformed by LSTM networks. When the activity is more common, graph-based neural networks show better predictive accuracy, which however usually remains lower than that of LSTM and Catboost models.

We can finally conclude that, when gradient boosting is used, the engineering of features to encode the object interaction enables obtaining prediction accuracy that is higher than that of graph-based neural networks. The preference of gradient-boosting-based techniques over those leveraging on graph-based neural network is further testified by the fact techniques relying on graph-based neural networks require a training time that is eight times longer. The comparison of graph-based neural networks and LSTM's shows that there is no clear winner: sometimes the former perform better, other times the latter does. This does not fully confirm the work by Adams et al. [2], which reported on the superiority of graph-based neural networks. But they only conducted two case studies, likely insufficient to reach more general conclusions.

This is likely partially linked to the manual engineering of object-interaction features, which is an informed tuning for the specific problem of object-centric process predictive analytics. Note how this paper does not flatten the event log as in [6] when Catboost or LSTM models are employed: here, graph instances are still created and encoded as sequences.

Graph-based neural networks conversely are general purpose, and are not informed on the specific characteristics of the graphs that encodes the interactions. Internally, they need to learn an abstraction of the graph structure, and no specific known characteristics of these graph instances can be leveraged on for a better abstraction. This yields lower accuracy, as well as the effort to learn an abstraction is paid via higher training time.

6 Conclusion

The object-centric process paradigm is increasingly gaining popularity in academia and industry. According to this paradigm, the process is seen as the interplay of numerous processes that constitute the life cycles of different objects of various types, where these life cycles periodically synchronize. The presence of many-to-many interactions between objects (i.e., processes) prevents the direct application of existing techniques, designed for traditional processes with one single id and execution flow.

This paper reports on the experience of comparing the predictions of two techniques based on gradient boosting on decision tree or LSTM against two based on graph neural networks. The four techniques were empirically evaluated on event logs related to two real object-centric processes, and more than 20 different KPI definitions. The results illustrate that the technique based on gradient boosting generally shows the highest accuracy, likely thanks to a proper engineering of the features to encode the object interaction. At the same time, it is more than 800% faster than the techniques relying on graph-based neural networks. For the two adopted datasets and the 21 KPIs employed in our experiments, gradient boosting on decision trees is more suitable than graph-based neural networks for object-centric process predictive analytics.

Note how the Gradient-Boosting technique discussed in this paper does not use the event-log flattening solution proposed in [6]: in this paper, we still represent the objects' interaction via graph instances, and the sequence conversion is only done as last step, retaining a set of features that meaningfully encode the graph-like structure.

Future work aims to conduct additional experiments on different publicly-available datasets and to also perform cross-validation to further validate the findings reported here. We also plan on testing different manual engineering of the object-interaction features; it cannot be excluded, indeed, that the features that so far we manually engineered are not the right abstraction of the object interaction in every case study. Moreover, we cannot exclude that a different trace encoding function for LSTM and graph-based neural networks could improve the performances of these models.

Acknowledgement. This research is partly funded by Department of Mathematics of University of Padua, through the BIRD project "Data-driven Business Process Improvement" (code `BIRD215924/21`).

References

1. van der Aalst, W.M.P.: Object-centric process mining: dealing with divergence and convergence in event data. In: Ölveczky, P.C., Salaün, G. (eds.) SEFM 2019. LNCS, vol. 11724, pp. 3–25. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30446-1_1
2. Adams, J.N., Park, G., Levich, S., Schuster, D., van der Aalst, W.M.P.: A framework for extracting and encoding features from object-centric event data. In: Troya, J., Medjahed, B., Piattini, M., Yao, L., Fernandez, P., Ruiz-Cortes, A. (eds.) Service-Oriented Computing. Lecture Notes in Computer Science, vol. 13740, pp. 36–53. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20984-0_3
3. Dorogush, A.V., Ershov, V., Gulin, A.: CatBoost: gradient boosting with categorical features support. In: Proceedings of the Workshop on ML Systems at NIPS 2017 (2017)
4. Farhang Ghahfarokhi, A., Park, G., Berti, A., van der Aalst, W.M.P.: OCEL Standard (2020). http://ocel-standard.org/1.0/specification.pdf
5. Galanti, R., et al.: An explainable decision support system for predictive process analytics. Eng. Appl. Artif. Intell. **120**, 105904 (2023)
6. Galanti, R., de Leoni, M., Navarin, N., Marazzi, A.: Object-centric process predictive analytics. Expert Syst. Appl. **213**, 119173 (2023)
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint: arXiv:1609.02907 (2017)
8. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. In: Proceedings of the 4th International Conference on Learning Representations (ICLR) (2016)
9. Márquez-Chamorro, A.E., Resinas, M., Ruiz-Cortés, A.: Predictive monitoring of business processes: a survey. IEEE Trans. Serv. Comput. **11**(6), 962–977 (2018)
10. Newman, M.E.J.: Power laws, Pareto distributions and Zipf's law, vol. 46, pp. 323–351. Taylor & Francis (2005)
11. Park, G., Song, M.: Prediction-based resource allocation using LSTM and minimum cost and maximum flow algorithm. In: Proceedings of the International Conference on Process Mining, ICPM 2019, Aachen, Germany, 2019, pp. 121–128. IEEE (2019)
12. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 477–492. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_30
13. Venugopal, I., Töllich, J., Fairbank, M., Scherp, A.: A comparison of deep-learning methods for analysing and predicting business processes. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2021)
14. Weinzierl, S.: Exploring gated graph sequence neural networks for predicting next process activities. In: Marrella, A., Weber, B. (eds.) Business Process Management Workshops. Lecture Notes in Business Information Processing, vol. 436, pp. 30–42. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94343-1_3

3rd International Workshop on Change, Drift, and Dynamics of Organizational Processes (ProDy 2023)

3rd Workshop on Change, Drift, and Dynamics of Organizational Processes (ProDy 2023)

The Workshop on Change, Drift, and Dynamics of Organizational Processes (ProDy) brings together researchers that are interested in how socio-technical processes are enacted and change over time. The workshop developed from the Workshop on Business Process Management and Routine Dynamics held in 2021 [1] and 2022 [2]. There are at least three perspectives on socio-technical processes that the workshop aims to encompass. First, the workshop aims to attract managerial contributions that explain how organizations can recognize and capitalize on change opportunities in processes. Second, we encourage algorithmic contributions that help in detecting and making drift in processes visible. Third, we also welcome contributions that further develop our understanding of why and how processes change over time. We organized and held the 3rd Workshop on Change, Drift, and Dynamics of Organizational Processes in conjunction with the 21st International Conference on Business Process Management in Utrecht, the Netherlands. We received a total of eight submissions. Each submission underwent regular single-blind peer-review by three members of the program committee. We accepted four papers for presentation at the workshop and organized a panel to reflect on the past and future of research on organizational processes.

The workshop comprised two sessions. In the first session, authors presented their respective papers. Vollenberg et al. (paper was not included in the proceedings) discussed how organizational actors reacted to unintended affordances. Proper et al. [3] outlined how simulation techniques can be added to process models. Aredes et al. [4] proposed a method to create a process architecture. Last, the paper by Sliż [5] examined how a shared service center employed ambidextrous business process management.

The second session was dedicated to the panel entitled "Looking back on the last 20 years of research on change, drift, and dynamics of organizational processes: What's coming next?" The panelists Daniel Beverungen, Fleur Deken, Brian Pentland, and Moe Wynn first shared their perspective on the most important advances in process research during the last 20 years and what they believe are the largest opportunities and challenges for the years to come. Afterwards, we moved towards an open discussion with the audience, moderated by one of the workshop organizers.

We would like to thank everyone who made the workshop possible. We would like to thank all authors, the panelists, all committee members, the workshop chairs, and all attendees. For the next iterations of the workshop, we encourage all research that, in one way or another, expands our knowledge of change, drift, and dynamics of organizational processes. We encourage theoretical as well as empirical papers and engineering-related as well as managerial contributions.

September 2023

Bastian Wurm
Waldemar Kremser
Jan Mendling

References

1. Wurm, B., Grisold, T., Kremser, W., Mendling, J.: Workshop on business process management and routine dynamics. In: BPM 2021 Workshop Proceedings. LNBIP (2021)
2. Wurm, B., Grisold, T., Kremser, W., Mendling, J.: Workshop on business process management and routine dynamics. In: BPM 2022 Workshop Proceedings. LNBIP (2022)
3. Proper, H., Zhu, Q., Ravesteijn, P., Gielingh, W.: Adding dynamic simulation to business process modeling via system dynamics. In: BPM 2023 Workshop Proceedings. LNBIP (2023)
4. Aredes, E., Lourenção, M., Burlton, R., Reijers, H., Pádua, S.: A method for creating process architecture: driving organizational change. In: BPM 2023 Workshop Proceedings. LNBIP (2023)
5. Sliż, P.: Ambidextrous business process management: unleashing the dual power of innovation and efficiency. In: BPM 2023 Workshop Proceedings. LNBIP (2023)

Organization

Workshop Chairs

Bastian Wurm	LMU Munich School of Management, Germany
Waldemar Kremser	Johannes Kepler Universität Linz, Austria
Jan Mendling	Humboldt-Universität zu Berlin, Germany

Program Committee

Han van der Aa	University of Mannheim, Germany
Annalisa Appice	University Aldo Moro of Bari, Italy
Christian Bartelheimer	Paderborn University, Germany
Markus Becker	University of Southern Denmark, Denmark
Iris Beerepoot	Utrecht University, The Netherlands
Nicholas Berente	University of Notre Dame, USA
Daniel Beverungen	Paderborn University, Germany
Johannes De Smedt	KU Leuven, Belgium
Martin Matzner	Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Brian Pentland	Michigan State University, USA
Hajo Reijers	Utrecht University, The Netherlands
Christoph Rosenkranz	University of Cologne, Germany

A Method for Creating Process Architecture: Driving Organizational Change

Emerson Lima Aredes[1], Marina Lourenção[1], Roger Burlton[2], Hajo Reijers[3],
and Silvia Inês Dallavale de Pádua[1(✉)]

[1] School of Economics, Business Administration and Accounting at Ribeirão Preto, University of São Paulo, Ribeirão Preto 14040-905, Brazil
dallavalle@usp.br
[2] Process Renewal Group, BPTrends Associates, Vancouver, BC, Canada
[3] Utrecht University, 3584 Utrecht, CS, The Netherlands

Abstract. The present study aims to validate the proposed Process Architecture Creation (PAC) method for driving organizational change and facilitating process optimization. To achieve this objective, a systematic literature was conducted to obtain a theoretical basis on existing approaches to PA creation method proposition. To validate the proposed model, a specialized validation was conducted through a focus group consisting of nine Business Process Management (BPM) specialists. The results present a new PAC method composed of four stages: Stage 1, Identify the organization's business context; Stage 2, Define the PA; Stage 3, Align PA and Business Architecture; and Stage 4, Define the measurement and control mechanism. The PAC Method may help organizations involved in BPM promotion initiatives to create their own PAs by following the stages proposed in this work. According to the focus group specialists, the architecture created using this method will provide a holistic view of the organization's processes and routines, helps optimize problem-prone processes, support the establishment of change plans aligned with process alignment, aids in identifying the impact of necessary changes, and contributes to the implementation of specific technologies, such as robotic process automation. The originality of this work consists of the consolidation of the PA concept, which will provide benefits to the research field by serving as a basis for in-depth knowledge of the subject.

Keywords: Process Architecture · Business Process Management · Process Architecture Creation Method · PAC

1 Introduction

Process Architecture (PA) is a critical success factor of Business Process Management [1–6]. PA emerged as a tool to aid process-oriented companies in managing business processes. Thus, PA can be seen as a high-level organization model, representing the

---

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-50974-2_40.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 539–551, 2024.
https://doi.org/10.1007/978-3-031-50974-2_40

overall structure of business processes and their relationships [7]. Some authors highlight the importance of process architecture in driving organizational change [8, 9].

However, the [9] emphasize that despite the benefits, PA implementation are often questioned and challenged. Practitioners also report the several challenges found in PA construction: (1) Some initiatives fail because they start asking people, "What do you do?" instead of understanding who the stakeholders are, what are the products and services, and the value that is delivered [10]. (2) Process executors have difficulty with self-understanding in an end-to-end process perspective and tend to see small pieces of processes or inner-departmental processes. On the other hand, top managers have a clear understanding of the value chain but cannot unfold the macro view into interconnected sub-processes and processes [10]. (3) Another mistake is to bias the PA is to reflect the technology used, or to make a blind copy of a framework or reference model without paying attention to the company's knowledge and competitive advantages [10].

Thus, to overcome these challenges research on how to develop PA is needed. However, [9] indicate that in the literature, there is still no common understanding of PA. The works discussing PA creation approaches differ regarding PA concepts, the goal of its creation and the building techniques [7]. These divergences result from adopting different PA concepts and reasons to use it, which is directly reflected in our research gap: the lack of a consolidated PA creation method that helps create PA and overcome the highlighted challenges. Thus, there is need for a consolidated PA creation method that enables organizations create the PA to effectively implement BPM and navigate the complexities of process optimization and transformation. Given these divergences and lack of consensus, the main question that motivates this study is the following: **How to create and validate a method that supports developing Processes Architecture to implementing BPM**? Thus, this study aims to create and validate a method for developing Process Architecture. Therefore, the need was found first to consolidate a PA concept based on existing works and then propose a new PA creation Method to aggregate the strengths of the pre-existing models.

