Chiara Di Francescomarino
Andrea Burattin
Christian Janiesch
Shazia Sadiq (Eds.)

LNBIP 490

Business Process Management Forum

BPM 2023 Forum
Utrecht, The Netherlands, September 11–15, 2023
Proceedings

Springer

Lecture Notes in Business Information Processing 490

Series Editors

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.

Chiara Di Francescomarino · Andrea Burattin ·
Christian Janiesch · Shazia Sadiq
Editors

Business Process Management Forum

BPM 2023 Forum
Utrecht, The Netherlands, September 11–15, 2023
Proceedings

Springer

Editors
Chiara Di Francescomarino
University of Trento
Trento, Italy

Andrea Burattin
Technical University of Denmark
Kgs. Lyngby, Denmark

Christian Janiesch
TU Dortmund University
Dortmund, Germany

Shazia Sadiq
The University of Queensland
Brisbane, QLD, Australia

ISSN 1865-1348 ISSN 1865-1356 (electronic)
Lecture Notes in Business Information Processing
ISBN 978-3-031-41622-4 ISBN 978-3-031-41623-1 (eBook)
https://doi.org/10.1007/978-3-031-41623-1

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

Preface

This volume comprises all papers presented at the BPM Forum of the 21st International Conference on Business Process Management (BPM 2023), held during September 11–15, 2023 in Utrecht, the Netherlands. Similarly to previous years, the BPM Forum hosted innovative research contributions characterized by their high potential to stimulate interesting discussion and scientific debate, although not yet reaching the rigorous technical quality criteria required to be included in the main conference proceedings. In this sense, the BPM Forum papers characterize themselves by novel ideas about emergent BPM topics.

This year, the conference received a total of 167 submissions, out of which 151 entered the review phase. The review process for each paper involved single-blind reviews by at least three Program Committee members and one Senior Program Committee member and a subsequent discussion that culminated into a summarizing meta-review with recommendation. In the end, 27 papers were accepted at the main conference, and 23 papers were included in the BPM Forum (the latter being compiled in this volume).

BPM 2022 in Münster, Germany marked the cautious and successful return to a full in-person conference. In light of BPM 2023's submission and attendance numbers, the appreciation and importance of BPM as a physical venue for the scientific community stands unquestioned. The conference was flanked by a multitude of events, such as the Blockchain, Educators, and RPA Fora, 11 workshops, tutorials, a doctoral consortium, and wonderful social events, which gave rise to the opportunity for networking and exchanging the latest research ideas.

We would like to thank all authors, both regular and senior members of the Program Committees, and the external reviewers of the three tracks, foundations, engineering, and management. They made a rigorous, extensive, and timely review procedure possible and thus enabled the high-quality research output reflected by the papers in both the main conference and BPM Forum proceedings. Further, we acknowledge our sponsors for their support in making BPM 2023 happen: Celonis and Software AG as platinum sponsors; BPM Consult as bronze sponsor; and Hogeschool Utrecht, the Netherlands Research School for Information and Knowledge Systems, Springer, and Utrecht University as academic sponsors.

Finally, we would like to express our gratitude to Hajo Reijers as the General Chair of BPM 2023, together with the Organizing Committee Chairs Inge van de Weerd, Jan Martijn van der Werf, and Pascal Ravesteijn, and their staff. The Utrecht team did an impeccable job in planning and organizing an unforgettable conference.

September 2023

Chiara Di Francescomarino
Andrea Burattin
Christian Janiesch
Shazia Sadiq

Organization

Steering Committee

Jan Mendling (Chair)	Humboldt-Universität zu Berlin, Germany
Marlon Dumas	University of Tartu, Estonia
Avigdor Gal	Technion – Israel Institute of Technology, Israel
Chiara Ghidini	Fondazione Bruno Kessler, Italy
Manfred Reichert	University of Ulm, Germany
Hajo A. Reijers	University of Utrecht, The Netherlands
Stefanie Rinderle-Ma	Technical University of Munich, Germany
Adela del Río Ortega	University of Seville, Spain
Michael Rosemann	Queensland University of Technology, Australia
Shazia Sadiq	The University of Queensland, Australia
Barbara Weber	University of St. Gallen, Switzerland
Matthias Weidlich	Humboldt-Universität zu Berlin, Germany
Mathias Weske	University of Potsdam, Germany

Executive Committee

General Chair

Hajo A. Reijers	University of Utrecht, The Netherlands

Main Conference Program Committee Chairs

Chiara Di Francescomarino (Track I Chair)	University of Trento, Italy
Andrea Burattin (Track II Chair)	Technical University of Denmark, Denmark
Christian Janiesch (Track III Chair)	TU Dortmund University, Germany
Shazia Sadiq (Consolidation Chair)	The University of Queensland, Australia

Workshop Chairs

Jochen De Weerdt	KU Leuven, Belgium
Luise Pufahl	Technical University of Munich, Germany

Demonstration and Resources Chairs

Tijs Slaats	University of Copenhagen, Denmark
Andrés Jiménez Ramírez	University of Seville, Spain
Karolin Winter	Eindhoven University of Technology, The Netherlands

Industry Day Chairs

Arjen Maris	University of Twente, The Netherlands
Willemijn van Haeften	HU University of Applied Sciences Utrecht, The Netherlands

Blockchain Forum Chairs

Julius Köpke	University of Klagenfurt, Austria
Orlenys López Pintado	University of Tartu, Estonia

RPA Forum Chairs

Ralf Plattfaut	University of Duisburg-Essen, Germany
Jana-Rebecca Rehse	University of Mannheim, Germany

Educators Forum Chairs

Jorge Munoz-Gama	Pontificia Universidad Católica de Chile, Chile
Katarzyna Gdowska	AGH University of Science and Technology, Poland
Fernanda Gonzalez-Lopez†	Pontificia Universidad Católica de Chile/Human and Process Data Science Research Lab, Chile
Koen Smit	HU University of Applied Sciences Utrecht, The Netherlands
Jan Martijn van der Werf	Utrecht University, The Netherlands

Tutorial Chairs

Michael Rosemann	Queensland University of Technology, Australia
Sietse Overbeek	Utrecht University, The Netherlands

Journal-First Track Chairs

Mieke Jans	Hasselt University, Belgium
Henrik Leopold	Kühne Logistics University, Germany

Doctoral Consortium Chairs

Dirk Fahland	Eindhoven University of Technology, The Netherlands
Barbara Weber	University of St. Gallen, Switzerland
Johan Versendaal	HU University of Applied Sciences Utrecht, The Netherlands

Best BPM Dissertation Award Chair

Jan Mendling	Humboldt-Universität zu Berlin, Germany

Publicity Chairs

Iris Beerepoot	Utrecht University, The Netherlands
Andrea Delgado	Universidad de la República, Uruguay
Mahendrawathi ER	Institut Teknologi Sepuluh Nopember, Indonesia

Diversity and Inclusion Chairs

Shazia Sadiq	The University of Queensland, Australia
Jens Gulden	Utrecht University, The Netherlands
Adela del Río Ortega	University of Seville, Spain

Proceedings Chairs

Xixi Lu	Utrecht University, The Netherlands
Felix Mannhardt	Eindhoven University of Technology, The Netherlands

Organizing Committee Chairs

Inge van de Weerd	Utrecht University, The Netherlands
Jan Martijn van der Werf	Utrecht University, The Netherlands
Pascal Ravesteijn	HU University of Applied Sciences Utrecht, The Netherlands

Track I: Foundations

Senior Program Committee

Wil van der Aalst	RWTH Aachen University, Germany
Jörg Desel	FernUniversität in Hagen, Germany
Claudio Di Ciccio	Sapienza University of Rome, Italy
Dirk Fahland	Eindhoven University of Technology, The Netherlands
Chiara Ghidini	Fondazione Bruno Kessler, Italy
Thomas Hildebrandt	University of Copenhagen, Denmark
Arthur ter Hofstede	Queensland University of Technology, Australia
Rick Hull	University of California, USA
Sander J. J. Leemans	RWTH Aachen University, Germany
Fabrizio Maria Maggi	Free University of Bozen-Bolzano, Italy
Andrea Marrella	Sapienza University of Rome, Italy
Marco Montali	Free University of Bozen-Bolzano, Italy
Oscar Pastor	Universidad Politécnica de Valencia, Spain
Artem Polyvyanyy	The University of Melbourne, Australia
Manfred Reichert	University of Ulm, Germany
Hagen Voelzer	University of St. Gallen, Switzerland
Matthias Weidlich	Humboldt-Universität zu Berlin, Germany
Jan Martijn van der Werf	Utrecht University, The Netherlands
Mathias Weske	Hasso Plattner Institute/University of Potsdam, Germany

Program Committee

Lars Ackermann	University of Bayreuth, Germany
Ahmed Awad	University of Tartu, Estonia
Patrick Delfmann	Universität Koblenz, Germany
Rik Eshuis	Eindhoven University of Technology, The Netherlands

Peter Fettke	German Research Center for Artificial Intelligence (DFKI)/Saarland University, Germany
Valeria Fionda	University of Calabria, Italy
Francesco Folino	ICAR-CNR, Italy
Maria Teresa Gómez López	University of Seville, Spain
Alessandro Gianola	Free University of Bozen-Bolzano, Italy
Guido Governatori	Independent Researcher, Australia
Giancarlo Guizzardi	University of Twente, The Netherlands
Ekkart Kindler	Technical University of Denmark, Denmark
Akhil Kumar	The Pennsylvania State University, USA
Irina Lomazova	National Research University Higher School of Economics, Russia
Hugo Andrés López	Technical University of Denmark, Denmark
Qinghua Lu	CSIRO, Australia
Xixi Lu	Utrecht University, The Netherlands
Felix Mannhardt	Eindhoven University of Technology, The Netherlands
Werner Nutt	Free University of Bozen-Bolzano, Italy
Chun Ouyang	Queensland University of Technology, Australia
Luigi Pontieri	ICAR-CNR, Italy
Daniel Ritter	SAP SE, Germany
Andrey Rivkin	Technical University of Denmark, Denmark
Arik Senderovich	York University, Canada
Tijs Slaats	University of Copenhagen, Denmark
Monique Snoeck	KU Leuven, Belgium
Ernest Teniente	Universitat Politècnica de Catalunya, Spain
Eric Verbeek	Eindhoven University of Technology, The Netherlands
Karsten Wolf	Universität Rostock, Germany
Francesca Zerbato	University of St. Gallen, Switzerland

Track II: Engineering

Senior Program Committee

Boualem Benatallah	University of New South Wales, Australia
Jochen De Weerdt	KU Leuven, Belgium
Remco Dijkman	Eindhoven University of Technology, The Netherlands

Boudewijn van Dongen	Eindhoven University of Technology, The Netherlands
Marlon Dumas	University of Tartu, Estonia
Avigdor Gal	Technion – Israel Institute of Technology, Israel
Massimo Mecella	Sapienza University of Rome, Italy
Jorge Munoz-Gama	Pontificia Universidad Católica de Chile, Chile
Luise Pufahl	Technical University of Munich, Germany
Hajo A. Reijers	Utrecht University, The Netherlands
Stefanie Rinderle-Ma	Technical University of Munich, Germany
Pnina Soffer	University of Haifa, Israel
Barbara Weber	University of St. Gallen, Switzerland
Ingo Weber	Technical University of Munich, Germany
Moe Thandar Wynn	Queensland University of Technology, Australia

Program Committee

Han van der Aa	University of Mannheim, Germany
Robert Andrews	Queensland University of Technology, Australia
Abel Armas Cervantes	The University of Melbourne, Australia
Nick van Beest	CSIRO, Australia
Cristina Cabanillas	University of Seville, Spain
Fabio Casati	ServiceNow, USA/University of Trento, Italy
Johannes De Smedt	KU Leuven, Belgium
Benoît Depaire	Hasselt University, Belgium
Joerg Evermann	Memorial University of Newfoundland, Canada
Walid Gaaloul	Télécom SudParis, France
Luciano García-Bañuelos	Tecnológico de Monterrey, Mexico
Laura Genga	Eindhoven University of Technology, The Netherlands
Daniela Grigori	Université Paris-Dauphine - PSL, France
Georg Grossmann	University of South Australia, Australia
Mieke Jans	Hasselt University, Belgium
Anna Kalenkova	The University of Adelaide, Australia
Dimka Karastoyanova	University of Groningen, The Netherlands
Agnes Koschmider	University of Bayreuth, Germany
Massimiliano de Leoni	University of Padua, Italy
Henrik Leopold	Kühne Logistics University, Germany
Francesco Leotta	Sapienza University of Rome, Italy
Elisa Marengo	Free University of Bozen-Bolzano, Italy
Rabeb Mizouni	Khalifa University, United Arab Emirates
Timo Nolle	Technical University of Darmstadt, Germany
Hye-Young Paik	University of New South Wales, Australia

Cesare Pautasso	University of Lugano, Switzerland
Pierluigi Plebani	Politecnico di Milano, Italy
Pascal Poizat	Université Paris Nanterre/LIP6, France
Simon Poon	The University of Sydney, Australia
Barbara Re	University of Camerino, Italy
Manuel Resinas	University of Seville, Spain
Stefan Schönig	University of Regensburg, Germany
Marcos Sepúlveda	Pontificia Universidad Católica de Chile, Chile
Natalia Sidorova	Eindhoven University of Technology, The Netherlands
Renuka Sindhgatta	IBM Research, India
Minseok Song	Pohang University of Science and Technology, South Korea
Niek Tax	Meta, UK
Seppe vanden Broucke	Ghent University, Belgium
Karolin Winter	Eindhoven University of Technology, The Netherlands
Nicola Zannone	Eindhoven University of Technology, The Netherlands
Sebastiaan van Zelst	RWTH Aachen University/Fraunhofer Institute for Applied Information Technology FIT, Germany

Track III: Management

Senior Program Committee

Daniel Beverungen	Paderborn University, Germany
Adela del Río Ortega	University of Seville, Spain
Paul Grefen	Eindhoven University of Technology, The Netherlands
Mojca Indihar Štemberger	University of Ljubljana, Slovenia
Marta Indulska	The University of Queensland, Australia
Peter Loos	German Research Center for Artificial Intelligence (DFKI)/Saarland University, Germany
Jan Mendling	Humboldt-Universität zu Berlin, Germany
Jan Recker	University of Hamburg, Germany
Maximilian Röglinger	FIM Research Center Finance & Information Management/University of Bayreuth, Germany
Michael Rosemann	Queensland University of Technology, Australia

Flavia Santoro	Universidade do Estado do Rio de Janeiro, Brazil
Peter Trkman	University of Ljubljana, Slovenia
Amy Van Looy	Ghent University, Belgium
Jan vom Brocke	University of Münster, Germany / University of Liechtenstein, Liechtenstein

Program Committee

Amine Abbad-Andaloussi	University of St. Gallen, Switzerland
Banu Aysolmaz	Eindhoven University of Technology, The Netherlands
Christian Bartelheiner	Paderborn University, Germany
Iris Beerepoot	Utrecht University, The Netherlands
Marco Comuzzi	Ulsan National Institute of Science and Technology, South Korea
Irene Bedilia Estrada Torres	University of Seville, Spain
Renata Gabryelczyk	University of Warsaw, Poland
Kanika Goel	Queensland University of Technology, Australia
Fernanda Gonzalez-Lopez†	Pontificia Universidad Católica de Chile/Human and Process Data Science Research Lab, Chile
Thomas Grisold	University of St. Gallen, Switzerland
Tomislav Hernaus	University of Zagreb, Croatia
John Krogstie	Norwegian University of Science and Technology, Norway
Michael Leyer	University of Marburg, Germany/Queensland University of Technology, Australia
Alexander Mädche	Karlsruhe Institute of Technology, Germany
Niels Martin	Hasselt University, Belgium
Martin Matzner	Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Alexander Nolte	University of Tartu, Estonia
Nadine Ostern	Queensland University of Technology, Australia
Ralf Plattfaut	University of Duisburg-Essen, Germany
Geert Poels	Ghent University, Belgium
Gregor Polančič	University of Maribor, Slovenia
Pascal Ravesteyn	Utrecht University of Applied Sciences, The Netherlands
Jana-Rebecca Rehse	University of Mannheim, Germany
Kate Revoredo	Vienna University of Economics and Business, Austria
Dennis M. Riehle	Universität Koblenz, Germany
Estefanía Serral	KU Leuven, Belgium

Rehan Syed	Queensland University of Technology, Australia
Oktay Turetken	Eindhoven University of Technology, The Netherlands
Irene Vanderfeesten	KU Leuven, The Netherlands
Inge van de Weerd	Utrecht University, The Netherlands
Sven Weinzierl	Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Axel Winkelmann	University of Würzburg, Germany
Bastian Wurm	Ludwig-Maximilians-Universität Munich, Germany
Patrick Zschech	Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Michael zur Mühlen	Stevens Institute of Technology, USA

Additional Reviewers

Nour Assy
Charlotte Bahr
Luca Barbaro
Marisol Barrientos
Janik-Vasily Benzin
Jonas Blatt
Lasse Bohlen
Katharina Brennig
Qifan Chen
Minsu Cho
Axel Christfort
Carl Corea
Paul Cosma
Sebastian Dunzer
Peter A. François
Yuhong Fu
Alexandre Goossens
Valerio Goretti
Heerko Groefsema
Michael Grohs
Jonas Gunklach
Nico Hambauer
Tiphaine Henry
Mubashar Iqbal
Adrian Joas
Merlin Knaeble
Sven Kruschel
Annina Liessmann
Yue Liu
Yang Lu
Bernd Löhr
Edoardo Marangone
Alessandro Marcelletti
Rahma Mukta
Roman Nesterov
Xavier Oriol
Philippe Queinnec
Luis Quesada
Kristo Raun
Julian Rosenberger
Gwen Salaün
Daniel Schloß
Julia Seitz
Nafiseh Soveizi
Klara Steflova
Vinicius Dani Stein
Willi Tang
Niek Tax
Wouter van der Waal
Ignacio Velásquez
Charlotte Verbruggen
Jia Wei
Bemali Wickramanayake
Boming Xia
Arash Yadegari Ghahderijani
Sandra Zilker

Contents

Foundations

Trusted Compliance Checking on Blockchain with Commitments: A Model-Driven Approach

Marcello Bertolini[1], Giovanni Meroni[2(✉)], and Pierluigi Plebani[1]

[1] Politecnico di Milano Piazza Leonardo da Vinci, 32 - 20133 Milan, Italy
marcello.bertolini@mail.polimi.it, pierluigi.plebani@polimi.it

[2] Technical University of Denmark, Richard Petersen Plads, 2800 Kgs, Lyngby, Denmark
giom@dtu.dk

Abstract. Blockchain and smart contracts are promising technologies to perform trusted compliance checking. By formalizing compliance rules with smart contract code and collecting information required to assess them on-chain, anyone can verify if a compliance violation occurred. To this aim, tools and techniques to execute business processes on-chain have been proposed. However, such techniques require the activities and the process data internal to an organization to be fully disclosed with all the participants. This may not be desirable for confidentiality reasons, and may also lead to high operational costs.