2 Research Method

The qualitative exploratory research method is used to provide better insight into the complex phenomenon under investigation [11]. The methods section is divided into three phases.

2.1 Phase 1 – PA Theoretical Review

The first phase consists in perform a theoretical review on PA in the BPM context that were communicated to the international scientific community and are indexed and available in high-impact and high-prestige scientific databases. Supplementary Table 1 presents the systematic literature review steps.

2.2 Phase 2 – The Creation of a New PA Method

After the review, the existing PA creation methods published in previous studies were evaluated according to the five PA relevant aspects by [12] considering if they met them on a scale from 0 to 5. The five relevant aspects where: (1) Relevant aspect 1: Hierarchical view of the processes, (2) Relevant aspect 2: Alignment between processes – end-to-end vision, (3) Relevant aspect 3: Alignment between processes and strategy, (4) Relevant aspect 4: Alignment between business processes and resources, (5) Relevant aspect 5: Measurement and control mechanism. Besides, the scale was described in Supplementary Table 2. The new method is proposed based on the phases and tools presented by the approaches that had the best ratings for each of the relevant aspects, presented at the results section.

2.3 Phase 3 – New PA Method Specialized Validation

To validate the new PA method, we conducted an online focus group with BPM specialists, as this method can contribute to understand perceptions, obtain important impressions and elaborate concepts [11]. The focus group was recorded and conducted on May 18, 2023 for 132 min. First, the moderator presented the method explaining its dimensions. Next, he asked questions to the participants about the alignment of the method dimensions layout and the points needed for further clarification and adjustments. The driving questions are presented in Supplementary Table 3. The focal group was conducted with 9 specialists on business process management. Supplementary Table 4 presents the data collection description.

3 Results and Discussions

The results were divided into two sections which are related to the three phases presented in the method: (1) Phase 1: PA theoretical review: analysis of existing approaches and (2) Phases 2 and 3: The creation and validation of a new PA method.

3.1 Phase 1: PA Theoretical Review: Analysis of Existing Approaches

To create the new PA method, the literature was reviewed to find existing approaches and understand their phases and objectives. After the filters in the article selection were applied according to the research method, 11 approaches that present PA creation methods or frameworks were selected. Table 1 shows a summary of each work found and their phases.

Table 1. Previous PA researches

Author	Results summary
[13]	A meta-model for a process mapping language is produced as a result. The authors do not present a framework to create PA, instead they present a meta-model with the main elements of a process map design

(*continued*)

Table 1. (*continued*)

Author	Results summary
[14]	The authors do not present a framework to create PA, instead they present a meta-model with the main elements of a process map design (highest level of PA)
[15]	The authors analyze the data objects dependences between processes and their life cycles to purpose a data PA. This approach is based on knowing in details the processes maps and data objects flow
[16]	The authors first introduce a definition of an initial list of processes and then considers adequate criteria to define the priorities of these processes. From this point, it is possible to create the PA design with the identification of the processes from the crossing between cases and functions
[8]	PA is a collection of the business processes of the organization, and their interdependence provides a minimum contract of cooperation among all the processes. In their work, the authors do not present a systematic set of steps to create a PA, but they focus on the description of the possible types of relationships among the processes, especially regarding the trigger and the flow of information
[17]	An approach that aims to align the company's process portfolio with its strategic intent is presented. Strategic alignment is a mechanism for a company to visualize the relationship between its business processes and strategies. With it, the organization's decision makers can collect important pieces of information, based on its processes
[18]	A model of PA construction is based on four macroprocesses and their relationships. They are called macroprocesses because they contain several processes, sub-processes and activities that are needed to produce key products offered to customers, in addition to strategic plans, new infrastructure and so on
[19]	This work is part of an approach to promoting BPM in an organization, the Process Renewal Group Business Process Management Framework. In this framework, the PA is created in the second stage, "Architect & Align", with the objective of collaborating with the alignment of the organization's capabilities. Therefore, before effectively creating the PA, it is necessary to execute other activities to collect and align organizational information
[20]	The authors present a model to develop Business Architecture, the first layer of Enterprise Architecture (EA). This architecture is composed of a set of diagrams that picture a part or vision of the business, including PA, in a view that only considers the high-level processes of the organization
[21]	This approach aims to describe the organizational behavior, display its levels of decomposition and indicate models to clarify these levels. The framework helps model processes in BPM initiatives and also document the organization's PA because each level is hierarchically related and represents a view of the company at a higher or lower level of detail
[22]	The work by [22] introduces the systematic technique for role and interaction modelling (STRIM), a framework for creating a PA that uses the role activity diagram

3.2 Phase 2 and Phase 3: The Creation and Validation of a New PA Method

After presenting the theoretical review, the 11 articles were evaluated according to the five PA relevant aspects (RA) by [12] using the scale from 0 to 5, as presented in the methods section. Supplementary Table 5 presents the punctuation attributed to each work following the scale of five PA relevant aspects by [12]. The punctuation attributed to the preceding frameworks and methods led us to select the elements of the articles with the highest rankings. Thus, based on the previous methods and framework analysis, the relevant aspects of PA and validation with specialists, a new PAC method was constituted, as shown in Fig. 1. The underlined sentences indicate the changes suggested by the specialists that were incorporated into the proposed method. Moreover, below we present the conceptual explanation of the four stages proposed for the new PAC method. The studies presented next to the topics were the theoretical basis used to support its creation.

1. Identify the organization's business context

1.1 Identify organizational scope of the architecture
1.2 Identify stakeholders
1.3 Identify the relationship between stakeholders and organization
1.4 Identify strategic drivers, plans, and indicators

2. Define Process Architecture

2.1 Detail the exchanges between the organization and stakeholders
2.2 Define the lifecycle stages (birth, evolution and death) of stakeholders, their relationships with the organization and of the organization's assets
2.3 List and name the processes
2.4 Create process map
2.5 Group processes hierarchically (value chain, level 1, level 2, level n)
2.6 Classify the processes as Core, Enabling and Guiding
2.7 Identify reference model and benchmark

3. Characterize Business Architecture

3.1 Identify relationships among processes
3.2 Gather architecture models from each domain in the scope of the Business Architecture

4. Define measurement and control mechanisms

4.1 Identify the value of each process ("The ideal process X is the one that...")
4.2 Identify the organization's strategic indicators, goals, plans, objectives or scorecards
4.3 Define indicators based on the values of the processes
4.4 Identify which processes collect information to be used in strategic indicators
4.5 Define a performance monitoring mechanism (e.g., BAM)

Fig. 1. New proposed method for PA creation (PAC Method)

1 **- Identify the organization's business context**

This stage seeks to understand which are the external and internal factors to the organization that influence its processes. The external factors are necessary to understand "to whom" the process must deliver value, i.e., the stakeholders (e.g., customers, competitors, regulators and collaborators), and what are the exchanges between them and the organization (inputs and outputs). The internal factors are important to understand and align the strategic plans for the whole organization or the chosen value chain(s) in scope for the organization. The activities of these phase are:

1.1 *Identify organizational scope of the process architecture* [18, 20]

The first task is to define the scope of the Process Architecture (and also Business Architecture). Will we draw the PA of a small organizational unit? Will we focus on the entire organization? Will we include one or more Value Chains of the organization? Or will we even look for the processes that involve multiple organizations?

A good way to segment and partition the architecture is to focus on an outside-in value creation perspective that tackles the organization's Value Chains. The value chain is the collection of processes for the production of the goods and services that the firm offers to its customers [18]. The processes must be in an "end-to-end" view, i.e., begin with a market or customer requirement (e.g., an order, a product idea) and finish with the satisfaction of the requests [20]. No matter the choice, this part of the architecture effort must happen early to get everyone on the same page and to be able to negotiate and commit the time and resources appropriate for the effort needed. Regarding this topic, E3 highlighted the importance of determining a scope not too broad, because its difficult to identify the value delivery to clients and stakeholders, neither too narrow, with the risk of a value chain that delivers an "internal value".

1.2 *Identify stakeholders* [19]

At this phase, it is necessary understand with whom we interact. To create a list of stakeholders, the first questions to be answered regarding external connections are "Who cares about us?" and "Who do we care about?". This reinforces the Out-side-in perspective of BPM. According to [19], the classic top levels of stakeholders types are: Customers, Owners, Staff, Suppliers, Community, Competitors, Enterprise and Overlaps and Oddballs. E2 pointed that at this point it's important to be careful with egos from stakeholders that may not to be seen as Staff or Suppliers and that the most important is identifying these stakeholders than to define its type. E1 complemented saying that the aim of this phase is to identify "Who cares" with the organization in focus and its processes.

1.3 *Identify the relationship between stakeholders and organization* [18–20]

After listing the stakeholders, it is necessary to identify the relationship between the stakeholders and the organization, i.e., the main exchange delivered or received. For example, the main exchange delivered for the customer is the product or service offered by the organization. Some exchanges include: products, services, information, knowledge and commitments. With that, an organization's context diagram, as shown in Fig. 3, can be created by grouping the stakeholders into Customers, Suppliers, Regulators and Partners and describing the relationship between them.

1.4 *Identify strategic drivers, plans and indicators* [17, 19]

After identifying the factors external to the organization that guide processes, it is necessary to identify its internal drivers, such as mission, vision, organizational values and strategic plans. It is important also to align the understanding of terms such as “strategy”, “mission”, “vision”, “objectives” and “goals” within all organizations. The lack of understanding of these concepts also leads to a difficulty in communicating the higher statements of intent and approaching them to the lower levels of the enterprise, ensuring the traceability of performance tracking from bottom to top. E4 pointed out that this identification is not a formality, instead is fundamental to drive the processes changing plan.

2 **Define PA**

The main cautions to be taken in this phase are as follows: attempt to identify all processes; identify customer-focused processes “end-to-end” (in other words, do not identify processes that are focused and limited to functional areas, but focus on delivery value to customers); and do not break processes into “pieces” that are either too small to deliver direct value to the customer or too broad and involving different deliveries. At this point, both vertical (between departments) and horizontal (between hierarchical levels) gaps must be filled. The activities of these phase are:

2.1 *Detail the exchanges between the organization and its stakeholders* [19].

After identifying the relationship between the organization and stakeholders, we must detail all the exchanges within them (what is delivered to and received from each stakeholder). The exchanges will represent the organization’s assets, and the greater the detail is, the more processes may be identified [19]. Here, we expect to find that an incoming item will often be paired with one or more outgoing exchange items. For example, a request for credit may come in, and a rejection or acceptance may go out in response [19].

2.2 *Define the lifecycle (birth, evolution and death) of stakeholders, their relationships with the organization and the organization’s assets* [14, 19]

Processes move ‘things’ through a lifecycle of state of changes in the organization. Basically, the processes are responsible for coordinating the work of transforming supplies, information, stakeholders, products, services and so on. All of these things will be moved in a lifecycle that goes from its birth (when it is discovered by the organization), through transformation and to its death (when it is out of the organization’s hands). Between these extremes, there are a series of state changes that require someone to do something to move them on to the next progressive state [19]. For example, a customer can be discovered as a Lead, transformed to Prospect and Customer, be treated by After-sales and Cleaned from the database. Additionally, every product has a life-cycle of states as development, production, sales, assessment and stopping.

At this phase, we will define the life-cycle of every stakeholder, its relationship with the organization shown in the Context Diagram and (if more details are wanted in the PA) the exchanges between the Stakeholder and Organization as described in the Stakeholder-Organization Exchange Diagram. E2 highlighted that at this phase, and all the PA construction, is important to be performed with high level managers from the organization. That's because they have a wide view of the states, while people from operational tend to focus on its tasks and department scope, leading to short life-cycles.

2.3 *List and name the processes* [19]

Each state of the life-cycles has the potential to be converted into a process. One must consider the organization's reality and culture to determine if the names given in the life-cycle make sense as a process that contains a group of activities. Keep end-to-end thinking in mind.

E1, E4 and E5 were unanimous on point that to define a proper name may be one of the most difficult tasks in this phase. While E5 brought cases of the importance of names adherents with organizational culture, E2 pointed out that this is an important moment to broke old and wrong cultures and determine names that really means the objective of the process.

2.4 *Create processes map* [23]

At this point, it is possible to create or refine the process map (or macroprocess or value chain) showing the main processes of the organization with an end-to-end view, i.e., from clients request to their satisfaction. The main elements of the process map are, according to [14], core, support and management processes presented in a related way.

2.5 *Group the processes hierarchically (value chain, level 1, level 2, level n)* [18, 19, 21]

The work in the previous section tends to develop a list of processes that sometimes have different detail levels and can be decomposed/grouped into level 1, level 2 and possibly level 3 of detail, as shown by [18]. A challenge of this phase is to not mix the detail levels of the listed processes. Perhaps one phase of the life-cycle can be described as "stamping a document" and another as a "purchase". It is certain that "purchase" comprises many tasks while "stamping a document" can be considered an atomic task (cannot be divided). Group these tasks until find a process name that focuses on delivering value to costumer. Or, as E1 said, until the name has the meaning of the process objective.