This paper proposes a model-driven approach that uses a choreography diagram annotated with commitments to model compliance rules and to identify the message exchanges that are relevant for enforcing business agreements. The resulting diagram is used to generate the smart contract code required to perform compliance checking, limiting the information stored in the blockchain to the one strictly needed to evaluate the compliance rules.

Keywords: Runtime compliance checking · Commitments · Blockchain · Smart contracts · Choreography diagrams

1 Introduction

Originally designed for cryptocurrency exchanges, blockchain technology has found its application in several different domains where untrusted parties need to cooperate. In particular, by providing immutability and persistence of the information being stored, the blockchain is often adopted as a tamper-proof registry for documents and other digital assets. In addition, second-generation blockchains introduced support for smart contracts, agreements between participants that can be formalized as executable code and thus be automatically enforced.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 3–19, 2023.
https://doi.org/10.1007/978-3-031-41623-1_1

Thank to these capabilities, the blockchain is seen as a promising technology to perform trusted process monitoring, and to identify violations in the agreements without the need of trusted third-parties. To this aim, several solutions that transform process models into executable smart contracts exist [10,17]. Nevertheless, designing a solution for process monitoring with a blockchain still presents some challenges [4]. In particular, the cost for storing information on a blockchain can be quite high. Also, to protect their know-how, organizations may not be willing to disclose their entire internal processes and data with other participants. Instead of monitoring the process as a whole, one may want to rely on the information exchanged among participants, which serves to coordinate their internal processes. In particular, compliance rules that predicate on this information could be defined to ensure that the agreements between participants are fulfilled.

To this aim, we propose a model-driven approach to identify the information that should be on-chain, and to produce the smart contract code required to monitor the agreements. To model compliance rules, we adopt the extension of BPMN choreography diagrams with commitments presented in [14]. The resulting code contains the business logic to store and retrieve information relevant for the compliance rules. Also, it contains the business logic to trigger the evaluation of the rules, as well as to detect anomalies in the information exchanges.

According to this goal, we identified the need to reduce the information on the blockchain to the one strictly needed to verify the commitments [4], which lead to the following research question: *RQ1: How can the information needed to evaluate commitments be identified and made available on a blockchain?* We also identified the need to assist developers in implementing smart contract code [19], which lead to the following research question: *RQ2: Can we simplify the implementation and deployment of smart contract code to monitor commitments?* To address these research questions, we developed the following artefacts:

– A model-driven approach to perform compliance checking with commitments on a blockchain. This artefact aims at addressing RQ1 and RQ2.
– A set of rules to identify the information to be stored on-chain from choreography diagrams annotated with commitments. This artefact aims at addressing RQ1.
– An application-independent smart contract code skeleton that implements the majority of the logic to enforce commitments. This artefact aims at addressing RQ2.
– A set of steps to deploy smart contract code implementing commitments. This artefact aims at addressing RQ2.

To validate these artefacts, we tested them against a case study coming from the logistics domain.

The rest of this paper is organized as follows. To make this paper self-contained, Sect. 2 provides an overview on commitments. Section 3 discusses the model-driven approach. Section 4 evaluates our approach against a real-world process. Section 5 discusses related work and their limitations. Finally, Sect. 6 concludes this paper and outlines possible future work.

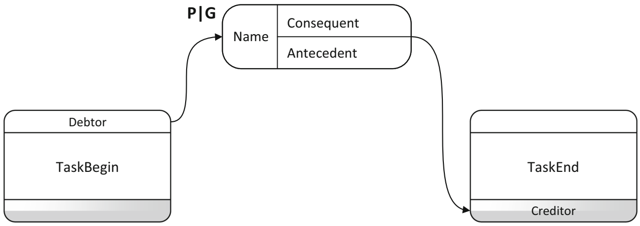


Fig. 1. Graphical representation of a timed-commitment, and its connection to choreography tasks.

2 Baseline

Social commitments have been introduced in the computer science literature to manage conditions under which interactions among parties should occur [15,16,18]. In this paper, we consider a variant of social commitments, i.e., timed-commitments, which take care also of the time flowing [14]. A timed-commitment is graphically represented as in Fig. 1 and it is used to express conditions to be respected while two participants interact in a choreographed process[1]. It is assumed that the BPMN choreography notation is used to model the interaction protocol among the participants. More in details, a timed-commitment is defined by:

– *Name*: a unique name of the commitment for a given process model. We assume that an instance of the commitment is created for each case of the process model. The union of the commitment name and the case id (that could be identified by the correlation set) represents the unique identifier of a commitment instance.
– *Debtor* and *creditor*: the two parties involved in the commitment, i.e., the party offering a service under a given condition and the party that is taking advantage of the service. In our context, the condition concerns the way in which the two parties are interacting: e.g., the time required for a response to a request, or the status of the resources that are exchanged.
– *Scope*: the period in which the commitment must be evaluated.
– *Antecedent* and *consequent*: two boolean logic expressions defining under which conditions the service must be provided and consumed. Informally speaking, when the antecedent becomes true, then the commitment starts being considered and the value of the consequent is evaluated.
– *Type*: a letter, either G or P, placed at the top-left indicates the time of validity of the commitment, i.e., when the consequent is evaluated. A *(G)oal* commitment implies the evaluation of the consequent to happen when the activity to which the commitment is attached ends. A *(P)ersisting* implies a continuous evaluation from when the antecendent becomes true until the end of the activity to which the commitment is attached.

[1] For a formal definition we refer the reader to [14].

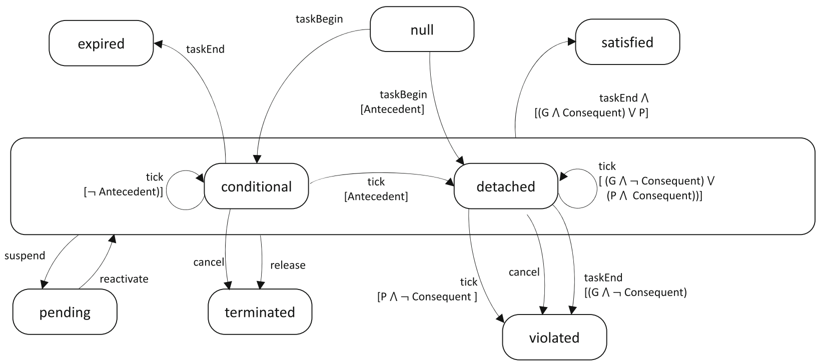


Fig. 2. Timed-commitment machine

While the name, the antecedent, the consequent, and the type are part of the timed-commitment construct, the debtor, the creditor, and the scope are derived from how the timed-commitment is attached to the choreography model. In particular, the debtor is defined as the participant indicated in the band from which a connector starts towards the commitment. Similarly, the creditor is defined as the participant indicated in the band that reaches a connector leaving from the timed-commitment. Finally, the scope of the commitment (i) starts when the interaction associated to the task defining the debtor occurs; the scope (ii) ends when the interaction associated to the task defining the creditor occurs.

The semantics of the timed-commitment model is expressed by the commitment machine shown in Fig. 2, where a specific *tick* event is introduced to model the time flowing. This event may be internally generated, or communicated from the external environment, based on who is aware of the flow of time. The frequence of the tick also defines how often the conditions are checked, thus the accuracy of the monitoring process. In addition, the events *taskBegin* and *taskEnd* will indicate the events that define the boundary of the commitment scope associated to the choreography tasks which the commitment is connected with.

As reported in the machine, a commitment is initially in a *null* state which indicates a non-instantiated commitment. We assume that a commitment is instantiated, and then goes to a *conditional* state, when the event *taskBegin* occur. This makes a commitment instance linked to a specific choreography instance. While in the conditional state:

– The antecedent is evaluated every tick and when true the commitment moves to a *detached* state. In case the antecendent is already true during the instantiation of the commitment, it goes directly to the detached state.
– A request for cancellation moves the commitment to a *terminated* state.
– If the end of the scope is reached, the status of the commitment is set to *expired*.

While the commitment is in a detached state, for every *tick* during the scope, i.e., before the *taskEnd* occurs, the consequent is evaluated and:

– In case of a persisting commitment, the consequent must be true. Otherwise, the commitment is considered as *violated* as the persisting condition is no longer holding before the end of the scope is reached.
– In case of a goal commitment, the commitment remains in the detached state as long as the consequent is false. Otherwise, the commitment is considered as *satisfied* as the goal is reached.

A dual situation is determined at the end of the scope, i.e., when the *taskEnd* occurs. More specifically:

– In case of a persisting commitment, the commitment moves to a *satisfied* state, as the condition of the consequent has always been true during the entire scope.
– In case of goal commitment, the commitment moves to a *violated* state, as the condition of the consequent has never been satisfied during the entire scope.

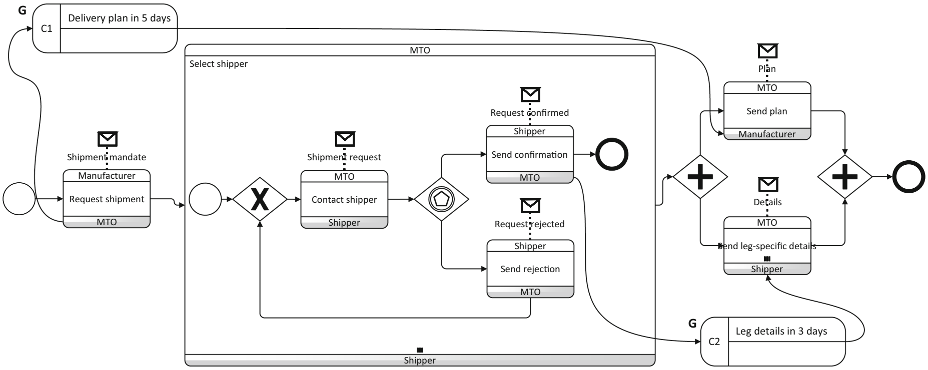
Finally, regardless of the commitment type and whether it is in conditional or detached state, a suspend event could be raised and the commitment goes to a *pending* state. In this case, a reactivate event restores the status to the one in which the commitment was when the suspend event arrived. A release event causes the *termination* of the commitment.

Figure 3 shows two examples of choreographed processes enriched with commitments. Compared to standard choreography diagrams, which can represent only the nature and the order in which the messages should be exchanged, these models also represent temporal constraints and conditions on the contents of the messages. For instance, the planning process (Fig. 3a), shows the usage of goal commitments: (i) C_1 indicates that the MTO (the debtor) must produce the plan (taskEnd) should in less than 5 days (the consequent) starting from when it receives the mandate (taskBegin) from the manufacturer (the creditor); (ii) C_2 indicates that each shipper (the debtor) involved in the end-to-end transport has to send, in 3 days (the consequent) from when the confirmation is received (taskBegin), to the MTO (the creditor) the details (taskEnd) of the specific leg of the transport to which it has been assigned. The last mile example (Fig. 3b) shows the usage of a persisting commitment where the Shipper (the debtor) has to guarantee the transport of the goods with a temperature lower than 5°C (the consequent) along the whole itinerary: from where the delivery start (taskBegin) to where the delivery terminates (taskEnd).

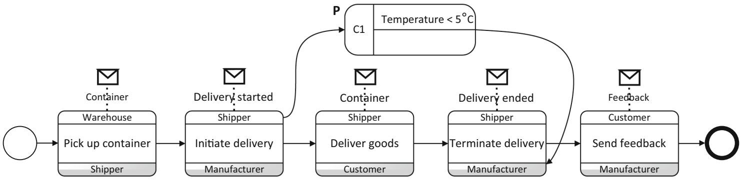
3 Approach

As discussed in the previous section, commitments are useful at design-time to model compliance rules targeting multi-party business processes. The formal semantic of commitments and the possibility to attach them to well established choreography models such as the BPMN, simplifies the design of business processes to include both functional and non-functional aspects.

When it comes to the need to monitor these commitments for checking their compliance, it is fundamental to avoid centralized solutions as in a multi-party



(a) Transport planning choreography process



(b) Last mile process.

Fig. 3. Example of commitment usage

setting imposing a common platform requires an agreement which could be not achieved [12]. At the same time, a distributed solution must ensure the interoperability of the monitoring systems adopted by the involved participants. Finally, the results of the monitoring activity must be trusted by all the participant.

To address these aspects, we envision to adopt a blockchain-based process monitoring as it is distributed by definition, it provides a communication infrastructure that can be connected with existing systems [19], and being based on a tamper-proof storage can introduce a layer of trust among the participant. Nevertheless, the adoption of a blockchain-based process monitoring is not straightforward: deciding what and how to store in the blockchain, implementing smart contracts which check the compliance of the commitments requires a significant effort. To this aim, we propose a model-driven approach that, starting from choreography diagrams annotated with commitments, can guide developers in the implementation of a blockchain based compliance checking platform.

The approach is organized in three sequential steps, which are shown in Fig. 4. The first step takes as input a choreography diagram annotated with commitments. It then enriches the diagram by identifying and marking the messages that contain the information required to evaluate the commitments. If no message contains the information for a commitment, the diagram is extended by adding choreography tasks deputed to sharing this information. This step can be fully automated. Details on the rules to enrich the diagram are presented in Sect. 3.1.

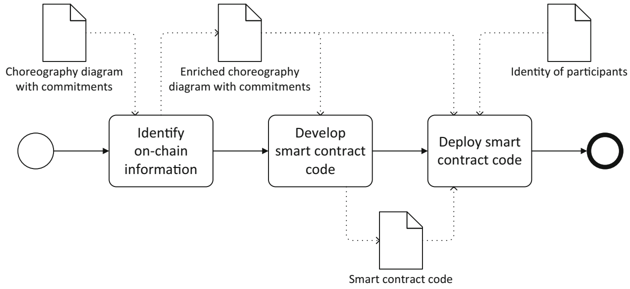


Fig. 4. Overview of the approach.

The second step takes as input the enriched choreography diagram annotated with commitments. For each commitment in the diagram, this step produces a smart contract code skeleton that models the commitment machine and handles the information required to evaluate the commitment. This step can be partially automated, since the antecedent and consequent expressions for the commitment, and the structure of the information, must be manually implemented. The architecture of the smart contract code is presented in Sect. 3.2.

The third and final step takes as input the smart contract code, the enriched choreography diagram annotated with commitments, and a list containing the identity of each participant in the process (i.e., the blockchain address being used by each of them). Based on this information, this step deploys the smart contract on the blockchain. This step can be fully automated. Details on the rules to deploy the smart contract code are presented in Sect. 3.3.

3.1 Identifying On-Chain Information

Due to their nature, smart contracts code is executed only when invoked by a blockchain transaction. Thus, they are not able to actively monitor a resource by querying its status. Instead, information such as when a commitment starts, ends and the data required to evaluate the antecedent and consequent expressions must be passed as transaction parameters. This means that activity related events – such as *taskBegin*, *taskEnd*, and *tick* events defined in Sect. 2 – are triggered by the arrival of specific messages in the choreography diagram.

This raises two questions: (i) which are the activities that are relevant for the evaluation of the commitment? (ii) how can we link the messages in the choreography diagram to the correct events, i.e. which are the messages that trigger the activity related events, *taskBegin* and *taskEnd*, of the commitment lifecycle?

In order to answer the first question, starting from the choreography diagram, it is possible to derive associations between the activities and the commitments.

- For each commitment C_i, if A is the set of Choreography tasks of the choreography diagram, $A_i^S \subseteq A$ is the subset of choreography tasks that belongs to

the scope of C_i. A task is considered in the scope of a commitment if it belongs to at least one of the possible traces of the choreography diagram between the *taskBegin* and the *taskEnd*.
- $A_i^C \subseteq A_i^S$ is the subset of choreography tasks for which either the debtor or the creditor refers to one of its participants and it is responsible for sending a message.