2.6 *Classify the processes as Core, Enabling or Guiding* [18, 19]*;*

The listed processes can be classified in three main categories according to your proposal: (1) Core Processes (sometimes referred to as Operational Process), those that the customer is willing to pay for or receives a direct value based on the meeting of their needs and expectations. It defines the interactions that come from and go to the customers, consumers, and main value chain partners. (2) Guiding Processes (sometimes referred to as Management Processes), those who sustain competitive advantage by recognizing and responding to changes in the internal and external environment. They are the ones that typically involve owners, regulatory, or influencing stakeholders. (3) The Enabling Processes (sometimes referred to as Support Processes), those who send and receive reusable resources such as technologies, people, and facilities. They exist to provide resources to support the value creation for customers and are directly responsible for assuring that the Core processes work efficiently and effectively [19, 14].

2.7 *Identify reference models* [16, 18, 19]*;*

Frameworks such as SCOR, APQC or others related to the business, may be used to evaluate the processes identified, its names, check if is there missing processes or the grouping level. It is starting point evaluated the PA itself and the organization. SCOR is the longest running framework that takes the perspective of end-to-end processes and has been highly respected and adopted in companies for more than a decade [19].

3 **Characterize Processes**

It is important to understand the PA beyond the hierarchical view and understand the relationship among processes, its activities, recognize where and by whom they are executed in the organization, what are the resources that support them, costs, business rules etc. I.e., how its connected with the Business Architecture. The activities of these phase are:

3.1 *Identify relationships among processes* [24]

One of the PA definitions most commonly found in the literature is that it shows the relationship between the organization's processes. Making this relationship explicit ensures the end-to-end view of the processes and helps to fill the vertical (between departments) and horizontal (between hierarchical levels) gaps.

One process can present information to another through an information flow. It can also be linked to another process (or itself) through throwing or catching events, which in turn can trigger and instantiate new processes. The relationships may be classified as Type (trigger or flow of information) or Event (initial, intermediate or final) [24].

3.2 *Gather Architecture models from each domain in the scope of the Business Architecture (BA)* [19, 20]

As previously presented, PA is the core element of EA that must be built with the objective of showing as the work is (or should be) executed, who does it (human and IT resources), and how the performance is managed, i.e. should not only demonstrate link between systems, applications and, sometimes, processes [8]. Then, to identify the structure of other organization's domains, besides processes, is crucial to show how the work is done considering the restrictions of empowerment, policies and rules, technologies, facilities, human competences and organizational structure [19].

An important domain to consider is human resources, because they do the work with or without IT support. Also, technologies that support processes execution, as information systems and technical infrastructure, are a key domain to demonstrate how processes are done. Show the alignment between end-to-end processes and those technologies helps to orchestrate, to control and to execute daily activities [19, 20]. Process Name, Purpose, Inputs, Outputs, Processes that trigger (and events), Processes triggered (and events), Activities, Cost drivers, Decisions, KPIs and Resources. The following items comprise the process characterization: Process Name, Purpose, Inputs, Outputs, Triggering Processes (and associated events), Triggered Processes (and associated events), Activities, Cost Drivers, Decisions, Key Performance Indicators (KPIs), and Resources. These items can be presented in the Process Characterization Diagram (Supplementary Fig. 1).

4 **Define measurement and control mechanisms**

All work until now was necessary to create the Process Architecture. However, we can go further and use the PA as the basis of a process-oriented measurement and change the prioritization mechanism. To that end, it is necessary to identify the process indicators and how they are aligned with the organization's strategy or, in other words, which processes meet each of the organization's strategic objectives and goals. The activities of these phase are:

4.1 *Identify the organization's strategic indicators, goals, plans, objectives or scorecards* [19, 20]*;*

The first step to a measurement structure based on processes is to identify strategic indicators of the organization, because processes must guarantee the unfolding of strategies in daily actions. Until this point, it must have been done in the phase 1 of our PAC method, however some refinement might be necessary.

4.2 *Define indicators based on value of the process*[19, 20]*;*

Now, for each PA process, usually from the second level in hierarchy and not from the macroprocess level in the value chain, it is identified its value with the following statement: "The ideal process X is the one that (…)". It is important to keep in mind that indicators must be meaningful to the client and meaningful in a strategic point of view, and that is not worthy to create a huge list of indicators that will generate bureaucracy and over control, not aiming what matters to the client and stakeholders. With a hierarchical view of processes in PA and the relationship between processes and indicators, it is possible to know which process or group of processes contributes directly to reach each strategical objective. Thus, besides a monitoring mechanism, it is possible to see PA as a mechanism of change by knowing what process is necessary to change in order to reach determined strategic goal.

4.3 *Define a Performance monitoring mechanism,*

A dashboard or Business Activity Monitoring (BAM) [20] allows managers to see collected data in the process by typically following balanced performance indicators for effectiveness (value), efficiency (cost, lead-time), quality and agility (time to market and change).These mechanisms depend on correct data collection, which can be generated by different information systems and must be online, so the organization is able to make decisions during processes execution not only based on historic (looking at the rearview mirror) where execution cannot be changed. These phases can be represented through a Characterization Diagram that provides a general view of the process and helps align all related information in only one picture. It was created to facilitate the understanding of the process, conduct the identification of the indicators that really matter and serve as documentation for future references.

4 Conclusion

The objective of this study was to create and validate a new PAC Method to contribute to the implement BPM initiatives. Eleven studies presenting approaches to PA creation were identified in the literature review and evaluated with regard to their compliance with relevant aspects. The selected studies were used as theoretical basis for the development of new method to create a PA composed of four phases, that were validated by BPM specialists, indicating that the new PA method fulfilled its objective of contributing to implement BPM. According to focus group specialists, the architecture created using this method will provide a holistic view of the organization's processes and routines, helping optimize problem-prone processes, and supporting the establishment of change plans aligned with process harmonization. Additionally, it assists in identifying the impact of necessary changes and contributes to the implementation of specific technologies, such as robotic process automation.

The present study contributes to the literature by providing a systematic literature review on PA. The consolidation of the knowledge extracted from those works contributes to achieving the state of the art on the Process Architecture. Regarding the contribution to researchers and to organizational leaders, it is expected our PAC Method will help organizations to implement BPM initiatives, obtain a holistic view of the organization's processes and identify necessary improvements and changes. The PA built from this method stands out and can be used to optimize complex processes and deviations and implement specific technologies, such as robotic process automation, aligning teams around the value delivered to the customers.

Moreover, our PAC method is different from the others found in literature, as a practical guide, mostly for two reasons: (1) It has a set of detailed instructions and tools that can easy be reused, and (2) it aims to create a PA that supports the holistic approach needed in BPM, as it covers all relevant aspects of PA. One of the limitations of the present study was the consideration of only works published in scientific databases. Regarding opportunities for further research, new studies may consider overcoming the limitations presented and applying the proposed method in a case study composed of several implementation cycles for its validation.

References

1. Smart, P.A., Maddern, H., Maull, R.S.: Understanding business process management: implications for theory and practice. Br. J. Manage. **20**, 491–507 (2009). https://doi.org/10.1111/j.1467-8551.2008.00594.x
2. Rosemann, M., Brocke, J.: The six core elements of business process management (2010)
3. Burlton, R.: Business Process Management: Profiting From Process. Pearson Education, Indianapolis (2001)
4. Rub, F.A.A., Issa, A.A.: A business process modeling-based approach to investigate complex processes: software development case study. Bus. Process. Manag. J. **18**, 122–137 (2012). https://doi.org/10.1108/14637151211215046
5. Hajiheydari, N., Dabaghkashani, Z.: BPM implementation critical success factors: applying meta-synthesis approach. In: International Proceedings of Economics Development & Research, vol. 5, p. 38 (2011)

6. Biazzo, S., Bernardi, G.: Process management practices and quality systems standards: risks and opportunities of the new ISO 9001 certification. Bus. Process. Manag. J. **9**, 149–169 (2003). https://doi.org/10.1108/14637150310468371
7. Gonzalez-Lopez, F., Bustos, G.: Business process architecture design methodologies – a literature review. Bus. Process. Manag. J. **25**, 1317–1334 (2019). https://doi.org/10.1108/BPMJ-09-2017-0258
8. Eid-Sabbagh, R.-H., Dijkman, R., Weske, M.: Business process architecture: use and correctness. In: Barros, A., Gal, A., Kindler, E. (eds.) Business Process Management. Lecture Notes in Computer Science, vol. 7481, pp. 65–81. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32885-5_5
9. Niemi, E., Pekkola, S.: The benefits of enterprise architecture in organizational transformation. Bus. Inf. Syst. Eng. **62**, 585–597 (2020)
10. BPTrends: Business Architecture Essentials: 'Developing your Process Architecture: It's all about your work'. http://www.bptrends.com/business-architecture-essentials-developing-your-process-architecture-its-all-about-your-work/
11. Malhotra, N., Nunan, D., Birks, D.: Marketing Research: An Applied Approach. Pearson, New York (2017)
12. Aredes, E.L., de Pádua, S.I.D.: Process architecture as a BPM critical success factor: a bibliographic review. Bus. Manage. Rev. **4**, 245–255 (2014)
13. Poels, G., García, F., Ruiz, F., Piattini, M.: Architecting business process maps. Comput. Sci. Inf. Syst. **17**, 117–139 (2020)
14. Malinova, M., Leopold, H., Mendling, J.: An explorative study for process map design. In: Nurcan, S., Pimenidis, E. (eds.) Information Systems Engineering in Complex Environments. Lecture Notes in Business Information Processing, vol. 204, pp. 36–51. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19270-3_3
15. Eid-Sabbagh, R.-H., Hewelt, M., Meyer, A., Weske, M.: Deriving business process data architecturesfrom process model collections. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) Service-Oriented Computing. Lecture Notes in Computer Science, vol. 8274, pp. 533–540. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45005-1_43
16. Dumas, M., Rosa, M.L., Mendling, J.: Fundamentals of Business Process Management. Springer, New York (2013)
17. Morrison, E.D., Ghose, A.K., Dam, H.K., Hinge, K.G., Hoesch-Klohe, K.: Strategic alignment of business processes. In: Maglio, P.P., Weske, M., Yang, J., Fantinato, M. (eds.) Service-Oriented Computing, pp. 9–21. Springer, Berlin Heidelberg (2012). https://doi.org/10.1007/978-3-642-31875-7_3
18. Barros, O., Julio, C.: Enterprise and process architecture patterns. Bus. Process. Manag. J. **17**, 598–618 (2011). https://doi.org/10.1108/14637151111149447
19. Burlton, R.T.: Delivering business strategy through process management. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 2. IHIS, pp. 45–78. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-642-45103-4_2
20. Rummler, G.A., Ramias, A.J.: A framework for defining and designing the structure of work. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1: Introduction, Methods, and Information Systems, pp. 83–106. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-00416-2_4
21. Aitken, C., Stephenson, C., Brinkworth, R.: Process classification frameworks. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture, pp. 73–92. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-01982-1_4
22. Ould, M.A.: Designing a re-engineering proof process architecture. Bus. Process. Manag. J. **3**, 232–247 (1997). https://doi.org/10.1108/14637159710192284

23. Malinova, M., Leopold, H., Mendling, J.: An explorative study for process map design. In: Nurcan, S., Pimenidis, E. (eds.) Information Systems Engineering in Complex Environments. Lecture Notes in Business Information Processing, vol. 204, pp. 36–51. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19270-3_3
24. Eid-Sabbagh, R.-H., Dijkman, R., Weske, M.: Business process architecture: use and correctness. In: Barros, A., Gal, A., Kindler, E. (eds.) Business Process Management. Lecture Notes in Computer Science, vol. 7481, pp. 65–81. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32885-5_5

Ambidextrous Business Process Management: Unleashing the Dual Power of Innovation and Efficiency

Piotr Sliż[1,2](✉)

[1] University of Gdańsk, Armii Krajowej 101, Sopot, Poland
piotr.sliz@ug.edu.pl
[2] Systems Research Institute Polish Academy of Sciences, Newelska 6, Warszawa, Poland

Abstract. Ambidextrous Business Process Management (ABPM) is a strategic approach that aims to balance innovation and efficiency in managing business processes. This concept recognizes the need to simultaneously exploit existing processes and explore new opportunities for improvement. However, there is a need for a conceptual understanding of how changes in business processes and routines can be effectively described and studied within the context of ABPM. This article presents a case study conducted on ABPM in a Shared Service Center (SSC) in Poland. The study aimed to explore the essence of ABPM and provide insights into its practical implications. By addressing the research question of how changes in business processes and routines can be conceptually described and theorized, this article contributes to the body of knowledge in the field of ABPM. It offers insights into the theoretical foundations and principles underlying the effective management of change in processes and routines. The findings presented in this case study have implications for both practitioners and researchers in the field of BPM. Practitioners can derive substantial advantages from acquiring a profound comprehension of the strategic management of change in processes and routines within a SSC. Researchers can leverage the furnished conceptual framework and theoretical insights to propel the advancement of knowledge in the domain of ABPM, specifically in the context of SSC.