Therefore, it can be assumed that the messages of interest for the evaluation of a commitment C_i, are the messages exchanged in the choreography tasks that belong to A_i^C.

The second question can be solved by dividing the messages exchanged in the choreography tasks $cTask_j$ that belong to A_i^C into three categories: (i) InitScope (ii) Scope, and (iii) TerminateScope. A message belongs to the InitScope category if $cTask_j$ triggers the activity related event target starts, i.e., if $cTask_j$ has an outgoing connection pointing to C_i. Similarly, a message belongs to the TerminateScope category if $cTask_j$ triggers the activity related event target ends, i.e., if $cTask_j$ has an incoming connection coming from C_i. All the remaining messages are assigned to the Scope category and, along with the ones belonging to the previous two categories, they can be used to evaluate the commitment antecedent and consequent logic conditions. To keep track of this classification in the next steps of the approach, the following transformation rule is applied for each commitment C_i.

TR 1. For each choreography task $cTask_j \in A_i^C$, an annotation A_j is created and attached to $cTask_j$ via an association relation. The text of A_j contains the expression $\{C_i.name\}$: $\{scope\}$, where $\{C_i.name\}$ is the name of C_i and $\{scope\}$ is InitScope, Scope, or TerminateScope depending on the category of the messages exchanged by $cTask_j$.

Also, as outlined in [4], smart contracts cannot keep track of time autonomously. The only way for a smart contract to know the current time, with an average error corresponding to the average block mining time, is to access the mined block timestamp. For this reason, the commitment *tick* event should be triggered every time the smart contract is called from the outside, i.e. every time a participant sends a message to the smart contract. If a timer is required by a commitment, the smart contract must expose a public method to trigger the `tick` event, which could be invoked at any time. This method could be called for example by an oracle and/or by any participant of the business process, in order to simulate the time flowing.

After a commitment C_i is attached to the tasks of the choreography diagram, the antecedent and consequent condition will predicate only on the messages sent to the smart contract that belong to A_i^C. It is therefore necessary to include in the choreography diagram all the information that is needed by the smart contract to evaluate these two conditions. However, when a persisting commitment is present in the business process, it means that it is necessary to persistently monitor a physical or virtual object involved in the process. This object may not be always of the same type, therefore, its monitoring requirements may change from one object to another even if the process itself does not change. For example, a delivery process may be modeled only once, regardless of the number of physical objects to be delivered. Whenever a choreography diagram contains a persisting commitment, the choreography task that informs the participants about the conditions of the object is missing. For this reason, since the smart contract accepts only the messages that are present in the choreography diagram, it is necessary to generate a choreography task that is unique for each type of object and that is responsible for periodically sending the information required to evaluate the consequent of a persisting commitment. We call this choreography task a monitoring task. When enriching the choreography diagram with monitoring tasks, it is important not to alter control flow dependencies of the existing choreography tasks.

The generation method proposed wants to add for each persisting commitment C_i^P defined in the choreography diagram a monitoring task $mTask_i$ that periodically sends the status of the object. When a persisting commitment is defined, the debtor must guarantee that the consequent condition is kept true for the entire commitment scope. For this reason, the following assumptions can be made:

- The participants of $mTask_i$ are the debtor and the creditor of C_i^P.
- $mTask_i$ must start as soon as the InitScope message is sent.
- $mTask_i$ must run in parallel with the choreography tasks belonging to the commitment scope.
- The initiator of $mTask_i$ is the debtor of the commitment, whilst the receiver is the creditor.
- $mTask_i$ must terminate when the TerminateScope message is sent.

Given these assumptions, to enrich the choreography diagram with monitoring tasks, the following transformation rules are applied for each persisting commitment C_i^P:

TR 2. A standard loop choreography task $mTask_i$ is created. $mTask_i$ has the debtor as initiator, and the creditor as receiver. $mTask_i$ is named *Send data for* $\{C_i^P.name\}$, where $\{C_i^P.name\}$ is the name of C_i^P.

TR 3. A message M_i is created and attached to the initiator in $mTask_i$. M_i is named $\{C_i^P.name\}$ *data*, where $\{C_i^P.name\}$ is the name of C_i^P.

TR 4. The control flow connection from $iTask_j$ to PE_k, where $iTask_j$ is the choreography task connected to C_i^P with an outgoing association, and PE_k is the process element (e.g., a choreography task, a gateway, etc.) which is a direct successor of $iTask_j$, is replaced with:

- A parallel split gateway GW_k.
- A control flow connection from $iTask_j$ to GW_k.
- A control flow connection from GW_k to PE_k.
- A control flow connection from GW_k to $mTask_i$.

In this way, $mTask_i$ starts being executed once the process reaches the begin of the scope of C_i^P.

TR 5. A conditional boundary event EC_i is attached to $mTask_i$. EC_i is triggered when the choreography task $tTask_j$, which is connected to C_i^P) with an incoming association, sends the TerminateScope message. In this way, $mTask_i$ stops being executed once the process reaches the end of the scope of C_i^P.

TR 6. An annotation A_i is created and attached to $mTask_i$ via an association relation. The text of A_i contains the expression $\{C_i^P.name\}$: *Scope*. In this way, M_i is required to evaluate the commitment.

It is worth noting that the BPMN 2.0 specifications do not allow boundary events to be applied to choreography tasks. Therefore, for TR5 to be applicable, the BPMN metamodel must be extended with the following changes:

– The *boundaryEventRefs* property is added to *ChoreographyTask* elements. This property is used to link zero or more boundary events to a choreography task.
– The *attachedToChoreographyActivityRef* property is added to *BoundaryEvent* elements. This property is used to link a choreography task to a boundary event.

3.2 Generating Smart Contract Code

Once the choreography diagram has been enriched with information on the messages required to evaluate commitments, it is then possible to generate the smart contract code. To this aim, we decided to use Solidity as the target smart contract language, given the high level of maturity of tools and documentation available.

Similarly to object-oriented languages such as Java, Solidity allows to create abstract smart contracts which, to be executable, need to be extended by a concrete smart contract. This allowed us to decouple the code that handles the lifecycle of the commitment, which is independent from the process model, to the one that handles the messages and evaluates the antecendent and consequent expressions. As shown in Fig. 5, we organized the automatically generated smart contract code into three contracts.

The abstract contract *Commitment* is responsible for implementing the commitment state machine presented in Sect. 2. To this aim, the functions *targetStarts()*, *targetEnds()*, *cancel()*, *suspend()*, *reactivate()*, *release()* and *tick()* correspond to the similarly named events in the commitment state machine. They are invoked to transition the commitment from one state to another one. Conversely, the functions *condA()* and *condC()* are responsible for evaluating, respectively, if the antecedent and consequent expression holds. Since these expressions are dependent on the type and content of the messages exchanged in the process, they are declared as abstract, leaving their implementation to the class that will extend *Commitment*. It is also worth noting that the *tick()* function is the only one with *public* visibility, whereas all the other functions are *protected*. In this way, participants cannot artificially trigger a change in the state of the smart contract. They can only notify the smart contract that some time has passed by invoking *tick()*.

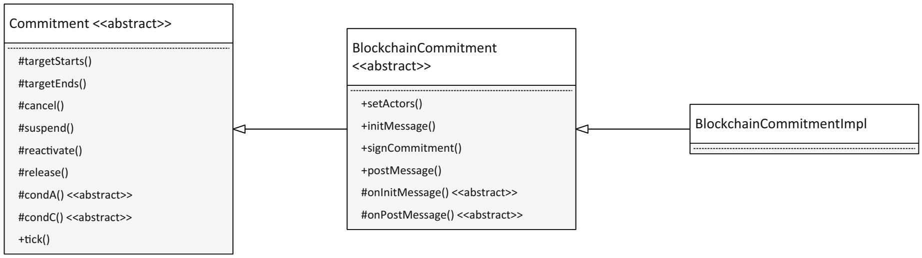


Fig. 5. UML class diagram showing the architecture of the smart contract code.

The abstract contract *BlockchainCommitment* extends *Commitment* by implementing the mechanisms to associate the messages with the participants, and to fire transitions in the commitment state machine by invoking the methods inmplemented in *Commitment*. To this aim, the function *setActors()* notifies to the contract the address of the debtor and the creditor. Similarly, the function *signDocument()* notifies to the contract that the debtor or the creditor agree on how the contract has been designed. The function *initMessage()* notifies to the contract which messages are expected to be sent, by whom, and if they belong to the InitScope, TargetScope, or Scope category. To do so, *initMessage()* stores the identity of the sender and the scope of the message. Then, *initMessage()* calls the *onInitMessage()* protected function, which is responsible for allocating the memory space required to store the message. Finally, the function *postMessage()* notifies to the contract that a new message has been sent, potentially causing the commitment to change state. To do so, after verifying that the sender is allowed to send the message, *postMessage()* calls the *onPostMessage()* protected function, which is responsible for storing the message. Then, if the message belongs to InitScope, *postDocument()* invokes the *targetStarts()* function. If the message belongs to TargetScope instead, *postDocument()* invokes the *targetEnds()* function. Finally, *postDocument()* invokes *tick()* to notify the state machine that time has passed. Since operations performed by *onInitMessage()* amd *onPostMessage()* are dependent on the structure of the message and of the data it contains, this function is declared as abstract, leaving its implementation to the class that will extend *BlockchainCommitment*.

The contract *BlockchainContractImpl* extends *BlockchainCommitment*, thus implementing the abstract methods inherited from *BlockchainCommitment* and *Commitment*.

With this architecture, the developer has to implement only the *condA()*, *condC()*, *onInitMessage()* and *onPostMessage()* functions. Since these functions are declared as abstract in *Commitment* and *BlockchainCommitment*, there is no need to alter the code in these abstract smart contracts. Instead, all the application-dependent code is entirely confined inside *BlockchainContractImpl*.

3.3 Deploying Smart Contract Code

Once the smart contract code has been generated, it is then possible to deploy it on a blockchain for each process instance. However, before being able to enforce the commitments, the smart contract also needs to be initialized. In particular, given a commitment C_i, the corresponding smart contract SC_i must be made aware of the identity of the debtor and creditor for that instance. Also, for each message belonging to the scope of C_i, SC_i must be made aware of the identity of the sender and the activity related events it triggers. Finally, SC_i must be signed by the debtor and creditor to indicate that they both agree in the way C_i was defined.

To this aim, given the enriched choreography diagram, the smart contract SC_i and a list L containing information on the participants in the process, the following deployment rules can be applied to the diagram.

DR 1. The debtor d and creditor c for C_i are identified from the choreography diagram. The debtor is the participant of the choreography task $cTask_k$ having an outgoing connection pointing to C_i. The creditor is the participant P'_j of the choreography task $cTask'_k$ having an incoming connection coming from C_i.

DR 2. SC_i is deployed on the target blockchain.

DR 3. The *setActors()* function of SC_i is invoked passing the address of d and c, which are extracted from L.

DR 4. For each choreography task $cTask_j$ that has an annotation A_j containing the expression $\{C_i.name\}$: $\{scope\}$, where $\{C_i.name\}$ is the name of C_i, the *initMessage()* function of SC_i is invoked passing the following parameters.

- The type of message, which is the name of the message M_j attached to $cTask_j$.
- The address of the participant, obtained by extracting from L the address of the participant that is the initiator of $cTask_j$.
- The type of activity-related event, which is $\{scope\}$.

DR 5. The *signCommitment()* function is invoked by both d and c.

The rules from DR1 to DR4 can be applied by any participant. However, rule DR5 must be performed by the creditor and the debtor, to confirm that they agree with the smart contract representing the commitment.

After that, the process can be monitored and the commitments enforced. In particular, participants are expected to send the messages modeled in the choreography diagram by invoking the function *postMessage()*. They are also expected to conform to the control flow dependencies modeled in the choreography diagram.

4 Evaluation

To validate our approach, we applied it to a real-world case study from the logistics domain. In particular, we focused on the last mile of a temperature-controlled delivery process, which is organized as follows: Firstly, a shipper picks

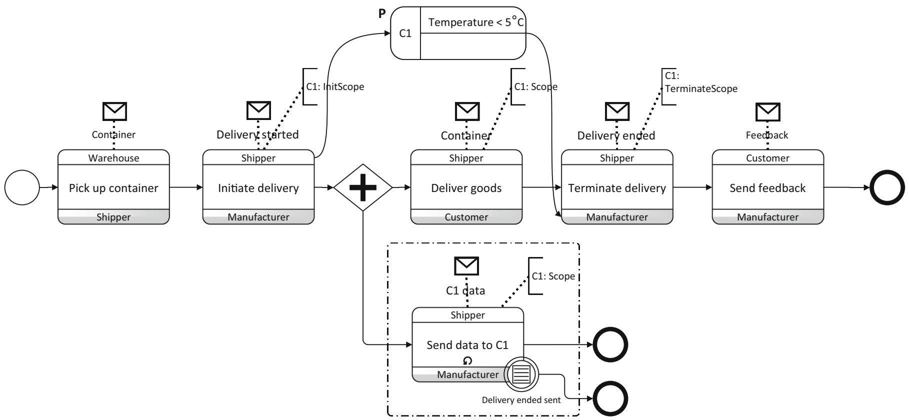


Fig. 6. BPMN choreography diagram with commitments obtained after applying transformation rules.

up a shipping container containing the goods to be delivered. Then, the shipper notifies the manufacturer of the goods that the delivery has started. After that, the shipper delivers the container to the customer, and notifies the manufacturer that the delivery is completed. Finally, the customer informs the shipper on the outcome of the delivery process. While the shipper is in charge of delivering the goods to the customer, he is subject to the following compliance rule: He must ensure that the temperature of the container stays below 5°C.

Since the participants do not want to disclose their internal processes (e.g., which route is taken by the shipper and under which conditions), this process can be modeled as a choreography. Also, the compliance rule can be modeled as a persisting commitment. Thus, we can use the extension of BPMN choreography diagrams with commitments to represent them, as shown in Figure 3b. In particular, commitment C1 is attached with an incoming arc to the *Shipper* actor in the choreography task *Initiate delivery*. In this way, we indicate that the shipper will be the debtor for C1, and that C1 will be enforced after *Initiate delivery*. Similarly, C1 is attached with an outgoing arc to the *Manufacturer* actor in the choreography task *Terminate delivery*. In this way, we indicate that the manufacturer will be the creditor for C1, and that C1 will no longer be enforced after *Terminate delivery*. Besides specifying the tasks carried out by the shipper, the compliance rule does not specify any other constraint on when it should be enforced. Therefore, no antecedent is specified for C1. Conversely, the consequent requires that the temperature of the container must not exceed 5 Celsius degrees.

Once we modeled the process and compliance rule as discussed above, we applied the transformation rules presented in Sect. 3.1 to identify the information required for blockchain-based monitoring. The resulting choreography diagram is shown in Fig. 6. In particular, by applying TR1, an annotation was added to *Initiate delivery*, specifying that the *Delivery started* message belongs to the InitScope for C1. Similarly, an annotation was added to *Deliver goods* and *Terminate delivery* to indicate that, respectively, the *Goods* message belongs to the Scope for C1, and the *Delivery ended* message belongs to the TerminateScope for C1. Also, since C1 is a persistent commitment, rules TR2 to TR6 were applied to add the information required to continuously evaluate the consequent. In particular, by applying TR2 and TR3, the loop choreography task *Send data for C1* and the message *C1 data* (which are highlighted) were introduced in the model. By applying TR4, a parallel split gateway was introduced after *Initiate delivery*, connecting to one branch *Deliver goods* (as originally specified in the process description) and on the other branch *Send data for C1*. By applying TR5, a condition boundary event is attached to *Send data for C1*, indicating that the task should stop being executed once the *Delivery ended* message – indicating that C1 should no longer be enforced – has been sent. Finally, by applying TR6, an annotation was added to *Send data for C1* indicating that the *C1 data* message belongs to the Scope for C1.

Once we enriched the choreography diagram as explained above, we generated the smart contract code as discussed in Sect. 3.2. Starting from the smart contract skeleton, we completed it by manually implementing the *onInitDocument()*, *onPostDocument()*, *condA()* and *condC()* functions[2].

Finally, we deployed and initialized the resulting smart contract code by performing the steps discussed in Sect. 3.3. Then, we used the deployed smart contract to monitor a simulated instance of the process. To perform the deployment and simulation, we used Truffle Suite[3]. Deploying the smart contract consumed 2521182 gas units. Initializing the smart contract consumed 731852 gas units. Signing the smart contract consumed on average 30272 gas units per participant. Finally, sending a message consumed on average 178652 gas units. This relatively high gas consumption is determined by data structures and functions required to store the messages on-chain, and to evaluate the antecedent and consequent condition. However, by relying on a distributed filesystem and a blockchain oracle, this information could be moved off-chain, significantly decreasing the gas consumption of the smart contract.

5 Related Work

In the pyramid of the process model correctness proposed in [7], compliance checking is related to the semantic correctness, i.e., to comply with imposed rules stemming from regulations, standards, and laws. As in [15], social commitments model the interaction among several participants inspired by the agent-based system literature, and translated into automaton as suggested in [5,13]. Social commitments have been already adopted to specify orchestrated processes [2] or to annotate choreographed ones [14]. In this way, the compliance concerns non-functional aspects of the business process execution which cannot be expressed using the usual process models.

[2] The resulting code is publicly available at http://purl.org/commitments-solidity.

[3] See https://trufflesuite.com.