Keywords: Ambidextrous BPM · BPM · ambidexterity · exploitation · exploration

1 Introduction

In the ever-changing landscape of organizations, change is the only constant. The need to understand and navigate this continuous change has led to the emergence of a new field of study: process science. Process science aims to explore and influence the coherent series of changes that unfold over time, encompassing both man-made and naturally occurring processes at various levels. It is based on four key principles: placing processes at the center of attention, conducting scientific investigations of processes, embracing perspectives from multiple disciplines, and actively shaping the unfolding of processes to create

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 552–564, 2024.
https://doi.org/10.1007/978-3-031-50974-2_41

impact [20]. By emphasizing the importance of time and illuminating the role of tensions and contradictions, process studies shed light on the driving forces behind patterns of change and demonstrate how interactions across different levels contribute to organizational transformation [11]. ABPM, which integrates exploitation and exploration, holds promise as a framework for managing processes in such a hyperturbulent environment. It necessitates striking a balance between exploitation, focusing on optimizing and refining existing processes and exploration, emphasizing radical process change, process innovation and enabling new business models [see: 10]. This perspective calls for an integrative approach that acknowledges the interplay between exploitative and explorative activities within BPM, as well as the integration and alignment with strategic management concepts. However, despite its potential, ABPM remains an underexplored area in the literature. Previous studies have highlighted the need to bridge the cognitive gap and enhance understanding regarding the balance between exploitation and exploration in the BPM context. The extant literature reveals a conspicuous dearth of research pertaining to the subject of ambidexterity in Business Process Management (BPM) [2, 4, 5, 10, 17]. This article endeavors to partially address this research gap through the utilization of qualitative inquiry, thereby contributing to the advancement of knowledge in this domain. The need for research on ABPM becomes evident in the current landscape of business process management, as highlighted by the following statement: "*Ambidextrous BPM demands the shift of focus from exploitative to explorative BPM*" [15]. This indicates the necessity to investigate and understand the transition from traditional exploitative approaches to a more balanced and innovative perspective in BPM practices. This paper fills a research gap in the realm of ABPM by acknowledging its under researched status. It recognizes the need to enhance our understanding of ambidexterity in BPM and its implications for organizations. By doing so, this study aims to contribute to the field by proposing a conceptual framework for future research and practical applications.

The objective of this article is to provide insights into ABPM and propose a conceptual framework for future research and practice. By addressing this objective, we seek to advance the understanding of how organizations can effectively balance exploitation and exploration within BPM, adapt to the rapidly changing environment and drive sustainable performance. The focus of this article is to answer the research question: How can changes in business processes and routines be conceptually described and theorized? By delving into this topic, the article aims to bridge a cognitive gap in the existing literature. Understanding how changes in business processes and routines can be conceptually described and theorized is of paramount importance for effective process management in dynamic business environments. By presenting a conceptual framework of ABPM, this paper contributes to advancing the ongoing discussion on ABPM. And the balance between exploratory and exploitative actions in process management. This question provides a foundation for a comprehensive exploration of this topic, allowing for the identification of existing theories as well as the development of novel conceptual frameworks for effectively managing these changes. By shedding light on this aspect, the article contributes to the broader understanding of process management and facilitates the development of practical strategies to navigate the challenges posed by evolving business processes within the context of ambidexterity assumptions.

2 Ambidextrous Business Process Management (ABPM)

Ambidexterity in Business Process Management (ABPM) is a concept that has gained considerable attention in recent years. It enables organizations to implement innovation in their business processes while maintaining efficiency and operational excellence [8]. The core idea of ABPM revolves around the integration of exploitative and exploratory approaches to managing business processes. Exploitative BPM, also known as classical BPM, traditional BPM, static BPM, or commodity BPM, focuses on optimizing existing processes for efficiency and operational excellence (See: 1 (see: 10, 14, 18]. On the other hand, exploratory BPM emphasizes the pursuit of innovation and the exploration of new opportunities [8]. In the context of implementing ambidexterity, the role of projects is crucial. Projects serve as vehicles for organizations to explore new ideas, technologies, and business models, thereby supporting the simultaneous execution of operational and innovative activities. This becomes particularly important in dynamic business environments where organizations must balance the management of current processes with the search for new opportunities [17]. This paper identifies ABPM as a management concept that integrates exploitative and exploratory processes within a systemic and organizational structural framework. Achieving the effects of ABPM necessitates the reconfiguration of management subsystems and the organizational structure to enable ambidexterity. Integration, in this context, refers to a set of managerial actions aimed at dynamically balancing exploitative and exploratory activities. Several premises based on the literature support the consideration of an ambidextrous approach to process management development. Firstly, organizational ambidexterity is recognized as a new research paradigm in strategic management [19]. Secondly, there exist methodological foundations related to business process management (BPM), business process orientation (BPO) and process-oriented organizations [9], which provide a solid basis for understanding ambidexterity and its implications. Thirdly, the growing interest among researchers in the intersection of BPM and ambidexterity is evidenced by numerous works [2, 7, 8, 12, 13]. BPM, as an organizational dynamic capability, enables organizations to model, implement, optimize, and manage processes in response to a turbulent environment [8]. This highlights the potential of BPM to facilitate the integration of exploitative and exploratory processes, including projects [17]. Additionally, a rich body of literature exists on the integration of systemic organizational models such as the Viable System Model (VSM), the European Foundation for Quality Management (EFQM) model, and the New St. Gallen Management Model. Cohesion between BPM culture and the principles of creativity, flexibility, and risk-taking is crucial for the successful implementation of ambidextrous process management [10]. Organizations need to foster a culture that encourages employees to embrace change, think creatively, and take calculated risks. This alignment between BPM culture and the principles of ambidexterity enhances the organization's ability to balance exploration and exploitation However, several potential limiting factors must be considered in implementing ABPM. Firstly, the lack of integrated maturity models tailored specifically for ambidextrous organizations and ABPM poses a challenge. These models play a crucial role in assessing an organization's readiness and guiding its journey towards ambidexterity. Secondly, there is a limited number of publications focusing on exploratory ABPM, which hinders a comprehensive understanding of this aspect of ABPM. Finally, there is a need for further research on systemic and structural solutions for choosing an appropriate strategy for achieving ambidexterity in different organizational contexts [3]. During the comprehensive literature review, notable cognitive deficiencies were observed, impeding the progress of research on the conceptualization and operationalization of ABPM. The primary lacuna pertains to the limited discourse surrounding systemic models and supporting structures that foster the ambidextrous effect within the BPM domain. Moreover, there exists a conspicuous absence of scholarly deliberations concerning the scrutiny of information technology tools capable of concurrently facilitating exploitation and exploration. Although isolated IT tools catering to these distinct activities are available, the organizational domain lacks requisite solutions for effectively achieving a harmonious integration of exploitation and exploration. A pivotal aspect in evaluating the fragmented implementation of ABPM revolves around the dearth of maturity models enabling a comprehensive assessment of ABPM implementation levels.

3 Study Design

The study was conducted in 2023 at an international financial Shared Service Center located in Poland. Due to agreements with the participating organization, specific details about its operations cannot be disclosed as the acquired information pertains to both core and innovative activities. To protect the confidentiality and anonymity of the participating organization and its operations, special care was taken to ensure that no identifying details were disclosed. Measures were implemented to prevent the identification of the organization and to maintain the privacy of its activities throughout the research process. Additionally, steps were taken to secure and protect any collected data from unauthorized access or disclosure, in accordance with applicable data protection regulations. These ethical considerations were strictly adhered to to safeguard the organization's confidentiality. The Shared Service Center (SSC) functions as a large organization employing over 250 employees. Its subsidiary focuses on providing financial services to the organization's branches worldwide. The research employed an interview method involving five participants in managerial and expert roles within different teams responsible for various processes in both the exploitative and explorative layers. The objective of the study was to identify actions characteristic of exploitative and explorative processes, as well as exploitative BPM, explorative BPM and ABPM. The research process was divided into stages. In the first stage, based on a literature review, assumptions were formulated to enable the identification of the organizational process maturity level. To achieve this, the MMPM (Multi-Criteria Model for Assessing Organizational Process Maturity), which had been operationalized in Polish studies, was utilized. The model's assumptions were described in the article by [16]. A reconfiguration of the model's questions was performed compared to its original assumptions, employing a 5-point Likert scale to assess the degree of identification of selected symptoms facilitating the classification of organizations into one of the 5 levels of process maturity. The adopted MMPM model follows a classical 5-level structure. It has a descriptive nature primarily facilitating the evaluation of process implementation effectiveness. According to the model's assumptions, Level 1 represents a state in which BPM principles have not been implemented in the organization. Level 2 signifies that the main and supporting processes are mostly defined and formalized, while Level 3 indicates the establishment of measurement systems for most processes. Level 4 corresponds to a state in which managerial decisions are made based on the measurement system and process metrics. At this level, processes are managed according to BPM methodology. Level 5 represents a situation in the organization where processes are continuously improved and/or optimized. Both Level 4 and Level 5 are characterized not only by quantitative aspects related to KPIs but also by considerations concerning the wide prerogatives of process performers, such as the ability to introduce process changes and solve problems during process execution, emphasis on interdisciplinary process teams, reconfiguration of the organizational structure towards matrix and process-oriented solutions to leverage the benefits of process dynamism, knowledge management and knowledge transfer in the organization, organizational culture supportive of BPM, evaluation of external and internal training systems, considering the process as a key building block of the system and structure and acknowledging the role and significance of external and internal customers as drivers of process improvement efforts to ensure that the outcomes of the processes are aligned or as close as possible to customer expectations and needs [see: 16]. In a subsequent stage, an attempt was made to assess the level of BPM maturity in the studied organization. The study was conducted using a multi-hierarchical approach [see: 6] to evaluate the maturity level in selected areas. The assumption of a chain axiom was adopted, which allowed obtaining results indicating both Level 4 and Level 5. The adoption of this axiom enabled the classification of the organization into Level 4, indicating that the processes are defined, formalized, measured, and managed according to BPM principles. This result qualified the organization for inclusion in the study. Table 1 presents the characteristics of the assessment dimensions for the implementation of an ambidexterity strategy. The table distinguishes between exploitation activities (ES) and exploration activities (ER) in the context of business process management (BPM). These dimensions aim to capture the extent to which an organization focuses on either short-term profit generation and improvement of existing processes (exploitation) or long-term profit generation and the creation of new processes that enable the generation of new products and services (exploration).

Table 1. Characteristics of ambidexterity strategy implementation assessment

Exploitation	Exploration
ES1. BPM is geared toward short-term profit generation: In this context, BPM focuses on implementing strategies and initiatives that directly contribute to the generation of immediate profits. The emphasis is on identifying and optimizing processes that have the potential to deliver quick financial gains	**ER1.** BPM is geared toward generation of long-term profits: In this context, BPM focuses on strategies and initiatives that contribute to the sustainable and long-term profitability of the organization. The emphasis is on identifying and implementing processes that have the potential to generate consistent and enduring profits over an extended period

(*continued*)

Table 1. (*continued*)

Exploitation	Exploration
ES2. BPM is geared toward improvement of existing processes: The primary objective of BPM is to enhance the efficiency, productivity, and effectiveness of current processes. This involves analyzing and redesigning workflows, streamlining operations, eliminating bottlenecks, and implementing technologies to optimize performance	**ER2.** BPM is geared toward modeling new processes that enable the generation of new products and/or services: BPM emphasizes the development and modeling of innovative processes that facilitate the creation and introduction of new products and services. The focus is on exploring and implementing processes that enable the organization to stay competitive and meet evolving market demands
ES3. Processes are modeled for the long term: BPM takes a forward-thinking approach by modeling processes with a long-term perspective. This involves considering the organization's future goals and objectives, anticipating changes in the business environment, and designing processes that can adapt and evolve over time	**ER3.** Processes are modeled dynamically in the organization, depending on changes in the organization or its environment: BPM recognizes the need for agility and adaptability in process modeling. Processes are designed and adjusted dynamically to accommodate changes within the organization or its external environment. This allows the organization to respond effectively to new opportunities and challenges
ES4. The organization is seeking solutions and tools to increase process efficiency or effectiveness: The organization actively explores and adopts innovative solutions, technologies, and tools to improve the efficiency and effectiveness of its processes. This may include process automation, digital transformation, and the utilization of analytics to gain insights and drive continuous improvement	**ER4.** Solutions and tools are explored in the organization to increase the flexibility of processes (ability to dynamically reconfigure processes): BPM actively explores and adopts solutions and tools that enhance the flexibility and agility of processes. This includes technologies and methodologies that enable dynamic process reconfiguration, allowing the organization to quickly adapt to changing circumstances and requirements
ES5. Actions are taken in the organization to increase the market share of the products and/or services hitherto developed: BPM aligns with the organization's strategic objectives and one of its aims is to increase the market share of the products and services already developed. This involves identifying opportunities for growth, analyzing market trends, and implementing strategies to gain a larger market presence	**ER5.** Efforts are being undertaken in the organization to generate new products and/or services: BPM emphasizes the generation of new products and services as part of its strategic focus. The organization invests resources and effort into research, development, and innovation activities to create and introduce new offerings that cater to evolving customer needs and market trends

(*continued*)

Table 1. (*continued*)

Exploitation	Exploration
ES6. Supplier selection is contingent on their impact on process effect (products/services) cost reduction: When selecting suppliers, BPM considers their potential impact on reducing process costs and improving product/service effectiveness. Suppliers are evaluated based on their ability to provide cost-effective solutions and contribute to overall process optimization	**ER6.** Supplier selection in the organization is contingent on the potential to increase the innovativeness of the products and/or services offered or generate new products: When selecting suppliers, BPM considers their potential to enhance the innovativeness of the organization's products and services or contribute to the generation of new offerings. Suppliers are evaluated based on their ability to bring fresh ideas, technologies, and capabilities to support the organization's innovation efforts
ES7. Customer surveys are aimed at assessing the level of satisfaction with the products/services provided: BPM recognizes the importance of customer satisfaction and utilizes surveys to gauge the satisfaction levels of customers regarding the products and services delivered. The feedback obtained helps identify areas for improvement and informs decision-making to enhance customer experience	**ER7.** Customer surveys in the organization are aimed at identifying customer needs, in the context of new products and/or services generation: BPM places a strong emphasis on understanding customer needs and preferences, particularly regarding the development of new products and services. Customer surveys are conducted to gather insights and feedback that inform the organization's innovation and product/service generation processes
ES8. Measures are taken in the organization to reduce the cost of products/services while maintaining the same level of quality: BPM prioritizes cost reduction without compromising the quality of products and services. Various cost-saving measures, such as process optimization, supply chain improvements and resource utilization, are implemented to achieve a balance between cost efficiency and maintaining high-quality standards	**ER8.** Efforts are undertaken in the organization to implement ICT technologies, to reduce process execution parameters (e.g., cost, execution time, etc.): BPM recognizes the potential of information and communication technologies (ICT) in optimizing process execution. The organization actively explores and adopts ICT solutions to reduce process costs, shorten execution time and improve overall process efficiency and effectiveness
ES9. The organization employs material consumption analysis to explore solutions for reducing this consumption: BPM recognizes the significance of sustainable resource management. Material consumption analysis is conducted to identify opportunities for reducing resource usage, minimizing waste, and implementing environmentally friendly practices	**ER9.** Material consumption analysis is employed in the organization to explore technologies increasing the quality of the products and/or services offered: BPM recognizes the importance of material consumption analysis to enhance product and service quality. The organization conducts analysis to explore technologies and practices that can improve the quality of its offerings, including the efficient use of materials and resources

(*continued*)

Table 1. (*continued*)

Exploitation	Exploration
ES10. Process management is geared toward increasing the market share of the process outputs (products and/or services) hitherto generated: BPM aligns process management efforts with the goal of expanding the market share of the products and services produced. This involves analyzing market dynamics, identifying customer needs, and continuously improving processes to gain a larger market presence	**ER10.** Process management contributes to the search for areas generating new added value in the organization: BPM actively engages in process management to identify and capitalize on areas that can generate new added value for the organization. This involves analyzing and optimizing processes to identify opportunities for innovation, differentiation, and value creation, ensuring the organization remains competitive and achieves sustainable growth
ES11. Process management is geared toward the improvement of processes related to the organization's genotype activities (core activity): BPM prioritizes the improvement of processes directly linked to the organization's core activities or areas of expertise. The focus is on enhancing the efficiency, quality and innovation of processes that form the foundation of the organization's competitive advantage	

For each ES/ER question, respondents provided their agreement level with the statements listed in the table on a scale of 1 to 5 (1 - strongly disagree, 5 - strongly agree).

In the subsequent stage, an attempt was made to identify whether the studied organization (SSC) implements both exploitation and exploration activities, thus exhibiting characteristics of an ambidextrous organization. The research group responded to each of these questions using a questionnaire tool, which is presented in Table 1. In the research process, interviews were conducted with five employees from the surveyed SSC, including representatives from the management team and subject matter experts. Initially, they were asked to refer to the results concerning the maturity levels. Subsequently, they were requested to provide their understanding of exploration and exploitation in the context of Business Process Management (BPM), as well as terms such as exploratory business processes, exploitative (commodity) BPM, exploratory BPM and ABPM. After collecting the responses, the research findings were analyzed. The conclusions were formulated based on the interpretation of the gathered data, considering the predetermined assumptions and relevant literature. Conclusions regarding the maturity level of the organizational processes and the understanding of exploration and exploitation in the context of BPM were derived from the respondents' statements.

4 Results and Discussion

The first step involved identifying the maturity level outcomes using the BPM methodology. Out of the 5 respondents, 3 employees in managerial positions (team leaders) indicated level 5, while 2 specialists and experts indicated level 4. Based on the assumptions of the adopted methodology, level 4 was chosen. At this stage in the surveyed SSC, both core and support processes have been identified and defined through graphical and descriptive documentation. Both employees and managers correctly define the term "business process" within the organization and performance metrics are established for all processes. Metrics related to the level of satisfaction of both internal and external customers hold particular importance. The organizational structure in place is matrix-based, with vertical divisions responsible for operations, while the horizontal layer encompasses business processes and projects. Processes have their process owners who are accountable for the outcomes generated by the processes. Data collected from measurement systems are used to manage the processes. Based on this data, corrective decisions are made. Employees can contribute to process improvement by submitting modification proposals to the process owner through documentation. Process management is supported by IT tools. Next, during the interviews, participants were asked about their understanding of exploitative and exploratory activities. The respondents unanimously associated exploitation with core activities, which primarily involve the execution of business processes. On the other hand, exploration refers to the area of discovering new solutions within the organization, as implied by the term "mining." It encompasses research and development activities, the search for new ideas and the creation of innovations. Additionally, it includes aspects related to improving the core operations of the surveyed SSC. An example of an exploratory solution (exploratory activities) is the design of knowledge-intensive processes with low levels of standardization, closely resembling projects. These are processes where technologies such as artificial intelligence play a significant role. The management team emphasized in the interviews that the differences between exploitative and exploratory processes are determined by the type of technology used. For instance, Robotic Process Automation is applied to routine activities in exploitative processes, while technologies like artificial intelligence and supporting tools (such as chatbots, mid-journey solutions, etc.) facilitate exploratory processes related to creating new products or services. Examples of exploitative and exploratory processes are presented in Table 2.

The management team also pointed out during the interviews that, in their opinion, the separation of the exploitation layer from the exploration layer in process management primarily depends on the adopted Key Performance Indicators (KPIs) for the processes. Based on this, the respondents were asked to provide examples of metrics used for these processes. Table 3 presents the main KPIs for the previously mentioned processes.

During the interviews, particularly with managers and experts, they were asked to provide insights into the similarities and differences between managing business processes from an exploitative and exploratory perspective. Their responses helped construct a compelling narrative that captures these two management perspectives. Respondents generally understood that exploitative management focuses on efficiently executing existing business processes, aiming for operational excellence and optimization. On the

Table 2. Types of Processes in the investigated Shared Service Center

Business Processes in ABPM	
Exploitative Business Process	Exploratory Business Process
Human Resources Operations	Financial Risk Analysis
Local controlling	Market Research and Intelligence
Master data	Business Process Innovation
Accounts payable	New Product Development
Accounts receivable	New Service Development
General accounting	Competitive Analysis
Tax operations	Technology Assessment

Source: own elaboration based on study conducted in 2023.

Table 3. Example metrics (KPI) for exploitation and exploration business processes

KPI	
Exploitation	Exploration
Efficiency	Number of new products/services
Accuracy	Standarization level
Cycle time	Number of new ICT technologies
Error rate	Technology maturity level
Internal customer satisfaction	Market exploration index
External customer satisfaction	

Source: own elaboration based on study conducted in 2023.

other hand, exploratory management emphasizes innovation, experimentation, and discovering new business opportunities. The interviewees also highlighted significant differences in the structure and management processes. Exploitative management involves stable, standardized processes within a hierarchical organizational structure. In contrast, exploratory management encompasses more dynamic, flexible, and innovative processes, often relying on less formal organizational structures. Another notable difference lies in the approach to constraints and risk. Exploitative management prioritizes risk minimization, error elimination and maintaining operational stability. Constraints are precisely defined and controlled. In exploratory management, a higher level of risk is accepted, encouraging experimentation, and allowing for mistakes. Constraints are more flexible to stimulate creativity and innovation. Regarding change management, exploitative management introduces changes gradually and cautiously to minimize disruptions. It focuses on continuous improvement of existing processes. In contrast, exploratory management adopts a dynamic and experimental approach to change, concentrating on generating and testing new ideas, products, and services. Indicators and performance evaluation also differ in the context of exploitative and exploratory management. Exploitative management utilizes indicators related to operational efficiency, productivity, costs, and quality. Performance evaluation is based on the achievement of pre-defined goals and indicators. Meanwhile, exploratory management places greater emphasis on indicators measuring innovation and the ability to generate new ideas, products, or market opportunities. In summary, the interviews with managers and experts revealed that exploitative and exploratory management differ in several aspects. These differences stem from distinct business objectives, process characteristics, risk tolerance, and performance indicators. Understanding these distinctions enables organizations to better align their management strategies with their specific needs and contexts. The conducted interviews were of an unstructured nature, serving as a preliminary investigation to outline the assumptions for a quantitative study. The objective of the quantitative study is to precisely determine the organization's concentration on exploitation or exploration, or on ambidexterity. Subsequently, it is essential to develop a diagnostic tool that enables the assessment of which types of ambidexterity are utilized (structural, contextual, sequential, or hybrid) and to what extent process management elements are implemented in both exploitative and exploratory senses. Undoubtedly, the proposed study requires an expansion of qualitative research and the inclusion of knowledge and innovation management areas to capture, in the most precise manner possible, the identification of exploitative and exploratory processes, as well as the mechanisms enabling the balance of these two operational categories within the organization and their integration, for instance, with projects.

5 Conclusion

ABPM is a strategic approach that aims to balance innovation and efficiency in managing business processes. This article presents a case study conducted on ABPM in a Shared Service Center in Poland, providing insights into its practical implications, and addressing the research question of how changes in business processes and routines can be conceptually described and theorized. The study highlights the need for a conceptual understanding of how changes in business processes and routines can be effectively described and studied within the context of ABPM. By exploring the essence of ABPM, the article contributes to the field's body of knowledge and offers insights into the theoretical foundations and principles underlying the effective management of change in processes and routines. The findings of the case study have implications for both practitioners and researchers. Practitioners can benefit from acquiring a profound comprehension of the strategic management of change in processes and routines within a Shared Service Center, leading to substantial advantages. Researchers, on the other hand, can leverage the conceptual framework and theoretical insights presented in this article to advance the knowledge in the domain of ABPM, specifically in the context of Shared Service Centers. The article emphasizes the importance of balancing exploitation and exploration within BPM and integrating them with strategic management concepts. It highlights the potential of ABPM in managing processes in a rapidly changing environment by simultaneously exploiting existing processes and exploring new opportunities for improvement. In summary, this article expands the knowledge of ABPM, offering practical implications, a conceptual framework and insights for both practitioners and researchers. While limitations exist, it contributes to filling the cognitive gap in the literature and advocates for the balance between exploitation and exploration in managing business processes. The article also calls for future research to advance understanding and address the challenges associated with ABPM.