According to the classification proposed in [6], the approach proposed in this paper can be placed among the hybrid approaches as it covers the compliance management at design-time, run-time, and post-mortem. In fact, commitments are used at design time, by linking them to the choreography process models, to express the compliance rules. The characteristics of the attached commitments also provide the information at run-time to check the compliance of the rules. Finally, the use of the blockchain introduces a novel aspect which completes the coverage of the compliance management lifecycle as it provides a tamper-proof technology for a distributed auditing.

A peculiarity of the approach is the focus on the choreographed processes. Unlike the orchestrated process, the compliance checking requires to enrich the interaction protocol among the parties with information concerning the way in which the interaction should occur. This is particularly relevant in case time constraints [1] or security constraints [8] are required.

A second relevant aspect covered in this paper concerns the usage of blockchain technology [4] to enable the creation of a distributed, while trusted, environment to transparently check whether the conditions expressed by the commitments are verified. In [11], the authors have studied how a blockchain like Ethereum plays an important role to support the technical challenges related to compliance checking. Similarly to our approach, [9] considers choreographed processes extending the semantics of the model, and proposes a blockchain-based solution to enact and monitor their execution. With respect to this approach, the solution proposed in this paper only focuses on monitoring, and it assumes the process to be enacted in a way that the blockchain is not able to control. As a consequence, monitoring the process becomes more challenging as the number of variants to be detected is higher.

To the best of our knowledge, mixing these two aspects - the usage of blockchain as a way to check the compliance of choreographed process - has not proposed in the literature yet. The closest approaches would be Caterpillar [10] and Lorikeet [17]. These approaches are based on collaboration processes which include, even if partial, some information about the internal structure of the pools. Model-driven techniques are adopted to automatically generate smart contracts which will support process execution and monitoring. A comparison among these approaches is provided in [3].

6 Conclusion

This paper has presented a model-driven approach to configure a blockchain-based solution to check the compliance of a multi-party business process. In particular, commitments are adopted to express obligations directly on a BPMN choreography model from which, according to a transformation rules, it is possible to derive the skeleton of code for the smart contracts which are involved in the monitoring.

In particular, the transformation rules defined in Sect. 3.1 identifies which message exchanges are relevant for a commitment and should be stored on-chain, thus answering to the research question RQ1. Moreover, by using the code-skeleton in Sect. 3.2 and the deployment steps in Sect. 3.3, we were able to easily turn commitments into smart contracts, and use the blockchain to monitor them, thus answering to the research question RQ2.

A limitation of this work, outlined during the validation, is the relatively high computational and storage requirements, that make it unsuited for low-value processes that rely on a public blockchain. This problem can be mitigated by relying on a permissioned blockchain which reduces the costs but does not provide the same level of interoperability, since the parties have to agree on the blockchain to be used. In addition to this, future work will focus on addressing this issue by implementing off-chain mechanisms to handle these issues.

Acknowledgements. This work has been partially funded by the Italian Ministry of University and Research under the MICS-PE program.

References

1. Capel, M.I., Mendoza, L.E.: Choreography modeling compliance for timed business models. In: Barjis, J., Pergl, R. (eds.) EOMAS 2014. LNBIP, vol. 191, pp. 202–218. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44860-1_12
2. Dalpiaz, F., Cardoso, E., Canobbio, G., Giorgini, P., Mylopoulos, J.: Social specifications of business processes with Azzurra. In: 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS), pp. 7–18 (2015). https://doi.org/10.1109/RCIS.2015.7128859
3. Di Ciccio, C., et al.: Blockchain support for collaborative business processes. Informatik Spektrum **42**(3), 182–190 (2019). https://doi.org/10.1007/s00287-019-01178-x
4. Di Ciccio, C., Meroni, G., Plebani, P.: On the adoption of blockchain for business process monitoring. Softw. Syst. Model. **3**, 1–23 (2022). https://doi.org/10.1007/s10270-021-00959-x
5. Ferrario, R., Guarino, N.: Commitment-based modeling of service systems. In: Snene, M. (ed.) IESS 2012. LNBIP, vol. 103, pp. 170–185. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28227-0_13
6. Hashmi, M., Governatori, G., Lam, H.-P., Wynn, M.T.: Are we done with business process compliance: state of the art and challenges ahead. Knowl. Inf. Syst. **57**(1), 79–133 (2018). https://doi.org/10.1007/s10115-017-1142-1
7. Knuplesch, D., Reichert, M., Mangler, J., Rinderle-Ma, S., Fdhila, W.: Towards compliance of cross-organizational processes and their changes (2012). https://doi.org/10.1007/978-3-642-36285-9_65
8. Köpke, J., Meroni, G., Salnitri, M.: Designing secure business processes for blockchains with SecBPMN2BC. Futur. Gener. Comput. Syst. **141**, 382–398 (2023). https://doi.org/10.1016/j.future.2022.11.013
9. Ladleif, J., Weske, M., Weber, I.: Modeling and enforcing blockchain-based choreographies. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 69–85. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_7
10. López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I., Ponomarev, A.: Caterpillar: a business process execution engine on the Ethereum blockchain. Softw. Pract. Exp. **49**(7), 1162–1193 (2019). https://doi.org/10.1002/spe.2702

11. Meironke, A., Seyffarth, T., Damarowsky, J.: Business process compliance and blockchain: how does the Ethereum blockchain address challenges of business process compliance? In: Human Practice. Digital Ecologies. Our Future. (WI 2019), pp. 1880–1891 (2019)
12. Meroni, G., Baresi, L., Montali, M., Plebani, P.: Multi-party business process compliance monitoring through IoT-enabled artifacts. Inf. Syst. **73**, 61–78 (2018). https://doi.org/10.1016/j.is.2017.12.009
13. Montali, M., Calvanese, D., De Giacomo, G.: Verification of data-aware commitment-based multiagent systems (2014)
14. Montali, M., Plebani, P.: IoT-based compliance checking of multi-party business processes modeled with commitments. In: De Paoli, F., Schulte, S., Broch Johnsen, E. (eds.) ESOCC 2017. LNCS, vol. 10465, pp. 179–195. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67262-5_14
15. Telang, P.R., Singh, M.P.: Specifying and verifying cross-organizational business models: an agent-oriented approach. IEEE Trans. Serv. Comput. **5**(3), 305–318 (2012). https://doi.org/10.1109/TSC.2011.4
16. Telang, P.R., Singh, M.P., Yorke-Smith, N.: Relating goal and commitment semantics. In: Dennis, L., Boissier, O., Bordini, R.H. (eds.) ProMAS 2011. LNCS (LNAI), vol. 7217, pp. 22–37. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31915-0_2
17. Tran, A.B., Lu, Q., Weber, I.: Lorikeet: A model-driven engineering tool for blockchain-based business process execution and asset management. In: Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018. CEUR Workshop Proceedings, vol. 2196, pp. 56–60. CEUR-WS.org (2018)
18. Verdicchio, M., Colombetti, M.: Commitments for agent-based supply chain management. SIGecom Exch. **3**(1), 13–23 (2001). https://doi.org/10.1145/844331.844334
19. Xu, X., Weber, I., Staples, M.: Architecture for Blockchain Applications. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-03035-3

The Dpex-Framework: Towards Full WFMS Support for Decentralized Process Execution

Christian Sturm(✉) and Stefan Jablonski

University of Bayreuth, Universitätsstraße 30, 95445 Bayreuth, Germany
{Christian.Sturm,Stefan.Jablonski}@uni-bayreuth.de

Abstract. Designing systems for inter-organizational process execution is not trivial and is affected by challenging issues due to the lack of a central coordinating authority. For this reason, current approaches connect to peer-to-peer architectures such as blockchains to synchronize all process participants upon a secure infrastructure. However, instead of integrating existing workflow management systems (WFMS) into decentralized infrastructures, a common strategy is to fuse the process world with the blockchain world by (re)implementing particular WFMS functions as smart contracts. Consequently, these approaches cannot encompass the functionality of mature WFMSs. We address this deficiency by introducing an abstract middleware layer that strictly decouples process management concerns from issues regarding the secure synchronization of decentralized WFMSs. This middleware then connects process engines with appropriate secure synchronization protocols, enabling (i) easy interchange of these key concepts and (ii) employment of existing WFMSs. Consequently, collaborations can leverage, for example, matured organizational policy management and collaboration-wide personalized work lists. We evaluate the feasibility of our approach with `dpex`, an open-source implementation, and its integration into a real-life use case.

Keywords: Decentralized Process Execution · Blockchain-based Process Execution · Inter-organizational Process Execution

1 Introduction

Enterprises collaborate with other companies to sharpen the focus on their core business competencies. For instance, they form supply chains by outsourcing certain activities of a value chain. Symbiotic economic effects, however, come with the cost of higher coordination efforts. Coordination and collaboration between companies can be realized through business processes. Business Process Management (BPM) generally investigates how business processes can be structured, automated, and optimized [5]. Thereby, the execution of a process model is a well-established strategy for companies to organize and optimize their internal

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 20–37, 2023.
https://doi.org/10.1007/978-3-031-41623-1_2

processes [28]. In this context, a workflow management system (WFMS) interprets a process model and guarantees the following key advantages: (i) it ensures conformance during execution, (ii) it enables global monitoring of the process state, and (iii) it provides personalized work lists for employees based on a model of the organizational structure [5]. Research on inter-organizational BPM transfers process execution into a cross-enterprise context, thus striving to reduce the administrative burden for cooperating ventures.

In contrast to company-internal process execution, inter-organizational collaborations lack a centralized infrastructure that is considered secure and trustworthy. In this regard, inter-organizational process execution can be conceptualized as a *decentralized application* with decentralized data management [9]. Therefore, a *secure communication infrastructure (SCI)* is mandatory. For providing such an SCI, recent research mainly suggests implementing decentralized process execution based on blockchain technology [21]. The earlier approaches focus on supporting a specific process modeling language on a particular infrastructure, for example, BPMN process diagrams on the Ethereum blockchain [16,23]. However, the monolithic solutions of encoding process semantics in smart contracts heavily interweave BPM matters and SCI concerns and cause the re-implementation of WFMS functionality within smart contracts. As a consequence, adaptions to low-level blockchain-specific smart contract code or the implementation of new smart contracts from scratch are necessary for supporting (i) alternative modeling languages, e.g., *YAWL* [6], (ii) sophisticated WFMS features such as the support of organizational policies [15], and (iii) alternative blockchain protocols, e.g., *Hyperledger Fabric* [20].

In this regard, the integration of an external WFMS (Camunda) and Hyperledger as SCI is discussed in [2]. In turn, the need for interchangeability of those core components, specifically blockchain protocols, is highlighted in [4,8]. In this context, current approaches utilize blockchain security advantages by outsourcing and automating the execution of particular activities as smart contract functions. This enables the implementation of critical use cases such as the exchange of cryptocurrencies, for example [8]. Instead, our architecture integrates blockchains as an instance of an SCI for collaboration-wide coordination based on work lists. In contrast to pure blockchain-based solutions [16], we propose a modular middleware that favors a strict decoupling of BPM and SCI concerns. This middleware empowers the integration of arbitrary WFMSs and connects to SCIs while ensuring proper communication between the process domain and the distributed systems sector. In particular, the suggested architecture fosters the following contributions.

- The **separation of** process **concerns** from the implementation of SCI logic is treated as a first-class citizen.
- **Integration of existing WFMSs** to retain process execution features, e.g., full-fledged work list coordination based on an organizational model.
- **Modular exchange** of core components to streamline the straightforward exchange of WFMSs and fault-tolerant infrastructures, such as blockchains.

– Enabling decentralized process execution **without a blockchain** by easy linking of existing WFMSs with traditional synchronization algorithms.
– In-depth consideration and support of **security requirements** to ensure proper operability and prevent intentional manipulation attempts.

We evaluate the feasibility of our approach by providing a proof-of-concept implementation of the conceptual framework called `dpex`. We demonstrate the integration in a real-life use case by integrating Camunda and Ethereum in a `dpex` application. To overcome the restrictions of Camunda and BPMN regarding the definition of organizational constraints, we highlight the flexible extensibility feature of `dpex`.

The remainder of the paper is structured as follows. Section 2 recaps the fundamental principles of WFMS-based process execution. Section 3 then introduces decentralized process execution in an inter-organizational context focusing on networking and security requirements. Section 4 recaps the requirements and presents our middleware architecture. Section 5 presents the `dpex` implementation and its integration in a real-world use case. We evaluate and discuss our approach in Sect. 6. Section 7 explores related work before Sect. 8 concludes the paper and previews future work.

2 Process Execution in a Workflow Management System

Process execution describes the software-supported execution of a process model that refers to a real-life business process. A process model is deployed in a software system called a workflow management system (WFMS) that is capable of distributing work items to employees or scheduling automated tasks automatically in accordance with the predefined process model. The main components of a WFMS are depicted in Fig. 1, and their functionality is described next.

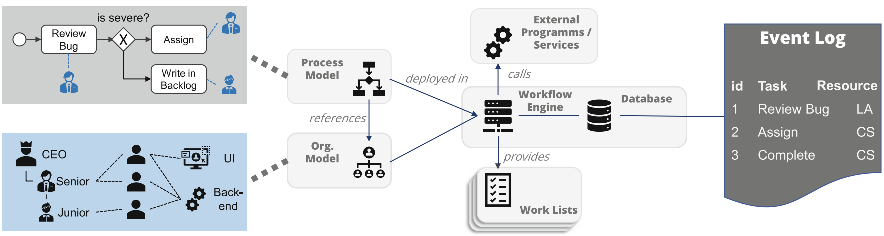


Fig. 1. Components of a WFMS

Main Components of a WFMS. The core of any WFMS is the *workflow engine* (execution engine) that interprets the semantics of a process model. Thus, the engine determines whether a specific task conforms to the process model and is further responsible for deriving subsequent possible actions. During execution, all occurring events, comprising, amongst others, the timestamp or ID, the task, and the executing human resource, are explicitly recorded in the *event log* that is persisted in a connected *database* along with further process-relevant data.

The main constituents or perspectives of a *process model* are the atomic partial steps of a process, called tasks (functional perspective), together with their temporal relationship allowing sequential or parallel execution (behavioral perspective). Additional information, which is captured in further perspectives of a process model, valorizes the WFMS-based process execution for companies [10]. Concerning the organizational perspective, process models reference entities of an externally defined *organizational model* to comply with organizational responsibilities. The box in the bottom left in Fig. 1 shows three employees allocated to roles (CEO, Senior, Junior) and groups (UI, Backend). The WFMS then may hand execution rights for a task only to employees that fulfill specified requirements, e.g., having a particular role or not having executed a preceding task. This is a fundamental principle for generating personalized *work lists* for employees by the WFMS. The automation of activities is subject to the operational perspective. Thereby, *executable programs* are assigned to activities whereby the execution thereof is automatically scheduled by the WFMS [19].

Advantages of a WFMS. The usage of a WFMS has several advantages [5]. First, a WFMS ensures process *conformance* to the predefined model by, e.g., safeguarding the execution sequence of tasks or preventing the accidental neglect of tasks. Then, employees are supported with personalized *work lists* according to the organizational restrictions, as defined along with the process model. On an administrative level, a WFMS enables the *monitoring* of global progress and *traceability* due to the fine-grained recording of events in an event log. Lastly, *automation* through implementing activities can boost business processes.

Execution of a Task. Usually, more than one employee can take responsibility for a particular task, i.e., the task is listed in multiple work lists. In this case, the WFMS is in charge of coordinating and synchronizing the execution of such a task to prevent, for example, activities from being performed twice. This feature is implemented according to the life cycle of a work item. A basic variant of such a life cycle (adapted from [19]) comprises the following steps. First, a work item is `offered` in the work lists of all eligible resources. Eventually, an employee sends an allocation request. Subsequently, the WFMS removes the work item from other employees' work lists and changes the work item status to `allocated`. The status of a work item is set to `completed` when the employee has finished task execution. When activities are automated, the WFMS may invoke a referred application program automatically while skipping life cycle stages.

3 Process Execution in an Interorganizational Context

Over the past decades, various approaches and concepts have been proposed for conducting inter-organizational processes. This section reviews the journey from message-based communication via decentralized process execution to blockchain-based process execution and emphasizes networking and security requirements.

3.1 Message-Based Collaborations

Before blockchain-based process execution, inter-organizational process execution focused on the specification of a message protocol and bilateral machine-to-machine integration of participants in a collaboration based on service-oriented architectures [25]. Each participant locally runs a WFMS; these WFMSs are coordinated through message exchange. Due to the lack of a tamper-proof centralized server [27], neither a global process model nor a global event log is available. Consequently, indispensable process execution features like collaboration-wide personal work lists or a global process status are hardly supported.

3.2 Decentralized Process Execution: Networking Perspective

The aforementioned WFMS features (cf. Section 2) are based on the availability of an event log and the respective process models. These artifacts are usually provided on a centralized client-server infrastructure, which is, however, not available under certain circumstances such as in inter-organizational settings. Therefore, decentralized process execution relies on peer-to-peer architectures and the following strategy to govern a global event log and retain process execution features. Each collaboration participant interprets the same process model in its locally running WFMS to facilitate a global event log. The WFMSs of all participants are constantly synchronized, i.e., when a task is processed locally at one participant, the collaborating WFMSs are notified, and their local process status is updated accordingly. The synchronized event log preserves globally coordinated work lists and status monitoring. For proper processing in distributed environments such as peer-to-peer networks, communication and synchronization require particular consideration.

Communication. The inter-organizational process model is interpreted by each participant locally. Sharing the global process state, including the execution history, requires continuous communication between process participants.

Synchronization. The participants constitute a distributed system affected by typical synchronization issues: when two participants claim a task simultaneously, listening nodes could receive the messages in a different order and assign privileges differently, resulting in varying process states. Concerning this issue, event sequencing mechanisms must be integrated to ensure proper processing.

Assuming reliable communication and synchronization techniques, each participating WFMS can agree upon an unambiguous execution state of a process. However, former research highlights the untrustworthiness of inter-organizational collaborations and the threat of manipulation attempts, wherefore security requirements for this system architecture are discussed next.

3.3 Decentralized Process Execution: Security Perspective

The security of a (process) application is a mandatory non-functional requirement and is particularly important in a cross-organizational context when decentralized process execution is conducted. As an exhaustive security discussion is out of this paper's scope, the following discussion tackles important security issues from a BPM point of view. Thereby, we differentiate between manipulation attempts after process execution and security threats during process execution.

We first focus on manipulation attempts after process execution. The global event log is a central artifact for security matters in decentralized settings. It must be protected against dishonest manipulation. Due to decentralized data management, each participant can manipulate the *local* event log, trying to manipulate the actual process course in retrospect. Hence, we must ensure that all honest participants agree upon an unfeigned event log and that such local changes are rejected. In this regard, the following security threats arise.

Manipulation. Assume a company that processes parts delivered by a supplier. The parts must be cured by the supplier, which is recorded in the event log. Unfortunately, the processing company stores those parts incorrectly why its production process fails. Unfairly, it tries to manipulate the event log by removing the curing step to accuse the supplier instead. Such manipulations must be prohibited by ensuring the *integrity* of the event log. A possible solution is to acknowledge each process action with a digitally signed message by all collaboration participants. **Repudiation.** Again assume the collaboration of companies. A quality check must be conducted what is feasible for all collaboration partners that can act as QM (quality management). In case of subsequent deficiencies with the controlled part, a company – formerly acting as QM – might want to disguise its responsibility for this quality check, although it has performed it. The security concept of *non-repudiation* addresses this issue, whereby all messages are digitally signed with asymmetric cryptography techniques.

Second, we concentrate on security issues during process execution. From a networking perspective, byzantine faults may occur during message distribution, meaning the (intended) sending of wrong information. We derive two security threats from this misbehavior.

Impersonation. Decentralized process execution also lacks a centralized authority for authentication purposes. Hence, malicious companies could impersonate a collaboration partner and claim tasks on their behalf unauthorizedly. This *authentication* issue can again be solved by asymmetric cryptography. **Conformance Violation.** Through malicious messages, companies could also try to claim tasks whilst required preceded tasks are not completed yet, or despite organizational restrictions. In other words, malicious companies try to violate process conformance. In WFMS-based process execution, all authorization rules are predetermined by a process model. Hence, the *authorization* security feature can be delegated to the preservation of process conformance.

3.4 Blockchain-Based Process Execution

Blockchain protocols were utilized to build process execution systems that are considered trustworthy. The following section discusses how a blockchain can meet the networking and security requirements (cf. Sect. 3.2 and 3.3).

Blockchains are built upon a peer-to-peer network and implement gossiping functionality so that the nodes in the network will receive new messages (called transactions). Hence, **communication** is enabled by default when using a blockchain. With respect to **synchronization**, blockchain-based systems broadcast messages without an agreed-upon order initially. They are collected in the mining pools of specific network nodes. To decide on the validity and order, the protocols integrate consensus mechanisms such as *proof of work* (*PoW*) or *proof of authority* (*PoA*), which algorithmically determine an accepted node that is in charge of validating and ordering a set of unconfirmed transactions by collecting and propagating them in a new block. In this step, these transactions get sequenced while the selected node can choose an arbitrary order. **Manipulations** are prevented by means of efficient integrity checks. The data is organized in contiguous blocks, where a subsequent block always contains a checksum-like reference to the data of the previous block in its header. Thus, when historical data is manipulated, all block headers change up to the current block. As creating new valid blocks is tedious and nodes always agree on the longest available chain of blocks, destroying the integrity and manipulating data is practically impossible. Asymmetric cryptography techniques enable both **non-repudiation** and authentication to prevent **impersonation**. All blockchain transactions must be signed with the sender's private key and are therefore non-reputable and unambiguously retraceable. Lastly, tackling **conformance violations**, blockchains not only store data but can also be configured to accept only new data that conforms to specific rules defined in so-called smart contracts. For our purposes, smart contracts are used to guard the semantics of a process model (language). Hence, an invalid transaction that violates process conformance is automatically rejected. When a valid transaction is included in a new block, it is considered persisted, and the process state is updated. Although blockchains meet the functional requirements, it has been discussed that there are more suitable approaches for collaboration between organizations [7]. In this paper, we address this issue and unchain decentralized process execution of blockchains and enable the integration of various trust-building modules.

4 Architecture for Decentralized Process Execution

In this section, we propose our middleware architecture for decentralized process execution. We first summarize the requirements for this architecture based on the challenges identified in the previous sections.

4.1 Requirements

Section 1 introduces the principle architecture and flexibility requirements identified in related work, e.g., [2,8], (R1, R2). The foundational concepts of WFMS-based process execution are presented in Sect. 2 (R3, R4, R5, R6), and the networking and security issues (R7) are identified in Sect. 3.

R1: Extensibility in Terms of Modeling Languages. Currently, we are not aware of a dedicated modeling language for inter-organizational processes or the organizational structure of collaborations. Hence, current approaches have adopted various process languages stemming from the intra-organizational domain, e.g., *BPMN Process Diagrams* [23], *DCR Graphs* [18], or *YAWL* [7]. Nevertheless, we regard it as necessary to be able to integrate arbitrary process modeling languages since collaboration requirements might be different in distinct application domains.

R2: Extensibility in Terms of SCIs. Multiple blockchain protocols have been adopted for process execution, which support different configurations. Permissionless PoW blockchains allow unrestricted and anonymous participation in the mining process, whereas permissioned PoA blockchains require explicit specification of empowered nodes. Besides blockchains, further promising synchronization algorithms exist [11]. An in-depth suitability study for process execution is still missing. Thus, our architecture must be open to arbitrary blockchain platforms and secure synchronization algorithms.

R3: Retain Basic WFMS Functions. We identified essential WFMS functions in Sect. 2. For instance, we regard process conformance checking, process monitoring, and in particular global work list handling as essential parts of a modern WFMS suitable for decentralized process execution.

R4: Support of the Operational Perspective. The invocation of tools and (web-)services is a mandatory feature of a WFMS to foster task automation. Run-time decisions on the selection of appropriate services are essential while a process model can specify constraints regarding the subset of possible resources.

R5: Support of the Organizational Perspective. Process execution requires a profound implementation of organizational perspectives. The first approaches to blockchain-based process execution focus solely on the control flow. More advanced solutions then include portions of the organizational perspective, e.g., voting-based resource allocation [15]. However, this does not fully cover the need for sophisticated organizational policies. This includes, e.g., an application-specific, flexible concept to specify eligible agents for process execution.

R6: Supervision of the Task Life Cycle. Support of the task life cycle is a critical feature. After the WFMS offers a task, the responsibility of task execution must be determined before an actor can perform it. Due to the distributed nature of the underlying infrastructure, different sub-nets (net partitions) may temporarily pursue task assignments in an uncoordinated way. This might happen when two participants from different network partitions claim tasks nearly at the same time. Hence, global coordination is mandatory.

R7: Network and Security Issues. Finally, the issues of decentralized process execution concerning networks, i.e., communication and synchronization, and concerning security, i.e., manipulation, repudiation, impersonation, and conformance violation, have to be reflected.

4.2 Architecture

The fundamental principle of our proposed architecture is to separate the concerns of the process domain and the collaboration domain. We achieve this by introducing a middleware allowing for the integration of (i) existing process software and (ii) secure communication infrastructures (SCI). The middleware then cares for proper communication between these key components automatically.

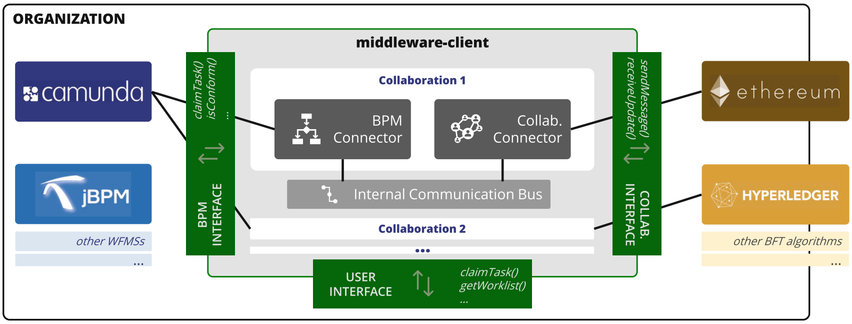


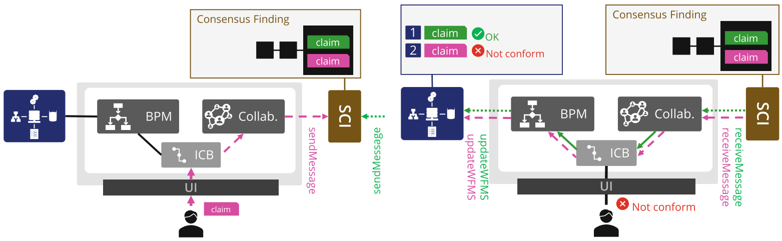
Fig. 2. Infrastructure of decentralized process execution with dpex

Consider Fig. 2 which illustrates the conceptual middleware architecture. Organizations may have different already deployed WFMSs running and also can maintain a local node of a peer-to-peer network. In this context, the middleware client is an additional service that manages multiple collaborations and can interact with various external process systems and different communication networks through so-called Connectors. The high-level BPM Interface and Collaboration Interface, which the respective connectors must implement, ensure flexible adaption to the external components, i.e., WFMSs and SCIs, respectively. Thereby, the Internal Communication Bus (ICB) handles the communication between a BPM Connector and a Collaboration Connector.

BPM Interface. The BPM Interface defines the core functionality that is mandatory for essential process execution such as `instantiate`, `getWorkList`, `isConform`, `claimTask`, or `completeTask`. The ICB relies on these functions to establish communication with an external WFMS. In the best case, an existing WFMS can satisfy the requirements of the inter-organizational process and provides an API for usage in the BPM Connector. However, if needed, the architecture allows extending the functionality of an external WFMS by design, for example, to add support for advanced organizational constraints (cf. Section 5.2).

Collaboration Interface. The Collaboration Interface specifies the messaging functionality `sendMessage` and `receiveUpdate`, whereby a Collaboration Connector is then in charge of implementing this functionality. This may require advanced considerations based on the configuration and implementation of the connected network. For example, in a permissioned Ethereum blockchain, deterministic finality is provided, meaning that once a new block update is propagated, this new data is considered irreversible, and the ICB can directly forward the update to the BPM Connector. Apart from that, a permissionless PoW blockchain provides probabilistic finality allowing for blockchain forks that invalidates propagated blocks with the included transactions. Hence, for permissionless blockchains, the Collaboration Connector should not forward updates directly but wait for a few blocks to arrive, as suggested by Falazi et al. [8].

Internal Communication Bus. The ICB cares for the proper invoking of components and information passing. Consider Fig. 3a. During a collaboration, a user initiates an update of the process state by claiming a task. Before the ICB updates the local WFMS, the Collaboration Connector is invoked to notify other participants of the task claim over the SCI (red, dashed arrow). Let us review a critical situation when another participant wants to claim the same task at the same time. In this case, a second claim message (green, dotted arrow) is propagated from an external dpex application in the shared SCI. The SCI is then in charge of synchronizing the messages. When the collaboration relies on an Ethereum network, claims are recorded as transactions waiting for validation. Eventually, the transactions are integrated into a new block which globally determines the order thereof. In our case, the external claim message (green) is set at position one, and the internal message (red) at position two. Consider now Fig. 3b, where the Collaboration Connector is triggered by the SCI to handle the transactions, which arrive in the respective order. The ICB forwards the messages to the WFMS via the BPM Connector while keeping their order. The WFMS will accept the claim of the first message, whereas the second claim is denied, as the task is already assigned.



(a) Two users (local and external) trying to claim a task over the SCI

(b) Eventually, the Collab. Connector receives both messages from the SCI

Fig. 3. Communication scheme with dpex

5 Implementation and Use Case

To elaborate on the feasibility of our middleware architecture, we provide an implementation thereof, called dpex, including connectors for integrating

Camunda and the Ethereum blockchain. This section first provides an overview of dpex before an implementation in a real-life use case is presented. The source code including examples and a demo screencast, is provided as a Git repository[1].

5.1 dpex-Library

dpex is an open-source Java implementation of the presented conceptual architecture. The framework is designed as a library and can be integrated into Spring Boot applications. According to the presented architecture, the purpose of dpex is not to interpret processes or find consensus but to integrate and interlink existing WFMSs and external SCIs. The core concepts and main classes of the framework are briefly presented next. The class `Alliance` refers to a certain inter-organizational process and captures all relevant information for execution. In particular, it references the abstract classes `ProcessModel`, `BPMEngine`, and `Collaboration` of dpex, which specify methods to be implemented according to the BPM Interface and the Collaboration Interface. When an alliance is instantiated, respective sub-classes must be provided that implement the connection. For example, a *CamundaAdapter* extending `BPMEngine` must specify the REST requests to a Camunda WFMS for claiming or completing tasks, etc. The `DPEXService` corresponds to the ICB and builds the core of the framework. In the course of flexibility, the business logic in this service is implemented solely based on the abstract classes. Given a specific alliance instance, the code of respective sub-classes is then invoked automatically. During process execution, the `DPEXService` is triggered by user interactions and invokes the appropriate BPM or collaboration adapters whilst passing the required information. The `Collaboration` class references the classes `Network`, `Agreement`, and `Security`, which can be extended, to wait for several additional blocks before triggering the ICB, which is required in case of using a permissionless blockchain.

5.2 Implementation in a Real-Life Use Case

Use Case. We demonstrate our approach with the implementation of the business trip application process at the University of Bayreuth (UBT). The UBT does not provide a centralized WFMS for this purpose, thus, three independent participants are involved, i.e., the responsible chair (RC), the personnel affairs department (PAD), and the accounting department (AD). The execution of the process (Fig. 4) must also be protected from fraud, e.g., the approval of the PAD should be non-repudiable because it is decisive for later reimbursement.

BPMN is incapable of expressing the required organizational constraints. Despite several BPMN extensions exist in this respect, we express the constraints in terms of BPMN annotations to demonstrate the flexible extensibility of dpex towards supporting any desired modeling constructs. An applicant starts the process by executing *Apply for trip*. The secretary of the applicant's chair must then prepare the application before an employee of the PAD can approve it.

[1] https://gitlab.com/bpm-dpex.

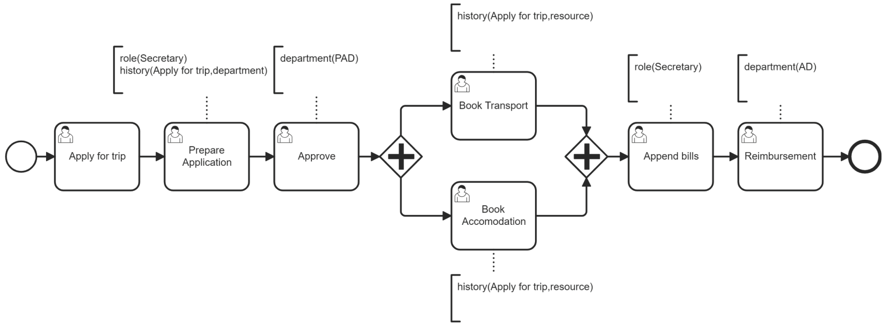


Fig. 4. Business Trip Process

After that, the traveling employee is in charge of booking transport and accommodation. When the trip is finished, the secretary collects all receipts accrued for the AD, which cares for reimbursement. Consider the mentioned screencast to review a possible infrastructure including required systems to be maintained.

Implementation. Before starting the collaboration with the dpex framework, the participants must agree on a process model and an SCI. Accordingly, any desired WFMS being able to interpret the process model must be deployed locally together with a respective client being able to connect to the selected SCI. Some SCIs require a locally running client (e.g. Ethereum), whereas other SCI algorithms can be implemented directly in the Collaboration Connector. For our use case, a Camunda WFMS and an Ethereum client are configured by each participant. Lastly, each participant runs an application including the dpex library that is configured to connect to Camunda and Ethereum in the following way. The dpex application implements the *CamundaMPModel* class as an extension of `ProcessModel`, which stores the BPMN elements for later access and processing of the annotations, which represent organizational constraints. The *CamundaMPEngine* class is responsible for implementing the BPM Interface. This class extends *CamundaEngine*, provided by the dpex library, which in turn extends `BPMEngine`. *CamundaEngine* already defines the required REST requests to Camunda. To address advanced organizational requirements, which Camunda is not applicable to interpret, the `isConform` functionality of *CamundaMPEngine* first calls an *OrgEngine* to evaluate organizational constraints directly before Camunda checks conformance w.r.t the control flow. On the collaboration side, we set up a private Ethereum network with a local node at each department. The wallet addresses of the participants are integrated into the global organizational model; for example, they are allocated to a group *research assistant* or a role *secretary*. This way, the *OrgEngine* can check, for example, if a role-based claim of a task is conform based on the sender of a message. The connector is then configured with the IP address of the local node. For SCI purposes, we extend the `Collaboration` class and integrate Ethereum using the *Web3j* library to send transactions within the Collaboration Connector. An event listener for smart contract events listens to newly validated transactions and triggers the ICB for further processing.