References

1. Ahmad, T., Van Looy, A.: Business process management and digital innovations: a systematic literature review. Sustainability. **12**, 6827 (2020). https://doi.org/10.3390/su12176827
2. Binci, D., Belisari, S., Appolloni, A.: BPM and change management: an ambidextrous perspective. BPMJ. **26**, 1–23 (2019). https://doi.org/10.1108/BPMJ-06-2018-0158
3. Constant, F., Calvi, R., Johnsen, T.E.: Managing tensions between exploitative and exploratory innovation through purchasing function ambidexterity. J. Purch. Supply Manag. **26**, 100645 (2020). https://doi.org/10.1016/j.pursup.2020.100645
4. Del Giudice, M., Soto-Acosta, P., Carayannis, E., Scuotto, V.: Emerging perspectives on business process management (BPM): IT-based processes and ambidextrous organizations, theory and practice. BPMJ **24**, 1070–1076 (2018). https://doi.org/10.1108/BPMJ-09-2018-336
5. Dezi, L., Santoro, G., Gabteni, H., Pellicelli, A.C.: The role of big data in shaping ambidextrous business process management: case studies from the service industry. BPMJ **24**, 1163–1175 (2018). https://doi.org/10.1108/BPMJ-07-2017-0215
6. Grajewski, P., Sliż, P., Jendza, D., Strumiłło, J.: The concept of multi-hierarchical assessment of process orientation implementation – production organization experience. In: SPSUTOM. 2022, pp. 201–227 (2022). https://doi.org/10.29119/1641-3466.2022.158.14
7. Grisold, T., et al.: The five diamond method for explorative business process management. Bus. Inf. Syst. Eng. **64**, 149–166 (2022). https://doi.org/10.1007/s12599-021-00703-1
8. Helbin, T., Van Looy, A.: Is business process management (BPM) ready for ambidexterity? Conceptualization, implementation guidelines and research agenda. Sustainability **13**, 1906 (2021). https://doi.org/10.3390/su13041906
9. Kerpedzhiev, G.D., König, U.M., Röglinger, M., Rosemann, M.: An exploration into future business process management capabilities in view of digitalization: results from a Delphi study. Bus. Inf. Syst. Eng. **63**, 83–96 (2021). https://doi.org/10.1007/s12599-020-00637-0
10. Kohlborn, T., Mueller, O., Poeppelbuss, J., Roeglinger, M.: Interview with Michael Rosemann on ambidextrous business process management. Bus. Process Mgmt J. **20**, 634–638 (2014). https://doi.org/10.1108/BPMJ-02-2014-0012
11. Langley, A., Smallman, C., Tsoukas, H., Van De Ven, A.H.: Process studies of change in organization and management: unveiling temporality, activity, and flow. AMJ. **56**, 1–13 (2013). https://doi.org/10.5465/amj.2013.4001
12. Lüftenegger, E.: Co-creating service-dominant business artifacts with action design research. RITA **28**, 63–77 (2021). https://doi.org/10.22456/2175-2745.107748
13. Mangialardi, G., Corallo, A., Lazoi, M., Scozzi, B.: Process view to innovate the management of the social housing system: a multiple case study. Sustainability **14**, 8294 (2022). https://doi.org/10.3390/su14148294
14. Mendling, J., Pentland, B.T., Recker, J.: Building a complementary agenda for business process management and digital innovation. Eur. J. Inf. Syst. **29**, 208–219 (2020). https://doi.org/10.1080/0960085X.2020.1755207
15. Rosemann, M.: Proposals for future BPM research directions. In: Ouyang, C., Jung, J.-Y. (eds.) AP-BPM 2014. LNBIP, vol. 181, pp. 1–15. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08222-6_1

16. Sliż, P.: Concept of the organization process maturity assessment. JEM **33**, 80–95 (2018). https://doi.org/10.22367/jem.2018.33.05
17. Sliż, P.: Process and project oriented organization: the essence and maturity measurement. In: Marrella, A., et al. (eds.) Business Process Management: Blockchain, Robotic Process Automation, and Central and Eastern Europe Forum. pp. 295–309. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16168-1_19
18. Szelągowski, M.: Process execution as a source of knowledge. Glob. J. Adv. Eng. Technol. Sci. **2**, 64–77 (2015)
19. Taródy, D.: Organizational ambidexterity as a new research paradigm in strategic management. In: Veztud, pp. 39–52 (2016). https://doi.org/10.14267/VEZTUD.2016.05.04
20. Vom Brocke, J., et al.: Process science: the interdisciplinary study of continuous change. SSRN J. (2021). https://doi.org/10.2139/ssrn.3916817

Adding Dynamic Simulation to Business Process Modeling via System Dynamics

H. A. Proper[1](✉), Q. Zhu[2], J. P. P. Ravesteijn[2], and W. Gielingh[3]

[1] TU Wien, Vienna, Austria
henderik.proper@tuwien.ac.at
[2] HU University of Applied Sciences Utrecht, Heidelberglaan 15,
3584CS Utrecht, The Netherlands
quan.zhu@hu.nl , pascal.ravesteijn@hu.nl
[3] Real Capital Concepts B.V., Tijgervlinder 26, 3863GP Nijkerk, The Netherlands
wim.gielingh@circularbuilding.com

Abstract. Business process modeling and system dynamics are different approaches that are used in the design and management of organizations. Both approaches are concerned with the processes in, and around, organizations with the aim to identify, design and understand their behavior as well as potential improvements. At the same time, these approaches differ considerably in their methodological focus. While business process modeling specifically takes the (control flow of) business processes as its primary focus, system dynamics takes the analysis of complex and multi-faceted systems as its core focus. More explicitly combining both approaches has the potential to better model and analyze (by way of simulation) complex business processes, while specifically also including more relevant facets from the environment of these business processes. Furthermore, the inherent ability for simulation of system dynamics models, can be used to simulate the behavior of processes over time, while also putting business processes in a broader multi-faceted context. In this paper, we report on initial results on making such a more explicit combination of business process modeling and system dynamics. In doing so, we also provide a step-by-step guide on how to use BPMN based models and system dynamics models together to model and analyze complex business processes, while illustrating this in terms of a case study on the maintenance of building facades.

Keywords: System Dynamics · BPMN · Facade Maintenance

1 Introduction

Business process models play a central role in the design and analysis of business processes. As its name suggests, business process modeling specifically takes the (control flow of) business processes as its primary focus. This does not mean that other aspects are not considered, but rather that the (control flow of) business processes are taken as the primary focus. Many different languages are used

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 565–576, 2024.
https://doi.org/10.1007/978-3-031-50974-2_42

to model business processes, where some modeling languages are more generic, while others are more explicitly dedicated to the task of modeling business processes. Examples include: Petri Nets [2], UML's - Activity Diagrams (UML-AD) [7], ArchiMate's business layer [9,12], and the Business Process Modeling Notation (BPMN) [5,14].

While business process modeling specifically takes the business processes as its primary focus, System Dynamics (SD) [20], as a method, takes the analysis of complex and multi-faceted systems as its core focus with the aim of providing advanced simulation models of the complex and multi-faceted dynamics of such systems. As such, both approaches can contribute to the design and management of processes in an organizational context. At the same time, the fundamental difference in the primary focus of these two approaches points at a potential benefit of explicitly combining the two approaches. More specifically, such an explicit combination would have the potential benefit of better modeling and analyzing complex business processes, while also including other relevant (dynamic) facets from the environment of these business processes. Examples of the latter facets include energy consumption, produced waste, workload on workers, usage of equipment, etc. Even more, the inherent ability of SD models to provide simulations of complex dynamics, can be used to simulate the behavior of the different involved processes (covering the business process as well as other relevant dynamic systemic facets).

The authors of [3] also argue, in a broad sense, that existing model-driven development related modeling languages, such as BPMN, need to be complemented with dynamic simulation. The authors of [16] and [10] second this by more specifically arguing the need for a complementary (SD-based) perspective of SD next to business process modeling notation in terms of hybrid process simulations across different levels of detail.

The potential relationship between business process modeling and system dynamics has been touched upon before in [16–19,21,22], where the focus has essentially remained on the control flow aspects only. The work as reported in [19,21,22] pioneered the joint grounding of (high-level) SD models and (more specific) business process models on top of a general purpose domain modeling language, with the aim to produce higher quality (in terms of their conceptual fidelity) SD and business process models [22, Fig. 1].

In this paper, we report on initial results towards a more explicit, and multi-faceted, combination of business process modeling and SD. More specifically, we will provide guidelines on how to use BPMN based models and SD models together to model and analyze complex business processes. These guidelines will be illustrated in terms of a case study in the context of maintenance of the facades of building, as part of a project in the Dutch construction sector aiming to improve the recycling of aluminum as used in building facades. This case will also illustrate how BPMN diagrams can be used as the starting point for creating SD models, as they provide a clear and comprehensive picture of the sequence of activities and events in a process. The resulting SD models, in turn, can then be used to simulate the behavior of the business process over time, while relevant additional (dynamic) facets of the operational environment of the business process can be included as well.

In line with this, the remainder of this paper is structured as follows. In Sect. 2 we provide a short introduction to SD. In Sect. 3 we then briefly discuss BPMN, while also more clearly differentiating between BPMN and SD regarding the abilities for (complex) simulations. Section 4 then elaborates the idea of using BPMN in concert with SD in terms of a mapping and associated guidelines. Before concluding, we illustrate these in terms of the case regarding the maintenance of facades in Sect. 5.

The mapping as discussed in Sect. 4 is actually a first (humble) iteration of a design science cycle [23] towards the design of a more complete integration between business process modeling and system dynamics. The case as used in Sect. 5 currently serves both as a case for the initial evaluation (towards the design cycle), as well as to identify ‘use cases’ (towards the relevance cycle) for such an integration and resulting simulation potential.

2 System Dynamics Modeling and Simulation

System Dynamics (SD) was originally developed by Jay W. Forrester at MIT Sloan in 1956 to study the behavior of systems. Presently, SD is used primarily to analyze complex systems by means of simulations [20]. It involves the construction of a model of a system in terms of feedback loops and other causal relationships, and then simulating the behavior of the system over time to gain insights into its dynamics and performance. SD can be used to understand how different facets of a system interact and how changes in one area may impact other areas. SD uses two diagram types to capture SD models: Causal Loop Diagrams (CLDs) and Stock-and-Flow Diagrams (SFDs). CLDs show the main *variables*, the system *boundaries* and the overall *structure* of the SD model. In the process of analyzing a system, these diagrams help to scope the system, to quickly capture hypotheses about the causes of dynamics, elicit and capture the mental models of individuals or teams, and communicate important feedback loops. CLDs show how systemic variables influence each other in terms of a qualitative (positive or negative) influence. However, they do not operationalize this in quantitative terms. Therefore, SFDs complement CLDs in terms of stocks and flows of quantitative accumulations of ‘things’ (materials, value, information, tasks, CO2 emissions, energy, etc) as they ‘move’ through a system [20].

3 BPMN and Simulation

BPMN [14] provides a graphical notation to create diagrams that show the flow of activities, decisions, and interactions between different actors. As stated in [14], the main goal of BPMN is to provide a notation that is understandable to all stakeholders involved in the design and analysis of organizational processes such as managers, business analysts, information managers, software developers, and end-users. As BPMN is often used in process improvement initiatives, all these roles are involved in identifying bottlenecks and inefficiencies in a process, while subsequently creating and implementing solutions.

Simulation is often used to help managers and analysts in understanding different solutions and deciding on which to implement. With the aim of assessing the effects of changes made to the processes and/or physical settings (e.g., the ability of resources to perform tasks), without disrupting current operations, simulation is a technique that may be used to understand the behavior of a system [10]. While there are BPM-tool suppliers that offer simulation functionality based on BPMN, Pereira and Freitas [15] found that, to be able to do so, specific elements for simulation needs to be incorporated, as BPMN is not designed for simulation as such. Based on a study into common simulation properties [4] and an analysis of 5 BPM-tools with simulation capabilities, several properties were found to be lacking [15]. None of the analyzed tools had the possibility to define periods of unavailability of resources. Also, functionality for defining transfer time, interruptions and execution priorities were only available in one tool (i.e., BPSim – Trisotech BPMN 2.0 Modeler for Visio version 4.2.0). Finally, the possibility to define an allocation plan for sharing of execution activities as well as stipulating the number of replications of the simulation were only provided by two of the five tools. Still, whether properties were found to be present or not does not say anything about the level of support by the tool, therefore these outcomes can only be taken as a first indication regarding the simulation capabilities and use of BPMN.

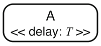
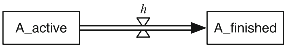
4 System Dynamics and Process Flows

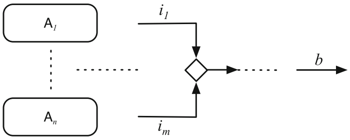
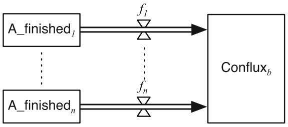
SD is often used for high-level organizational analysis, such as strategy development and analysis of policy options, where capturing information flow and feedback are essential considerations [10]. Simulation models that are based directly on BPMN can only perform analyses and decision making at an operational (discrete) level, c.f. [13]. SD, however, can be used on a more abstract level than BPMN in terms of a more continuous approach to everything in the process. For instance, for managers with the responsibility to predict future resource needs and anticipating what various options might cost their organizations, SD offers more than conventional (business process) flow modeling [8]. A conventional flow plan and some simulation in any of the conventional process modeling analysis tools will identify many problems facing organizations. However, a conventional flow plan would not provide much insight into a supply chain problem in which shipping costs, resource costs and personnel costs are all varying in different ways, each affecting the other. SD, on the other hand, is very useful for its feedback options that can influence various related parameters of the process like learning and improvement during the process, communication overhead, error rates and even increasing experience of human resources [11].

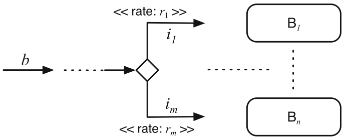
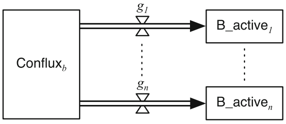
In principle, BPMN and SD share a common logic, i.e. sequential progression of activities from start to finish [10]. BPMN's flow-and-gate based notation are conceptually similar to SD's stocks and flows. Based on this, Table 1 provides three key mapping patterns to translate (the control flow of) BPMN to SFD.

The mapping as shown in Table 1 provides the *requirements* on the valves and flows as identified on the system dynamics side. We are currently investigating the most effective way (in terms of representation and simulations) to represent the semantics of the different BPMN gateways in terms of system dynamics.