Manipulation Handling. Manipulation handling with dpex is different compared to pure blockchain-based solutions. Consider Fig. 5 which compares (i) process execution implemented directly on an Ethereum blockchain (left) with (ii) the usage of the dpex framework with Ethereum as SCI (right). The task *Apply for trip (AfT)* is already executed and recorded as a transaction on the blockchain. Now, through active manipulation or erroneous behavior, the transaction for executing *Book transportation* is propagated by RC. The Ethereum clients on the left-hand side immediately regard the transaction as invalid, but with dpex, the transaction representing a non-conform process action is still recorded on the blockchain. However, with the globally accepted order of (all) events, honest participants will agree on the same process state. Non-conformal events are detected as such by the local WFMS, and will be ignored and not be included in the local event log.

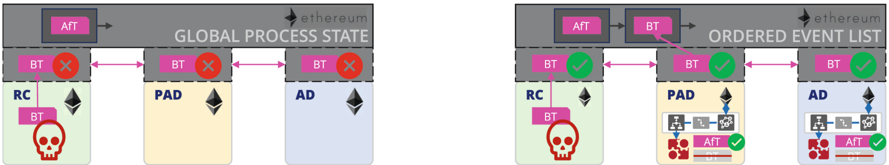


Fig. 5. Handling of non-conform transactions: pure blockchain-based solution (left) vs. dpex (right). The task life cycle is neglected in this example.

6 Evaluation of the Architecture

This section evaluates our approach and emphasizes whether and how the stated requirements are either directly supported by the proposed architecture or how they can be implemented in dpex due to the flexible design of the architecture.

R1, R2: Extensibility. First, we ensure that external systems can be integrated. In this regard, our proposed middleware defines high-level interfaces based on the essential requirements of process execution and networking. This enables the integration of existing components through connectors to entirely outsource process interpretation and consensus finding. We demonstrate this in the business trip application, where Camunda and Ethereum are integrated. As we do not rely on the disruptive capabilities of blockchains, e.g., proof of work or smart contracts, our architecture is also extensible with non-blockchain-based fault-tolerant synchronization algorithms from a conceptual viewpoint. Instead of securing the execution itself with smart contracts, the algorithms guarantee a globally accepted order of all occurring events according to our Collaboration Interface. Based on this globally accepted event order, all WFMSs can derive the process state at any time, even if non-conform events are recorded.

R3: Retain Basic WFMS Functions. Section 2 presented the key advantages of a WFMS, while an event log is the decisive artifact to support those features. Although blockchain-based process execution is based on the availability of a globally accepted data basis, most approaches omit manually implementing advanced WFMS functionality. Our approach is driven by a globally accepted order of events. Together with their interpretation in an external WFMS, the actual event log can be derived, and the features of global monitoring, traceability, and work lists are supported in consequence.

R4: Support of the Operational Perspective. The operational perspective connects external services to process execution. In contrast to centralized workflow enactment, all participating WFMSs will execute a particular task on independent nodes. Hence, we have to consider the consequences of multiple service invocations. We identify three cases. First, the service updates local, process-external systems such as ERP-Systems. In this case, multiple service invocation on each particular node is desirable. However, service definition, e.g., connection URLs, as a rigid part of the (globally uniform) process model must be synchronized over all participants, which could be challenging due to the heterogeneous internal infrastructures. Second, a service affecting external scope with deterministic behavior is called. This is straightforward, as every node will retrieve the same results, e.g., when an open API is called to retrieve flight traffic data. However, challenges can arise when the service is instructed to send an E-Mail, which causes multiple emails to be sent. Third, an external service is called that may return different results. This is a showstopper, especially when the result is decisive for the subsequent process execution, e.g., in the case of an external credit check, where a single, globally accepted result value is decisive. In this respect, a one-time smart contract call can realize the external services to retrieve an *agreed* value. Our evaluation shows that support for the operational perspective is not straightforward and requires additional consideration, e.g., a *skip* operation in WFMSs. This would allow a one-time execution of an inter-organizational service task, while the WFMS of non-responsible participants uses the *skip* operation to continue the process locally. Due to the stated issues, our architecture does not fully support the operational perspective currently.

R5: Support of the Organizational Perspective. In our use case, advanced organizational requirements apply, which BPMN can not express. Consequently, popular WFMSs such as Camunda rely on custom implementations. This impedes integration into our decentralized architecture because the process model as a single source of truth must capture all process-relevant information. In this regard, we specify the rules by means of BPMN annotations. The ICB injects the model into the *CamundaMPEngine*, which calls an *OrgEngine* that can finally evaluate the organizational constraints based on the process model, the current event log, and an organizational model. Due to the strict separation of this BPM matter from SCI concerns, the support of organizational perspective can be implemented independently of any communication infrastructure.

R6: Supervision of the Task Life Cycle. The task life cycle is explicitly supported in the architecture, as the BPM Interface specifies respective methods, e.g., `claimTask`. The implementation thereof is again not in the scope of the architecture but must be implemented by connected WFMSs. If more advanced life cycle models must be supported, the respective interfaces can be extended.

R7: Network and Security Issues. When the security of blockchain-based process execution is discussed, the protocol and its configuration must be considered [8]. Our architecture, however, does not promote a particular communication infrastructure. Instead, we outsource the implementation and solely assume that an (irreversible) globally accepted order of all messages can be provided. Hence, the security of our architecture is determined by the security of connected protocols, and the discussion thereof is out of the scope of this paper.

In conclusion, the dpex implementation indicates that the requirements regarding extensibility are addressed, as external systems can be integrated as long as they can serve essential functionality defined in the BPM and Collaboration Interface. The integration of existing WFMSs also allows for retaining their functionality. By means of the use case, we demonstrate sophisticated support of dpex with respect to the organizational perspective. The operational perspective is critical to implement because current WFMSs lack the functionality to coordinate a one-time invocation of external services. This requires particular consideration in future work. Due to the separation of SCI concerns, the security issues must be discussed by means of the integrated protocol.

7 Discussion of Related Work

Blockchains provide a trustworthy infrastructure for inter-organizational process execution [27]. Numerous approaches with varying focus have been published, which have recently been analyzed within systematic literature reviews. Viriyasitavat et al. provide a broad overview of the opportunities of integrating blockchains into the BPM domain [26], whereas Stiehle et al. instead focus on the categorization and comparison of blockchain-based *process execution* approaches [21]. Their studies show that existing approaches discuss the use of different modeling languages such as BPMN process diagrams [16,23], BPMN choreographies [17,27], YAWL [2], and Petri nets [7]. Also, declarative modeling paradigms have been investigated [18]. On the collaboration side, different blockchain protocols are adapted, for example, Ethereum [16,22], Hyperledger [20], or BFT blockchains using the PBFT framework for consensus building [7]. In terms of multiple process perspectives, the enforcement of organizational constraints is mainly supported through direct allocation [23] and role-based allocation [4,17,27]. However, most approaches support compile-time binding only instead of evaluating role assignments of possible participants to offer tasks in work lists. During run-time, related work is focused on deferred allocation using voting-based mechanisms [12,15]. Thereby, the important role of a profound organizational model in capturing collaboration needs has not been discussed yet. Also, time constraints have been considered in blockchain-based systems [1,14,24] whilst relying on the timestamp of a particular block in the blockchain.

The approach of Alves et al. integrates a listener to the Camunda WFMS to establish communication with a Hyperledger Fabric network [3]. Despite integrating a WFMS, the approach lacks flexibility in terms of exchanging the WFMS or blockchain protocol. Some approaches favor the integration of multiple blockchains into inter-organizational process execution. The BlockME approach of Falazi et al. [8] uses a multi-layer architecture to integrate a process engine independently from a certain blockchain configuration. Thereby, the process model refers to particular smart contract functions that are automatically invoked by a WFMS. Hence, instead of retaining prominent WFMS features, they aim to implement particular blockchain tasks (operational perspective). Thereby, their approach lacks flexibility in the selection of modeling languages. Further, the approach of Adams et al. favors flexibility and is independent of any specific WFMS or blockchain protocol [2]. However, their use case demonstrates that they focus on a machine-to-machine integration via blockchain, where each participant executes their local workflow additionally. In contrast, we favor a global process model to enable prominent process execution features (cf. Sect. 2). Ladleif et al. [13] consider cross-chain communication by design in their approach. In contrast, we do not make assumptions on the collaboration layer, so cross-chain communication is irrelevant in our architecture. Our focus is further not to support specific protocol features, such as the establishment of channels for confidentiality purposes but to retain traditional process execution features. Even though the multi-chain approach of Corradini et al. [4] supports flexibility in terms of the blockchain being connected, it is not a WFMS-based solution as they instead provide a translator per blockchain protocol to derive specific smart contracts that directly encode process semantics.

8 Conclusion and Future Work

Our architecture integrates existing BPM and secure communication infrastructure (SCI) into a decentralized process execution system. Top priorities comprise the extensibility and retention of features of WFMSs, such as personalized work lists. We provide an example implementation of the architecture, called `dpex`. It respects the demanded separation of process-related and SCI concerns. The `dpex` framework demonstrates an easy connection of WFMSs like Camunda to trusted infrastructures such as blockchains. Arbitrary extensions can seamlessly be deployed through the explicit definition of adapters, e.g., to evaluate organizational constraints. In contrast to related work that relies on blockchains, e.g., to decide on process conformance on-chain or automate particular tasks as smart contract functions, we instead use the blockchain as a trusted protocol to decide event ordering. We discussed the opportunity to integrate non-blockchain-based secure fault-tolerant algorithms. In future work, we investigate additional process systems (*Declare*, *jBPM*) and SCI systems (*Hyperledger*, *BFT-SMaRt*) that serve the high-level interface at first glance. The first experiments show promising results that will be provided on Git as soon as they are completed.

References

1. Abid, A., Cheikhrouhou, S., Jmaiel, M.: Modelling and executing time-aware processes in trustless blockchain environment. In: Kallel, S., Cuppens, F., Cuppens-Boulahia, N., Hadj Kacem, A. (eds.) CRiSIS 2019. LNCS, vol. 12026, pp. 325–341. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-41568-6_21

2. Adams, M., Suriadi, S., Kumar, A., ter Hofstede, A.H.M.: Flexible integration of blockchain with business process automation: a federated architecture. In: Herbaut, N., La Rosa, M. (eds.) CAiSE 2020. LNBIP, vol. 386, pp. 1–13. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58135-0_1
3. Alves, P.H., Paskin, R., Frajhof, I., et al.: Exploring blockchain technology to improve multi-party relationship in business process management systems. In: International Conference on Enterprise Information Systems (2020)
4. Corradini, F., et al.: Model-driven engineering for multi-party business processes on multiple blockchains. Blockchain Res. Appl. **2**, 100018 (2021)
5. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-642-33143-5
6. Evermann, J.: Adapting workflow management systems to BFT blockchains - the yawl example. In: 2020 IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW) (2020)
7. Evermann, J., Kim, H.M.: Workflow management on BFT blockchains. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. **15**, 14:1-14:22 (2020)
8. Falazi, G., Hahn, M., Breitenbücher, U., Leymann, F., Yussupov, V.: Process-based composition of permissioned and permissionless blockchain smart contracts. In: EDOC 2019. IEEE (2019)
9. Fridgen, G., Radszuwill, S., Urbach, N., Utz, L.: Cross-organizational workflow management using blockchain technology - towards applicability, auditability, and automation. In: Hawaii International Conference on System Sciences (2018)
10. Jablonski, S., Bussler, C.: Workflow Management: Modeling Concepts, Architecture, and Implementation (1996)
11. Kleppmann, M.: Designing Data-Intensive Applications. O'Reilly, Sebastopol (2017)
12. Klinger, P., Bodendorf, F.: Blockchain-based cross-organizational execution framework for dynamic integration of process collaborations. In: WI (2020)
13. Ladleif, J., Friedow, C., Weske, M.: An architecture for multi-chain business process choreographies, pp. 184–196 (2020)
14. Ladleif, J., Weske, M.: Time in blockchain-based process execution. In: 24th IEEE International Enterprise Distributed Object Computing Conference (2020)
15. López-Pintado, O., Dumas, M., García-Bañuelos, L., Weber, I.: Dynamic role binding in blockchain-based collaborative business processes. In: Advanced Information Systems Engineering (2019)
16. López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I., Ponomarev, A.: Caterpillar: a business process execution engine (...). Softw. Pract. Exp. **49**, 1162–1193 (2019)
17. Loukil, F., Boukadi, K., Abed, M., Ghedira, C.: Decentralized collaborative business process execution using blockchain. World Wide Web **24**, 1645–1663 (2021)
18. Madsen, M.F., Gaub, M., Høgnason, T., Kirkbro, M.E., Slaats, T., Debois, S.: Collaboration among adversaries: distributed workflow execution on a blockchain. In: Symposium on Foundations and Applications of Blockchain (2018)
19. 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
20. Schinle, M., Erler, C., Andris, P., Stork, W.: Integration, execution and monitoring of business processes with chaincode (2020)

21. Stiehle, F., Weber, I.: Blockchain for business process enactment: a taxonomy and systematic literature review. In: Marrella, A., et al. (eds.) BPM 2022. LNBIP, vol. 459, pp. 5–20. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16168-1_1
22. Sturm, C., et al.: A blockchain-based and resource-aware process execution engine. Future Gen. Comput. Syst. **100**, 19–34 (2019)
23. Sturm, C., Szalanczi, J., Schönig, S., Jablonski, S.: A lean architecture for blockchain based decentralized process execution. In: BPM Workshops (2019)
24. Tonga Naha, R., Zhang, K.: Pupa: Smart contracts for BPMN with time-dependent events and inclusive gateways. In: Business Process Management: Blockchain, Robotic Process Automation, and Central and Eastern Europe Forum (2022)
25. van der Aalst, W.: Process-oriented architectures for electronic commerce and interorganizational workflow. Inf. Syst. **24**, 639–671 (1999)
26. Viriyasitavat, W., Xu, L.D., Niyato, D., Bi, Z., Hoonsopon, D.: Applications of blockchain in business processes: a comprehensive review. IEEE Access **10**, 118900–118925 (2022)
27. Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., Mendling, J.: Untrusted business process monitoring and execution using blockchain. In: BPM - 14th International Conference, Proceedings (2016)
28. Weske, M.: BPM: Concepts, Languages, Architectures, 3rd edn. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-59432-2

A Reference Data Model to Specify Event Logs for Big Data Pipeline Discovery

Dario Benvenuti[1], Andrea Marrella[1(✉)], Jacopo Rossi[1], Nikolay Nikolov[2], Dumitru Roman[2], Ahmet Soylu[3], and Fernando Perales[4]

[1] Sapienza Università di Roma, Rome, Italy
{d.benvenuti,marrella,j.rossi}@diag.uniroma1.it
[2] SINTEF AS, Oslo, Norway
{nikolay.nikolov,dumitru.roman}@sintef.no
[3] Oslo Metropolitan University, Oslo, Norway
ahmetsoy@oslomet.no
[4] JOT Internet Media, Madrid, Spain
fernando.perales@jot-im.com

Abstract. State-of-the-art approaches for managing Big Data pipelines assume their anatomy is known by design and expressed through ad-hoc Domain-Specific Languages (DSLs), with insufficient knowledge of the dark data involved in the pipeline execution. Dark data is data that organizations acquire during regular business activities but is not used to derive insights or for decision-making. The recent literature on Big Data processing agrees that a new breed of Big Data pipeline discovery (BDPD) solutions can mitigate this issue by solely analyzing the event log that keeps track of pipeline executions over time. Relying on well-established process mining techniques, BDPD can reveal fact-based insights into how data pipelines transpire and access dark data. However, to date, a standard format to specify the concept of Big Data pipeline execution in an event log does not exist, making it challenging to apply process mining to achieve the BDPD task. To address this issue, in this paper we formalize a universally applicable reference data model to conceptualize the core properties and attributes of a data pipeline execution. We provide an implementation of the model as an extension to the XES interchange standard for event logs, demonstrate its practical applicability in a use case involving a data pipeline for managing digital marketing campaigns, and evaluate its effectiveness in uncovering dark data manipulated during several pipeline executions.

Keywords: Big Data Pipeline Discovery (BDPD) · Big Data Pipeline · Reference Data Model · Process Mining · Event Log · Dark Data · XES

1 Introduction

In the current era of Big Data and Internet-of-Things (IoT), we are witnessing the transformation of traditional working domains into new challenging cyber-

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 38–54, 2023.
https://doi.org/10.1007/978-3-031-41623-1_3

physical environments (e.g., smart manufacturing) characterized by the availability of a large variety of sensors that monitor the evolution of several real-world objects of interest and produce a considerable amount of data. Nonetheless, many data are stored for compliance purposes only but not turned into value, thus becoming *dark data*. Gartner defines dark data as the information assets organizations collect, process and store during regular business activities, but generally fail to use for other purposes (e.g., analytics, business relationships and direct monetizing).[1] Examples range from server log files, which can give clues related to the workers' habits while executing their tasks, to geolocation data that could reveal traffic patterns. Nowadays, storing and securing dark data usually entails more expenses and risks than the potential return profit [10,30].