Table 1. BPMN to SD mapping

BPMN	System Dynamics (SFD)
 $T \geq 0$	 $h(t) \triangleq \mathsf{Inflow}(\mathsf{A\_active}, t - T)$


no specific pre-conditions
	 $A\_finished_1$
	$A\_finished_n$
	f_1
	f_n
	$Conflux_b$
	base:
	for each $i \in \{1, \ldots, n\}$:
	$O_i(t) := \mathsf{Stock}(\mathsf{A\_finished}_j, t)$
	$R_i := 1$
	recurse:
	if AND-split, then for each $i \in \{i_1, \ldots, i_m\}$:
	$O'_i(t) := \mathsf{Min}_{j \in \{i_1, \ldots, i_m\}} O_j(t)$
	$R'_i := R_i/m$
	if (X)OR-split, then for each $i \in \{i_1, \ldots, i_m\}$:
	$O'_i(t) := O_i(t)$
	$R'_i := R_i$
	final:
	for each $i \in \{1, \ldots, n\}$:
	$f_i(t) \triangleq O_i(t) \times R_i$

 << rate: r_1 >>
i_1
b
i_m
<< rate: r_m >>
B_1
B_n
if AND-split:
$\forall_{1 \leq i \leq m} [r_i = 1]$
if OR-split:
$\forall_{1 \leq i \leq m} [r_i > 0] \wedge \sum_{i \leq i \leq m} r_i \leq m$
if XOR-split:
$\forall_{1 \leq i \leq m} [r_i > 0] \wedge \sum_{i \leq i \leq m} r_i = 1$
	 $Conflux_b$
	g_1
	g_n
	B_active_1
	B_active_n
	base:
	for each $i \in \{1, \ldots, n\}$: $R_i := 1$
	recurse:
	for each $i \in \{i_1, \ldots, i_m\}$: $R'_i := R_i \times r_i$
	final:
	for each $i \in \{1, \ldots, n\}$: $g_i(t) \triangleq \mathsf{Stock}(\mathsf{Conflux}_b, t) \times R_i$

The first row in Table 1 shows the mapping of a BPMN activity `A` to an SFD. As an activity would require a time $T \geq 0$ to finish after it is started, the mapping to an SFD needs to take this into account. Since such a mapping from BPMN to SFD requires this delay to be known, we have added it also to the BPMN side in terms of (a slight 'abuse' of) the stereotype notation: $\mathsf{<<delay} : T\mathsf{>>}$. On the SFD side, we have to create two stocks for activity `A`. One representing the instances of `A` that are `active`, and one for those that are `finished`. The flow between them is controlled by a valve h, which depends on the inflow of `A_active` in the sense that it releases, at point in time t, the entire inflow from $t - T$. The situation as shown in the first row is actually the basic situation in which there is a fixed waiting time of T for activity `A` to finish. More generally, however, one could also consider T as a probability distribution function over time, signifying how long it would take to complete an instance of task `A`, such that:

$$\int_0^\infty T(t)\,\mathbf{d}t = 1$$

For a fixed delay, we would then have $\exists!_t\,[T(t) = 1]$, where that (unique) t is the fixed delay. For this generalized approach, we could define h (when using a continuous time axis) as:

$$h(t) \triangleq \int_{-\infty}^{t} \mathsf{Inflow}(\mathsf{A\_active}, t - u) \times T(u)\,\mathbf{d}u$$

We take the view that the control flows between activities in BPMN essentially involve two kinds of trees: a *join-tree* with zero or more join-gateways, and a *split-tree* with zero or more split-gateways. In between a combination of such trees there will be one trigger b that bridges between the two trees. In the mapping to an SFD, the latter trigger leads to a stock we refer to as the $\mathsf{Conflux}_b$. This is illustrated at the top of the second and third rows of Table 1. In the case of a join-tree with sources $A_1, \ldots, A_n$, for some n, we have the SDF pattern with $\mathsf{A\_finished}_i$ stocks and a flow f_i to $\mathsf{Conflux}_b$. In the case of a split-tree with targets $B_1, \ldots, A_n$, for some (other) n, we have the SDF pattern right with $\mathsf{A\_active}_i$ stocks and flows g_i from $\mathsf{Conflux}_b$.

The lower parts of rows two and three define requirements for the flows contained in the SFD. For a join-tree (second row) we use a recursive definition following the structure of a join tree, involving O_i which defines the *outflow* from A_i into f_i at point in time t and ratio R_i with which the *outflow* from A_i is translated to an *inflow* for $\mathsf{Conflux}_b$. The actual outflow O_i depends on the combination of the different join gateways from the sources to trigger b bridging between the join-tree and split-tree. Depending on the join gateway, the outflow and inflow ratio need to be computed differently.

For the split-tree (third row) we also use a recursive definition. In this case, this only involves the rate R_i in which the stock from $\mathsf{Conflux}_b$ is turned into an inflow of the target stocks. This rate reflects the division of the stock of $\mathsf{Conflux}_b$ based on the probabilities associated to the different options of the split-gateways. On the left, we see the conditions of on these probabilities (r_i) depending on the kind of split-gateway.

In the case of a join-tree without any gateway, i.e. $n = 1$, we can optimize the resulting SFD by merging f_1 and $\mathsf{Conflux}_b$ into $\mathsf{A\_finished}_1$. Similarly, when the split-tree does involve gateways, but the join-tree does not, we can merge $\mathsf{Conflux}_b$, g_1 and $\mathsf{B\_active}_1$.

It is important to stress that the mapping of the control-flow from BPMN to SFD as shown in Table 1 is only intended to provide the 'backbone' of an

SD model, which can then be further complemented with other facets such as energy consumption, produced waste, workload on workers, usage of equipment, etc. However, for the flows related to the consumption/production of such facets, the flow(s) through the business activities (as based from the original BPMN model) will be the main driver.

As an integrated procedure to create a BPMN and SD model together, to analyze complex business processes in conjunction with all relevant facets, we propose the step-by-step guide as discussed in the remainder of this section. This guide extends the work of [10], in particular in step 3 (i.e., the way of mapping) and steps 6 to 8 (i.e., new steps). The guide is visualized in Fig. 1.

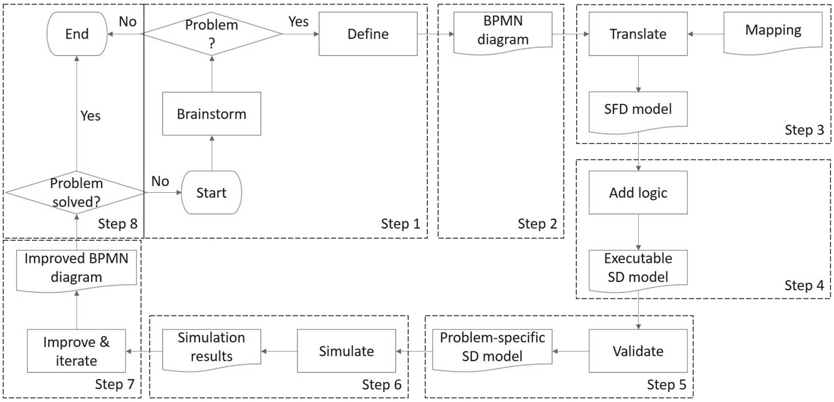


Fig. 1. Visualized step-by-step guide

Step 1: Define the scope and objectives of the process. Identify the key stakeholders, inputs, and outputs of the process, and define the performance metrics to evaluate the process.
Step 2: Create a BPMN diagram of the process using the standard notation by modeling the activities, decisions, and events involved in the process.
Step 3: Translate the BPMN model to the SFD via the mapping from Table 1.
Step 4: Add model logic and variables to the SFD to enable its execution as an SD model.
Step 5: Validate the SD model by comparing it to real-world data or historical performance data for the process. Ensure that the SD model accurately reflects the behavior of the process, and that it is consistent with the BPMN diagram. Note: in practice, this step should draw upon results from e.g. process mining in general [1], and process mining towards SD models in particular [18].
Step 6: Use the SD model to simulate the behavior of the process over time. Test different scenarios and strategies, and observe how the process responds to changes in key variables. Use the BPMN diagram to help interpret the results of simulations, and to identify opportunities for improvement.

Step 7: Use the insights gained from simulations to identify areas for improvement in the process. Modify the BPMN diagram as necessary to incorporate these changes, and test them using SD simulations. Iterate this process until identifying the optimal configuration for the process.
Step 8: Communicate the results of analysis to stakeholders using the BPMN diagram and other visualizations. Explain how the insights gained from simulations can be used to improve the performance of the process, and make recommendations for future improvements.

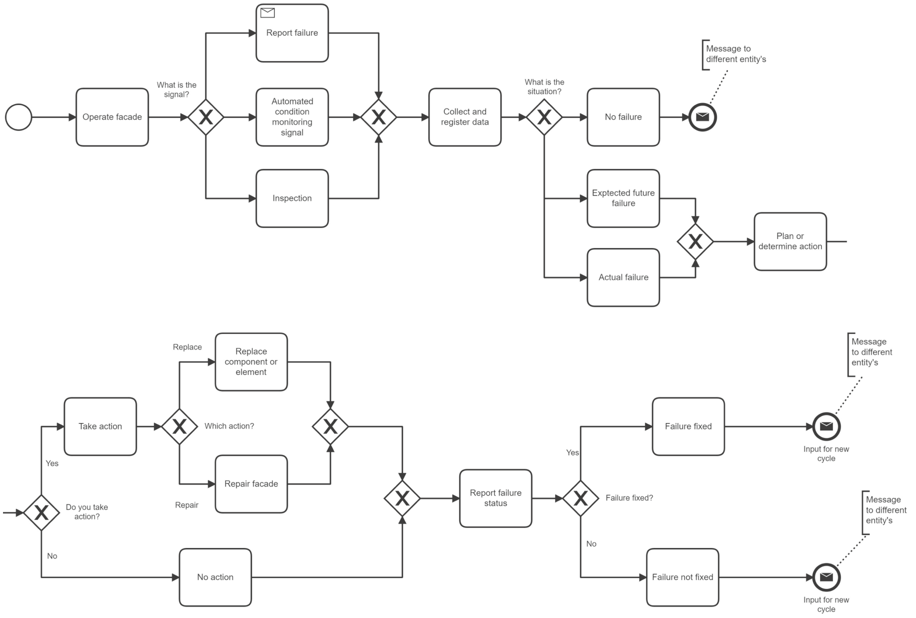


Fig. 2. Adjusted BPMN diagram for facade maintenance

5 The Facade Maintenance Case

The application of our proposed eight-step guide is illustrated by a case study from the project PerpetuAl, which is in the context of circular Aluminum chain. The aim of our case study is to demonstrate the management of facade panels of buildings. Given the confidentiality agreement, although our model is based on a real case, the presented structure and data is not case specific. The software to build our BPMN model was Camunda, while the software for our SD model was Vensim PLE.
Step 1: A group of four researchers visited our focal firm and interviewed two customer service (maintenance) managers to identify inputs and outputs

of facade construction and maintenance processes. The initial BPMN diagram was then created.

Step 2: Another researcher with multiple years of experience with BPMN modeling in practice then checked and adjusted the initial BPMN diagram, according to the standard notation. The adjusted BPMN diagram was then validated by the fourth Author of this paper. The adjusted BPMN diagram is shown in Fig. 2.

Step 3: The adjusted BPMN diagram was then translated according to the mapping provided in Table 1 by the second author of this paper. The resulting SFD is shown in Fig. 3.

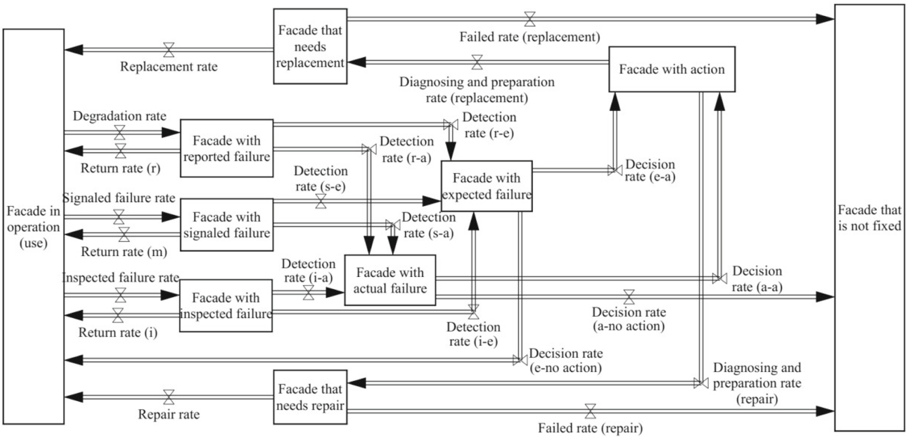


Fig. 3. Translated stock-and-flow diagram for facade maintenance

Step 4: Besides stocks and flows, other variables (mostly related to time/delay) and relationships (i.e. arrows) were added to the SD model (Fig. 4) to enable the simulation runs.

Step 5: Authors two and four acquired historical performance data from above-mentioned customer service (maintenance) managers and validated the SD model.

Step 6: Four scenarios were simulated. The base scenario, Scenario 1, is corrective maintenance (CM), which only repairs facades after failure. Scenario 2 is to combine CM with time-based maintenance (TBM). Scenario 3 is to combine corrective maintenance with condition-based maintenance (CBM). While Scenario 4, i.e. the full model, is to combine all three maintenance practices. The scenario simulation results are summarized in Fig. 5. Please note that "Dmnl" is the short term of "dimensionless".

Step 7 and 8: will be done in the near future. Based on the scenario simulation results (Fig. 5), Scenarios 2 and 3 will be further checked with investment/cost data to calculate whether the investment on TBM or CBM can break even within an acceptable period (e.g., two years).

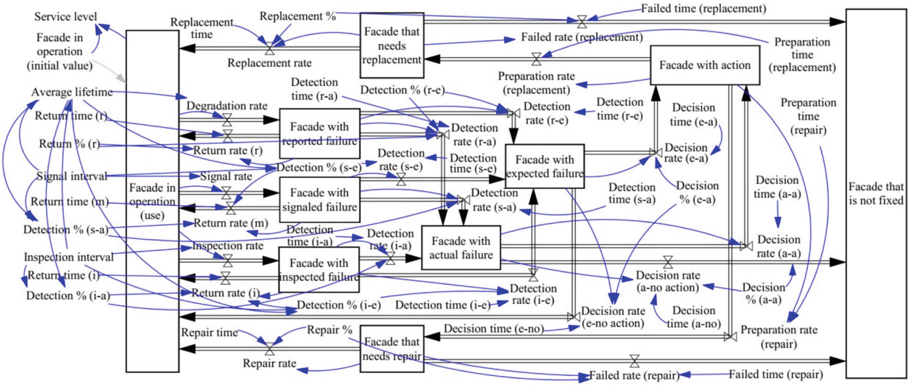


Fig. 4. System dynamics model for facade maintenance

The above reported case study is a part of the PerpetuAL project, which is a feasibility study for closing the circular material loop of aluminum facades. It was decided, within the project, to use BPMN for the AS-IS analysis of the existing processes, as well as for the design of the TO-BE optimized processes. The fact that BPMN enables easy process automation played an important role in this decision. Another factor that influenced this decision was the possibility to use BPMN models for process simulation. However, the standard BPMN tools do not allow us to append the models with user defined process parameters. In such a context, SD is more suited for process simulation. As reported in [8], it can be a valuable tool for greening the company's processes, such as the modeling of facade wear, maintenance and repair, as also illustrated in the case study above. Predictive maintenance schemes (i.e., TBM and CBM) could replace traditional maintenance schedules (i.e., CM), anticipating the presence of a failure based on different data. This information can be used to feed continuous simulation models to extend the facade service life [6].

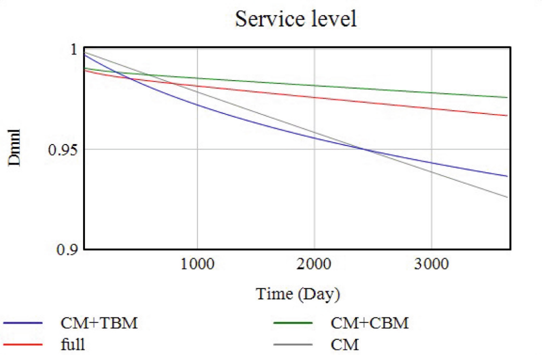


Fig. 5. Scenario simulation results

6 Conclusion and Next Steps

In this paper, we reported on initial steps towards a more explicit combination of business process modeling and system dynamics. We provided a step-by-step guide on how to use BPMN based models and system dynamics models together to model and analyze complex business processes, while illustrating this in terms of a case study on the maintenance of building facades.

As a next step, we aim to (1) further elaborate and validate the mapping as provided in Table 1, and the associated modeling guidelines, (2) develop experimental tool support to provide automatic support for this mapping and associated modeling activities, (3) develop standardized patterns to include specific facets such as energy consumption, CO2 emissions, etc., in the SFDs, (4) use these in further case studies, e.g. in relation to the PerpetuAL project and other circular economy efforts.

References

1. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19345-3
2. Caballero-Villalobos, J.P., Mejía-Delgadillo, G.E., García-Cáceres, R.G.: Scheduling of complex manufacturing systems with petri nets and genetic algorithms: a case on plastic injection moulds. Int. J. Adv. Manuf. Technol. **69**(9–12), 2773–2786 (2013)
3. Fayoumi, A., Loucopoulos, P., Fayyoumi, A.: Hybrid enterprise modelling: integrating modelling mechanisms for socio-technical systems analysis and design. J. Softw. Eng. Appl. **7**(1), 6–13 (2014)
4. Freitas, A.P., Pereira, J.L.M.: Process simulation support in BPM tools: the case of BPMN. In: 5th International Conference on Business Sustainability, Povoa de Varzim, Portugal, pp. 1–9 (2015)
5. Freund, J., Rücker, B.: Real life BPMN. Camunda (2012)
6. Giovanardi, M., Konstantinou, T., Pollo, R., Klein, T.: Internet of Things for building façade traceability: a theoretical framework to enable circular economy through life-cycle information flows. J. Clean. Prod. **382**, 135261 (2023)
7. Gopalakrishnan, S., Krogstie, J., Sindre, G.: Adapted UML activity diagrams for mobile work processes: experimental comparison of colour and pattern fills. In: Halpin, T., et al. (eds.) BPMDS/EMMSAD -2011. LNBIP, vol. 81, pp. 314–331. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21759-3_23
8. Harmon, P.: System Dynamics and Business Processes (2008). https://www.bptrends.com/bpt/wp-content/publicationfiles/advisor20080909.pdf
9. Iacob, M.E., Jonkers, H., Lankhorst, M.M., Proper, H.A., Quartel, D.A.C.: ArchiMate 2.1 Specification. The Open Group (2013)
10. Jnitova, V., Joiner, K., Efatmaneshnik, M., Chang, E.: Modelling workforce employability pipelines for organisational resilience. Int. J. Eng. Bus. Manag. **13**, 1–19 (2021)
11. Kožusznik, J., Štolfa, S., Ježek, D., Kuchař, Š: Comparison of system dynamics and BPM for software process simulation. In: Snasel, V., Platos, J., El-Qawasmeh, E. (eds.) ICDIPC 2011. CCIS, vol. 189, pp. 1–15. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22410-2_1

12. Lankhorst, M.M., et al.: Enterprise Architecture at Work - Modelling, Communication and Analysis. The Enterprise Engineering Series, 3rd edn. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29651-2
13. Messi, L., Spegni, F., Carbonari, A., Ridolfi, L., Vaccarini, M.: Process-based simulation models using BPMN for construction management at runtime. In: Proceedings of the Conference CIB W78 2021, Luxembourg, pp. 519–528 (2021)
14. OMG: Business Process Modeling Notation, V2.0. Technical Report. OMG Document Number: formal/2011-01-03, Object Management Group (2011)
15. Pereira, J.L., Freitas, A.P.: Simulation of BPMN process models: current BPM tools capabilities. In: New Advances in Information Systems and Technologies. AISC, vol. 444, pp. 557–566. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31232-3_52
16. Pourbafrani, M., van der Aalst, W.M.P.: Hybrid business process simulation: updating detailed process simulation models using high-level simulations. In: Research Challenges in Information Science, pp. 177–194. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05760-1_11
17. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Scenario-based prediction of business processes using system dynamics. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) OTM 2019. LNCS, vol. 11877, pp. 422–439. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33246-4_27
18. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Supporting automatic system dynamics model generation for simulation in the context of process mining. In: Abramowicz, W., Klein, G. (eds.) BIS 2020. LNBIP, vol. 389, pp. 249–263. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53337-3_19
19. Proper, H.A.E., Hoppenbrouwers, S.J.B.A., van der Weide, T.P.: A fact-oriented approach to activity modeling. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2005. LNCS, vol. 3762, pp. 666–675. Springer, Heidelberg (2005). https://doi.org/10.1007/11575863_86
20. Sterman, J.D.: Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill, New York (2000)
21. Tulinayo, F.P., van Bommel, P., Proper, H.A.: Enhancing the system dynamics modeling process with a domain modeling method. Int. J. Cooper. Inf. Syst. **22**(2), 1350011 (2013)
22. Tulinayo, P., Hoppenbrouwers, S.J.B.A., van Bommel, P., Proper, H.A.: Integrating system dynamics with object-role modeling and petri nets. In: Enterprise Modelling and Information Systems Architectures: Proceedings of the 3rd International Workshop on Enterprise Modelling and Information Systems Architectures; Ulm, Germany, 10–11 September 2009. LNI, vol. 152, pp. 41–54. GI (2009)
23. Wieringa, R.J.: Statistical difference-making experiments. In: Design Science Methodology for Information Systems and Software Engineering, pp. 295–317. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43839-8_20

Author Index

A
Abb, Luka 18, 453
Adams, Jan Niklas 496
Alman, Anti 289
Alpers, Sascha 161

B
Baldoni, Matteo 5
Baroglio, Cristina 5
Barthe-Delanoë, Anne-Marie 205
Benaben, Frederick 205
Benkert, Kay 440
Berti, Alessandro 427
Beyel, Harry H. 92
Bork, Dominik 205
Brennig, Katharina 440
Bresolin, Davide 383
Brzychczy, Edyta 302
Burattin, Andrea 118
Burlton, Roger 539

C
Calegari, Daniel 508
Chatain, Thomas 186
Corea, Carl 326, 396
Cremerius, Jonas 149

D
Dallavale de Pádua, Silvia Inês 539
de Leoni, Massimiliano 383, 521
De Weerdt, Jochen 245
Delfmann, Patrick 326
Delgado, Andrea 508
Di Federico, Gemma 118
Diaz, Juan F. 314
Dunzer, Sebastian 217

E
Elsayed, Nourhan 453

F
Fani Sani, Mohammadreza 369
Fettke, Peter 18
Franceschetti, Marco 131

G
Galanti, Riccardo 521
Gallik, Florian 105
Gdowska, Katarzyna 302
Gielingh, W. 565
Goni, Joklan Imelda Camelia 229
Goyal, Saurabh 70
Grant, John 396
Groher, Wolfgang 341
Grohs, Michael 453
Gusikhin, Oleg 92

H
Höchstädter, Nico 217

I
Isahagian, Vatche 44, 70

K
Kabierski, Martin 369
Kampik, Timotheus 326
Ketykó, István 31
Kilic, Emre 353
Kirikkayis, Yusuf 105
Kluza, Krzysztof 302
Kuhlmann, Isabelle 396
Kumar, Ritesh 70
Kutzner, Kristin 466

L
Laue, Ralf 466
Läuter, Martin 466
Lim, Jungeun 174
Lima Aredes, Emerson 539
Löhr, Bernd 440

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2024
J. De Weerdt and L. Pufahl (Eds.): BPM 2023 Workshops, LNBIP 492, pp. 577–578, 2024.
https://doi.org/10.1007/978-3-031-50974-2

Lomazova, Irina A. 483
López, Hugo A. 314
Lourenção, Marina 539
Lu, Xixi 57

M
Macé-Ramète, Guillaume 205
Maggi, Fabrizio Maria 289
Majidian Eidgahi, Mehran 205
Makke, Omar 92
Mannhardt, Felix 31
Marengo, Elisa 5
Matzner, Martin 217
Meroni, Giovanni 118
Micalizio, Roberto 5
Mitsyuk, Alexey A. 483
Montali, Marco 289, 409
Müller, Oliver 440
Muthusamy, Vinod 44

N
Namaki Araghi, Sina 205
Narcomey, Austin 44

O
Ouyang, Chun 259

P
Patzlaff, Hendrik 149
Pfeiffer, Peter 18
Proper, H. A. 565

Q
Quesada, Luis 314

R
Rabe, Markus 353
Ravesteijn, J. P. P. 565
Rehse, Jana-Rebecca 18, 453
Reichert, Manfred 105
Reijers, Hajo 539
Rino, Neha 186
Riss, Uwe V. 341
Rivkin, Andrey 289, 483
Rizk, Yara 44
Rosero, Juan C. 314

S
Schüler, Selina 161
Schuster, Daniel 427
Seiger, Ronny 105, 131
Sepúlveda, Marcos 272
Sliż, Piotr 552
Song, Minseok 174
Syed, Rehan 259

T
Tang, Willi 217

V
van der Aalst, Wil M. P. 87, 92, 369, 427, 496
Van Looy, Amy 229
van Luijken, Mathieu 31
van Zelst, Sebastiaan J. 369
Velásquez, Ignacio 272
Venkateswaran, Praveen 44
Verbeke, Wouter 245
Verhoef, Bart J. 57
Verma, Ashish 70

W
Wang, Zeping 259
Weber, Barbara 131
Weske, Mathias 149
Weytjens, Hans 245
Winkler, Sarah 409

Z
Zavatteri, Matteo 383
Zhu, Q. 565
Zilker, Sandra 217