The recent literature on Big Data processing agrees that discovering and interpreting the *Big Data pipelines* that run within the organization workflow is essential to valorise dark data for insights and decision-making [8,25]. Big data pipelines are composite steps for processing data with non-trivial properties, referred to as the Vs of Big Data (e.g., volume, velocity, etc.) [22]. They: *(i)* ingest raw data from disparate sources; *(ii)* process such data in the computing continuum, which offers on-demand resource provisioning through a fluid ecosystem integrating Cloud, Fog, and Edge technologies; and *(iii)* move it toward the data consumers, which undertake further transformations, visualizations, etc.

State-of-the-art approaches for managing Big Data pipelines work assuming their anatomy is known by design and expressed using one of the many available domain-specific languages (DSLs) [19].

To tackle this issue, in the context of the recently funded EU H2020 Data-Cloud project[2], one of the main targets is to realize a new breed of *Big Data pipeline discovery* (BDPD) solutions to infer the structure and behavior of a data pipeline by solely analyzing the *event log* that keeps track of its past executions. Relying on the similarity among the concepts of "data pipeline" and "business process", one of the project's vision is to leverage and customize well-established *process mining* techniques to reveal fact-based insights into how data pipelines transpire and access dark data [4]. However, traditional event logs used for process mining are limited in scope [2]. They include attributes tailored to recording sequence-flow details of process execution (e.g., timestamp and completion of activities, etc.), thus neglecting any data- and technological-related aspects needed to perform BDPD. In addition, to date, a standard format to specify the concept of Big Data pipeline execution in an event log does not exist, making it challenging to apply process mining techniques to achieve the BDPD task. This leads to the following research questions:

- **RQ1**: Which attributes are required in an event log to keep track of data pipeline executions and properly perform BDPD?
- **RQ2**: Which process mining techniques can be exploited to uncover and valorize dark data manipulated during data pipeline executions?

[1] https://www.gartner.com/en/information-technology/glossary/dark-data.
[2] https://cordis.europa.eu/project/id/101016835.

– **RQ3**: Does process mining provide an effective way to perform BDPD and uncover dark data in real-world data pipeline executions?

In answering these questions, in this paper, we: (i) formalize a universally applicable reference data model to conceptualize the core properties of a data pipeline execution; (ii) implement the model as an extension to the XES[3] interchange standard for event logs; (iii) demonstrate its practical applicability in a use case involving a real-world data pipeline for managing digital marketing campaigns; and (iv) evaluate the effectiveness of (some) process mining techniques in uncovering relevant dark data manipulated during pipeline executions.

The rest of the paper is organized as follows. Section 2 describes our research methodology, based on design science principles. Section 3 presents the relevant background on event logs and data pipelines, together with a concrete use case. The reference data model, its underlying design concepts, and an accompanying interchange format, are presented in Sect. 4. Section 5 demonstrates the practical applicability of the reference model in the use case. Section 6 evaluates the effectiveness of applying process mining over the reference model to perform BPDB and uncover dark data. Finally, Sect. 7 draws conclusions, discusses the limitations of this work, and traces future work.

2 Research Methodology

Our research methodology is inspired to the Design Science approach described by Johannesson and Perjons in [12]. The methodology is applied in five distinct sequential phases: problem formulation and objectives, requirements definition, design and development, demonstration and evaluation.

Problem Formulation and Objectives. In this phase, which is addressed in Sect. 1, we first specify the research problem to be tackled, i.e., *realizing a BDPD solution to identify and take advantage of the dark data accessed during a data pipeline execution*. In Sect. 3, we justify its significance in the Big Data processing field trough a motivating use case. Then, we elaborate on three research questions, i.e., RQ1, RQ2 and RQ3, to guide our research toward defining an artefact to solve the problem. A *reference data model* to specify a data pipeline and its core properties, and its implementation as an extension to the XES standard for event logs, represent such an artefact, which opens the possibility of applying process mining techniques to perform BDPD.

Requirements Definition. The second phase consists of eliciting the requirements for the outlined artefact. In Sect. 3, after providing the required background concepts on data pipelines and event logs, we discuss the main findings of our previous work [19]. In [19], we analyzed the literature on Big Data pipeline modeling to extract *three requirements* that guided us to formalize a novel DSL (called DC-DSL) toward a standardized representation of the structure of a data pipeline. However, while pipeline modeling through DSLs represents, by nature,

[3] https://xes-standard.org/.

a "subjective" and static view of reality, BDPD is "instance-driven", i.e., it targets extracting concrete pipeline execution data from event logs. In this direction, we rely on the main concepts defined in DC-DSL and its requirements to build the skeleton of our reference data model, and we augment it through the key process mining notions of "event" and "trace".

Design. Based on the analysis of the background and the requirements, in the third phase we make design decisions explicit, discussing the reference data model and describing its main features in Sect. 4. Moreover, we present in detail an implementation of the model as an extension to the XES interchange standard for event logs, which enables us to answer RQ1.

Demonstration. In the fourth phase, to answer RQ2, we demonstrate the practical applicability of the reference model in the use case. Specifically, we show in Sect. 5 how the targeted use of process mining techniques can support uncovering and understanding the dark data accessed during many executions of a real-world data pipeline for managing digital marketing campaigns.

Evaluation. Finally, to answer RQ3, in Sect. 6 we perform a preliminary evaluation involving 10 expert users from research institutions and companies engaged in Big Data pipeline management activities. The aim is to assess the effectiveness of applying process mining techniques over the reference model to untangle the relevant dark data manipulated by the use case data pipeline.

3 Background

3.1 Process Mining and Event Logs

Process mining [1] is a family of data analysis techniques that enable decision-makers to discover flowchart models from event data [5], compare expected and actual behaviours [7], and enhance models. It focuses on the real execution of processes, as reflected by the footprint of reality logged by the information systems (ISs) of an organization. The starting point is an *event log*, which is analysed to extract insights and recurrent patterns about how processes are executed. Event logs consist of *traces* that each correspond to one process instance. Each trace contains a sequence of *events* that occurred during the execution of the process instance. Events are related to a particular step in a process with an activity label, a timestamp, and a trace identifier.

To enable the exchange of event logs between different ISs, the process mining community has developed an interchange standard that defines the structure and general contents of event logs. Since 2016, the official IEEE standard for storing, exchanging and analysing event logs is XES (eXtensible Event Stream) [3]. In XES, event logs are organized in a three-level hierarchy of log, trace, and event objects, with a minimal set of explicitly defined attributes on each of the levels. The standard is designed to allow for additional attributes to extend its scope. Some relevant extensions to XES were proposed to support communications [13], privacy-preserving data transmission [24], and uncertain data management [21].

3.2 Big Data Pipelines

The concept of Big Data pipeline can be traced back to 2012 [23], where data pipelines are described as a "*mechanism to decompose complex analyses of large data sets into a series of simpler tasks*". Over the years, many definitions of a data pipeline were provided. Among the most relevant, in [20], the authors refer to a data pipeline as the "*path through which Big Data is transmitted, stored, processed and analyzed*". In [18], a data pipeline is defined as "*a complex chain of interconnected activities from data generation through data reception, where the output of one activity becomes the input of the next one*".

While the literature lacks a rigorous specification of the concept of data pipeline, some common features that are related to it can be identified:

- A data pipeline consists of chains of processing elements that manipulate and interact with data sets;
- The outcome of a processing element of a data pipeline will be the input of the next element in the pipeline;
- Each processing element of a data pipeline interacts with data sets considered as "big", i.e., with at least one of the Vs dimensions that is verified to hold.

Looking at the above characteristics, it is evident that many similarities exist between the concepts of "data pipeline" and "business process". With the main difference that any step of a data pipeline is thought to manipulate some data. Conversely, processes include activities that do not necessarily interact with any kind of data [1]. Nonetheless, since BDPD resembles the discovery of processes, as both require an event log to enact the discovery task, it is worth employing process mining techniques to support the development of novel BDPD solutions. To achieve this objective, a reference model that formalizes the main properties of a data pipeline and an extension of the XES standard for event logs is required to capture the data and technological aspects related to a data pipeline execution.

3.3 Big Data Pipeline Specification Trough DSLs

The literature on Big Data processing has proposed several ad-hoc DSLs for specifying the structure of a data pipeline in graphical format or as an XML file [19]. DSLs are specification languages targeted to describe a specific application domain. This is in contrast to a general-purpose language (GPL), which is broadly applicable across domains. Compared to GPLs, DSLs cannot cover all aspects of a given problem due to their limited scope. Still, they fill this gap with improved expressiveness, offering better domain-specificity and significantly improving collaboration between domain experts and developers [17].

In our previous work [19], we analyzed the literature on Big Data processing to categorize the existing DSLs for modeling data pipelines based on their expressiveness. We found that the majority of DSLs: (i) propose similar constructs having different semantics to specify a data pipeline; and (ii) are often characterized by an ambiguous semantics, which can hardly be formalized and does not enable the application of any reasoning technique. Driven by this analysis, in [19] we derived three requirements to build a novel DSL that integrates and formalizes the main concepts underlying the structure of a data pipeline:

1. The DSL must include a pipeline definition mechanism with a clear separation between design and run-time aspects and not limited to a specific technology stack, application domain or ad-hoc processing models;
2. The DSL must include a run-time support that considers pipelines as separate units, rather than a single unit, for individual pipeline steps;
3. The DSL must include an enactment approach with run-time driven execution and support for race-condition-free parallel branches.

The above requirements were realized through DC-DSL (DC stands for Data-Cloud), which enables different stakeholders to create Big Data pipelines exploiting containerization and orchestration technologies. These are required concepts to allow data pipeline execution on the resources available in the Computing Continuum. Since DC-DSL is *event log agnostic*, in Sect. 4, we show how we used it to build the skeleton of our reference data model, which - in contrast - will be aware of the key process mining concepts "event" and "trace". In addition, the reference model keeps track of many execution parameters used to monitor a data pipeline execution. They are typically recorded by ISs in different data sources and neglected by DC-DSL, thus becoming dark data.

3.4 Use Case

Let us consider the real-world case of a Big Data pipeline targeting higher mobile business revenues in smart marketing campaigns. This use case is offered by one of the small-medium enterprises involved in the H2020 DataCloud project. To discover the structure of the use case pipeline, we relied on the interview-driven methodology defined in our previous work [6], which allowed us to specify various simulation scenarios to frame the boundaries of all possible pipeline executions. Then, we generated a simulated event log in the traditional XES format using the Simio[4] tool, obtaining 10,000 execution traces (the log is available for testing at: https://dx.doi.org/10.5281/zenodo.7387553) compliant with the simulation scenarios. In Fig. 1, it is shown the Directly-Follow Graph (DFG)

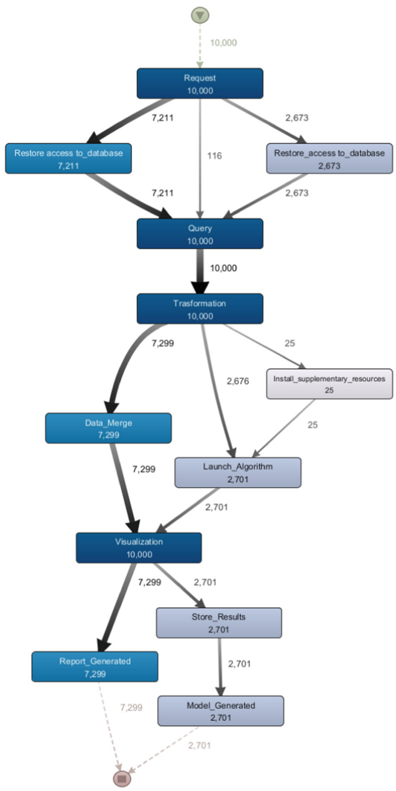


Fig. 1. DFG of the use case pipeline.

4 https://www.simio.com/.

representing the pipeline structure, discovered by feeding a process discovery tool (we used Disco by Fluxicon[5]) with the simulated event log. The pipeline is triggered when the system receives a request to model a new marketing campaign or to report on how an already existing one is performing. In both cases, the first two steps of the pipeline are querying the required data and applying specific transformations to it. If the request is for a report, there is a need to merge the queried data. On the other hand, for a model request, an algorithm to compute it is launched, and the results are stored in dedicated databases. Finally, the pipeline ends with either the report or the model being generated. By applying traditional process mining techniques, only sequence-flow details of the pipelines and related metrics can be obtained. This information can be used to grasp insights on the workflow running behind the data pipeline (e.g., how steps are sequenced, branching probabilities, potential bottlenecks, and other issues in the process flow). While useful, they do not allow us to infer further details from different perspectives, e.g., the flow of data accessed and manipulated during pipeline execution, the technologies used to process the Big Data in each pipeline step, etc. In a nutshell, *there are relevant execution data that could be easily captured by any logging system during pipeline execution, but are lost during the analysis, thus becoming dark data*. The first step to enable process mining techniques accessing and elaborating such data is to capture them in the event log, as shown in the next sections.

4 Reference Model and XES Extension for Event Logs

In this section, we present our reference data model to capture data pipeline executions by analyzing its UML class diagram, which is shown in Fig. 2, and explaining how it relates to the concept of event log. Then, Sect. 4.2 examines how to extend the XES standard to capture the properties defined in the model.

4.1 Reference Data Model

We start by looking at the class ***Big Data Pipeline***, which has an *ID*, a *Name*, and a *Communication Medium* (e.g., a message queue) on which data flows. Each Big Data Pipeline needs to have at least one ***Step*** by definition, and a Step belongs to only one pipeline. As can be seen from its attributes, a Step has an *ID*, a *Name*, and operates on a *Continuum Layer* (e.g., edge, fog or cloud) and has a *Type* depending on the computed data transformation (e.g., it can be a data consumer, a data producer, or both). Finally, a Step needs to have at least a ***Data Source***. A Data Source has an *ID*, a *Name*, a *Type*. The latter specifies if it is used as input, output, or both, and can be characterized by how it relates to the Vs of Big Data, e.g., by looking at its *Volume*. We specialized Data Source by highlighting ***Data Streams*** which are data sources with a certain *Velocity*, and we acknowledge that this class can be further specialized to include all the

[5] https://fluxicon.com/disco/.

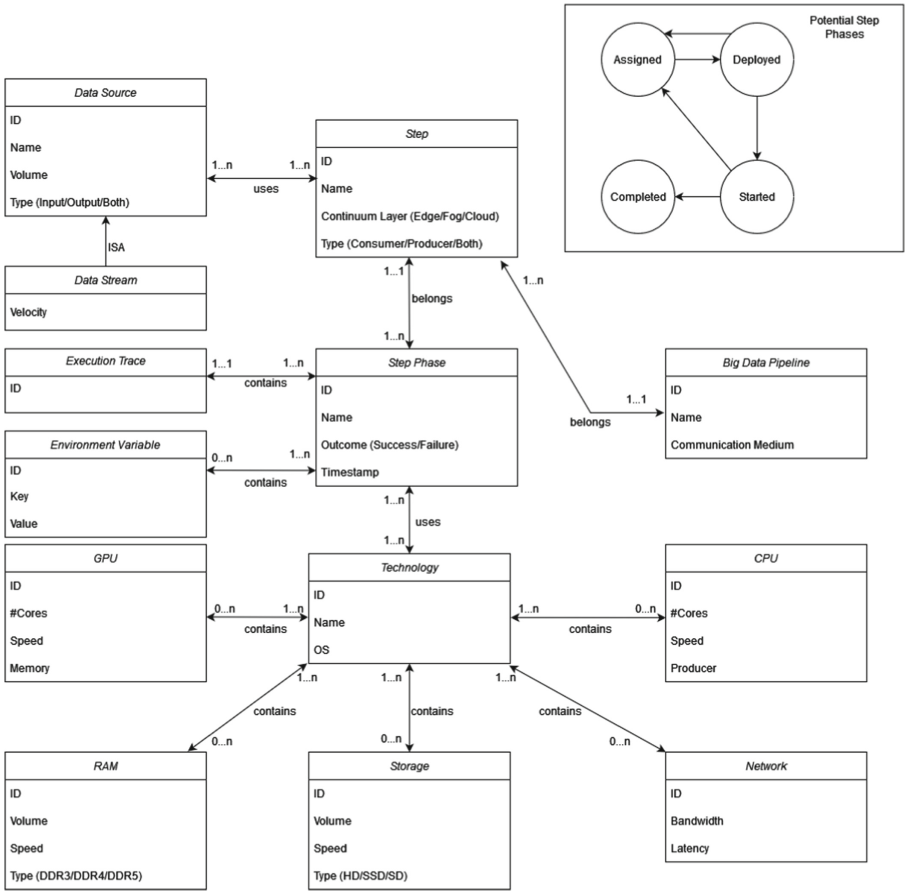


Fig. 2. The UML class diagram for the reference data model.

different Vs that can be appropriated to the context in which the model will be used. Finally, a Data Source needs to be used by at least one Step. A Step is made up by at least one ***Step Phase***, which is the core component of the reference model. A Step Phase belongs to only one Step and has an *ID*, a *Name*, an *Outcome* (either success or failure), and a *Timestamp*. We highlight four potential step phases that are commonly used to describe the life cycle of the steps of a data pipeline, *Assigned*, *Deployed*, *Started* and *Completed*. Still, we acknowledge that this could vary depending on the context. A Step Phase taps at least from one ***Technology***, which has an *ID*, a *Name*, and an *Operating System* (OS). We consider of interest only Technologies used by at least one Step Phase. Detailed technological information can be expressed using the ***GPU***, ***CPU***, ***RAM***, ***Storage*** and ***Network*** classes, which contain an *ID* and a series of self-explanatory attributes related to the specific class. We consider of interest only CPUs, GPUs, RAMs, Storages, and Networks that are related to at least

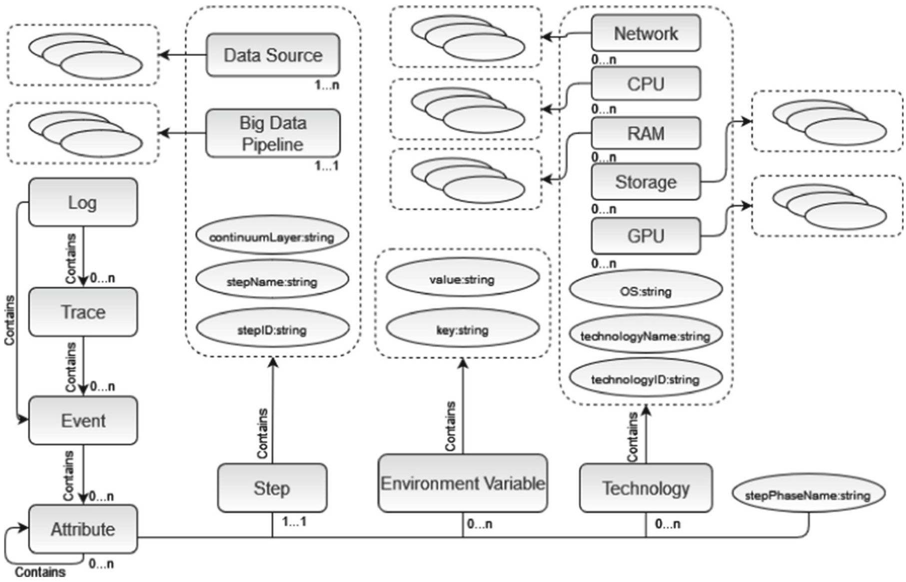


Fig. 3. Extension of XES derived from the proposed reference model. The nested attributes at the level of each Container attribute (e.g., Data Source or Big Data Pipeline) are omitted to avoid overwhelming the diagram.

one Technology. Optionally, a Step Phase can be associated with ***Environment Variables***. They consist of simple pairs of *Key* and *Value* attributes, which can be used to describe the Step Phase domain-specific properties. An Environment Variable is of interest only if it is associated with at least one Step Phase. Finally, we model an ***Execution Trace*** with its *ID*. Each Execution Trace needs to contain at least one Step Phase, and each Step Phase is contained only in one Execution Trace. We point out that *Data Source* and *Environment Variable* are derived directly from DC-DSL, while *Technology*, and each class in relationship with it, has been customized by interpreting the requirements details defined for DC-DSL. It is worth noticing that while we based our reference model on some of the core concepts of DC-DSL, its structure has been iteratively evaluated with the domain experts of the business cases involved in the DataCloud project.

The main connection between the proposed UML class diagram and the concept of event log is the tight relation between pipeline steps and business process activities. Indeed, traditional business processes, whose executions are responsible for generating event data, consist of activities, while data pipelines contain a sequence of steps. Thus, a pipeline's step can be seen as an activity that performs some data transformation. We exploited this notion in the reference model by associating each Step with the information about its data sources and the computing layer (i.e., Cloud, Edge, or Fog) on which the transformation is applied. Even if steps are the core elements of a data pipeline, we decided to further split a single step into different phases. In this way, when it comes to run-

Table 1. Attributes at the Event and Trace levels in the XES extension.

Attribute Level	Key	Type
Trace	ID	Trace ID
Event	Step Phase ID	string
Event	Step Phase Name	string
Event	Outcame	boolean
Event	Timestamp	Date
Event	Step	Container
Event	Environment Variable	Container
Trace	Big Data Pipeline	Container
Event	Data Source	Container
Event	Technology	Container
Event	CPU	Container
Event	GPU	Container
Event	RAM	Container
Event	Storage	Container
Event	Network	Container

time, information about when a step enters one of the phases of its lifecycle can be exploited for analysis purposes. Hence, an event log following the proposed model will have multiple entries for the same pipeline step, one for each of its phases.

4.2 Extending XES

Figure 3 describes an extension of the XES standard able to represent Big Data pipelines as formalized in the reference data model. As reported in the latest XES Standard Definition, the concepts of *log*, *trace*, and *event* contain no information, but they only define the structure of event data. Thus, information in an event log should be stored in *attributes*. All attributes in XES have a string-based key. Logs, traces, and events each contain an arbitrary number of attributes. There are six types of elementary attributes, each defined by the type of data value they represent, and two complex types: *List* and *Container*. These attributes hold any number (may be empty) of child attributes. The value of a List/Container attribute is derived from the values of its child attributes. Only in the case of the List, child attributes are ordered and their keys need not be unique.

For this reason, except for Execution Trace and Step Phase classes that match with the concept of *Trace* and *Event* in XES, we translated any other class included in the UML diagram of the reference model in a *Container* attribute. At the same time, we exploited the *List* attribute to represent relations between classes. The child attributes of any reference model class have been represented with one of the elementary attribute types.

In Table 1 we show the definition of the main attributes in the XES extension, at the *Event* and *Trace* levels. In Table 2, it is shown how we translated

Table 2. List attributes in the XES extension to represent relations between classes.

Attribute Level	Key	Type
Event	Environment Variables	list[Environment Variable]
Event	Technologies	list[Technology](*)
Technology	CPUs	list[CPU](*)
Technology	GPUs	list[GPU](*)
Technology	RAMs	list[RAM](*)
Technology	Storages	list[Storage](*)
Technology	Networks	list[Network](*)
Event	Step	Step
Step	Big Data Pipeline	Big Data Pipeline
Step	Data Sources	list[Data Sources](*)

relationships between classes through lists at the levels of one of the container of the classes participating in the relationship. The lists highlighted with *(*)* should contain at least one item. Finally, we translated each attribute of the UML classes into Attributes at the level of each respective Container. For the sake of space, we do not show here all the details of the XES extension, whose specification (obtained through the above considerations) is straightforward. Nonetheless, it is worth noticing that its definition represents our answer to **RQ1**.

5 Demonstration

To tackle **RQ2**, in this section we discuss how process mining techniques can be applied over our XES extension to: (i) get more detailed process-centric information on the use case pipeline of Sect. 3.4; (ii) obtain data-centric insights about its execution and untangle dark data accessed by the pipeline to generate new potential business value.

First, the XES extension works when an IS is recording events at the step phase level. This enables us to extract more insights into the executed steps of the big data pipeline using traditional process discovery techniques. Indeed, by applying one of the process discovery techniques available in [5] over the XES extension, we can: (i) get the duration of individual step phases and more accurate measures of the overall duration of steps; (ii) get more detailed information about which step phases are causing bottlenecks or other performance issues; (iii) have an idea of how the execution of pipeline steps is distributed on the computing continuum; and (iv) understand how different step phases interleave.

For example, by analyzing the DFG in Fig. 1, we notice that sometimes after the *Transformation* step there might be the need of installing supplementary resources. The usage of the XES extension would allow traditional process mining tools like Disco to get more detailed information about this issue, such as understanding: (i) in which phase of the *Transformation* step these additional resources are needed; (ii) how it relates to the outcome of the current step phase;

(iii) in which continuum layer the associated computations are taking place, and (iv) if the remote (hardware) resources are needed to consume or produce data within the pipeline execution. Even more important, with a complete knowledge of the resources employed in each step phase it may support a fast understanding of the technological threshold that was exceeded during the *Transformation* step, which led to the need of installing supplementary resources.

Second, when both the DC-DSL representation of a data pipeline and its past executions recorded into an event log are available, we can apply traditional conformance checking techniques [7] to detect discrepancies (e.g., steps missing in some pipeline execution or performed in a wrong order, etc.) between the expected pipeline behavior and its observed executions. In addition, trace alignment [14] can further support experts to associate a severity to such discrepancies suggesting a repair strategy to fix them. For example, in the presence of an event log represented through our XES extension, we can catch the exact moment in the range of a step life-cycle where a deviation occurs. By looking at the data pipeline model of Fig. 2, there could be execution traces in which the "Query" step starts before the "Transformation" step but is completed after it, generating a misalignment.

Third, to obtain further insights from the event log other than the traditional sequence-flow based information and give value to the dark data involved in many pipeline executions, we can leverage some recent process mining solutions that exploit database (DB) theory to carry on analysis on the process behavior recorded over a relational DB. A couple of approaches specifying DB schemas fully compatible with the XES standard have been proposed to store log data [9,29], and they can be easily extended to accommodate the additions provided by our XES extension. Overall, our idea is to build patterns of SQL queries to discover relations between the steps of a data pipeline that would be not made explicit by relying on traditional process discovery techniques. In this direction, we present some interesting query performed on a DB whose schema perfectly matches the UML class diagram of the reference model in Fig. 2.

For example, to discover potential dark data sources, we could search for all the data sources that appear in the log as the output of some steps but are never used as input, as it is expressed in the following query (Query 1):

```
SELECT ds.name as DataSourceName
FROM DataSource ds
WHERE NOT EXISTS(
    SELECT * FROM stepUsesDataSource suds
    WHERE suds.dataSourceName = ds.name
    AND (ds.type = 'Input' OR ds.type = 'Both'))
```

Similarly, we may be interested in knowing the amount of dark data that the pipeline is producing on the cloud. Since dark data are known to be a potential risk for organizations, having them on the cloud can increase the probability of attackers that exploit them. To get an idea of this measure, we can run the following query (Query 2):

```
SELECT  ds.name as DataSourceName, ds.volume as DataSourceVolume
FROM    Step s, DataSource ds, StepUsesDataSource suds
```

```
WHERE   s.continuumLayer = 'Cloud' AND s.name = suds.stepName AND
        ds.name = suds.dataSourceName
AND NOT EXISTS(
        SELECT * FROM StepUsesDataSource sudsrc
        WHERE   sudsrc.dataSourceName = ds.name
        AND (ds.type = 'Input' OR ds.type = 'Both'))
```

Moreover, if we perform one of the queries available in the literature [26,27] implementing process discovery over an event log stored as a relational DB, we can translate the DFG of Fig. 1 in a DB view *DFG(StepPhaseID1, StepPhaseID2)*, in which each record describes pairs of subsequent step phases in the DFG. Coming back to our use case pipeline, we can rely on such a view to understand if the need to install supplementary resources is associated with the outcome of the phases of the step *Transformation*. To this aim, we can run the following query (Query 3), which is searching for the outcome of all the step phases belonging to the step *Transformation* followed by a step phase belonging to the step *InstallSupplementaryResources*:

```
SELECT  sp1.name as StepPhaseName, sp1.outcome as
    StepPhaseOutcome
FROM    Step s1, Step s2, StepPhase sp1, StepPhase sp2,
        StepPhaseBelongsToStep spbts, DFG d
WHERE   (d.stepPhaseID1 = sp1.ID
    AND sp1.ID = spbts.StepPhaseID
    AND spbts.StepID = s1.ID
    AND s1.name = 'Transform')
AND     (d.stepPhaseID2 = sp2.ID
    AND sp2.ID = spbts.StepPhaseID
    AND spbts.StepID = s2.ID
    AND s2.name = 'InstallSupplementaryResources')
```

In a similar fashion we can obtain the resource threshold that is triggering this issue, by looking at the technologies used by step phases belonging to the step *Transformation* and followed by a step phase belonging to the step *InstallSupplementaryResources*, as expressed in the following query (Query 4):

```
SELECT  t.ID as TechnologyID, t.name as TechnologyName
FROM    Technology t, StepPhaseUsesTechnology sputc,
        Step s1, Step s2, StepPhase sp1, StepPhase sp2,
        StepPhaseBelongsToStep spbts, DFG d
WHERE   (d.stepPhaseID1 = sp1.ID
    AND sp1.ID = spbts.StepPhaseID
    AND spbts.StepID = s1.ID
    AND s1.name = 'Transform')
AND     (d.stepPhaseID2 = sp2.ID
    AND sp2.ID = spbts.StepPhaseID
    AND spbts.StepID = s2.ID
    AND s2.name = 'InstallSupplementaryResources')
AND     sp1.ID = sputc.StepPhaseID
    AND t.ID = sputc.TechnologyID
```

Once we have the IDs and names of those technologies, we can fetch more details about GPU, CPU, RAM, Storage or Network by performing simple queries, which are omitted here for the sake of brevity.

Finally, in many cases, dark data derives from domain-dependent values that are stored in the log but can not be exploited by general-purpose process discovery techniques. To address this issue, we can structure generalized queries that can be instantiated on a domain basis to infer insights on those values which would have not been used in a traditional setting. An example of such a query (Query 5) is the one in which we fetch all the steps containing at least a step phase with a specific pair of key and value between its environment variables:

```
SELECT  ev.key as Key, ev.value as Value, s.name as StepName
FROM    Step s, StepPhase sp, EnvironmentVariable ev,
        StepPhaseBelongsToStep spbts,
        StepPhaseContainsEnvironmentVariable spcev
WHERE   spbts.StepID = s.ID AND spbts.StepPhaseID = sp.ID
    AND sp.ID = spcev.StepPhaseID AND spcev.
        EnvironmentVariableID = ev.ID
    AND ev.key = 'domain-specific-key'
    AND ev.value = 'domain-specific-value'
```

6 Preliminary Evaluation

To address **RQ3**, we performed a test involving 10 Big Data pipeline management user experts. We aimed to understand if the use of targeted process mining techniques enabled us to perform BDPD effectively over an event log expressed through our XES extension. Specifically, starting from the event log employed in the real-world use case of Sect. 3.4, suitably augmented to accommodate the new concepts introduced in the XES extension, we presented to the users the results of the application of process mining techniques over the event log, as discussed in Sect. 5 (thus, related to process discovery, conformance checking and querying mechanisms). For each suggested application of process mining, we administered the users a questionnaire asking to rate: (i) the perceived effectiveness of the adopted solution in performing BDPD and uncovering dark data, and (ii) the complexity of extracting dark data from pipeline executions without the support of process mining. Questions are rated with a 4-point average scale structured as follows: 1 ("None"), 2 ("Low"), 3 ("Moderate"), 4 ("High"). Each user was allowed to add textual feedback to explain a score better. The average score for any question is reported in Table 3.

Table 3. Questionnaire results.

Process Mining solution	Effectiveness in performing BDPD and uncovering dark data (1–4)	Complexity of extracting data without BDPD (1–4)
Process Discovery	3.4	3.4
Conformance Checking	3.8	3.2
Query 1	3	3
Query 2	3	3.6
Query 3	3.8	3.6
Query 4	3.6	3.6
Query 5	3.6	3.6

The majority of the users (70%) were selected from the business case partners of the H2020 DataCloud project, which focus their business on managing big data pipelines targeting reduced live streaming production costs of sports events, trustworthy eHealth patient data management, and analytics in Industry 4.0 manufacturing. The remaining 30% of users are academics engaged in research activities related to data pipeline definition, deployment, and adaptation.

The questionnaire results clearly outline that: (i) process mining solutions represent an effective means to perform BDPD towards data pipeline analysis and dark data extraction, and (ii) it is extremely complex to obtain the same findings shown in Sect. 5 by employing traditional Big Data processing solutions. These conclusions are enforced by the fact that all the scores range from 3 to 4. It is worth noticing that, in their textual feedback, many users pointed out that similar results to process discovery and querying mechanisms could be obtained through an extensive analysis of the system logs recorded during pipeline executions by the IS of a company, which are often scattered among different data sources. The users also recognized this is a time-consuming and error-prone activity requiring manual and trial-and-error testing over the logs. On the other hand, the users confirmed that the same results obtained by applying CONFORMANCE CHECKING in Sect. 5 can not be straightforwardly emulated using traditional Big Data processing techniques, unless ad-hoc programming scripts are developed.

7 Concluding Remarks

In the era of Big Data, the application of process-oriented solutions to deal with issues requiring data awareness is increasing [11,15,16]. In this context, the analysis of Big Data pipelines and the realization of novel techniques for efficiently managing their life-cycle is considered as a relevant research challenge in the field of Big Data processing. In this context, a BDPD solution that provides a precise knowledge of the characteristics of the processing steps performed during a pipeline execution (e.g., CPU usage, resource consumption, size of the involved data, etc.) can be used for better scheduling the available cloud resources, enabling smart load-balancing and memory management decisions before the execution of a data pipeline. The discovery activity is also crucial to interpret bottlenecks, inefficiencies and risks hidden behind the complexity of data pipelines, which prevent or delay their proper enactment.

To realize this vision, in this paper we have formalized a universally applicable reference data model to conceptualize the core concepts and properties of a Big Data pipeline execution. We provided an implementation of the model as an extension to the XES interchange standard for event logs and demonstrated its practical applicability in a concrete use case data pipeline.

Our reference model can contribute to the Big Data pipeline field by providing a common application-independent conceptual framework for capturing data pipeline executions. Moreover, by achieving the objectives of this research, we envision a relevant impact also for process mining future developments. Indeed, differently from the existing process mining techniques, we aim at discovering models of data pipelines that not only include the traditional sequence-flow constructs, but that embed information about the performance of the observed pipelines and the data manipulated by any step of the pipelines. This result would allow us to push forward the research on the discovery of data-aware and object-aware business processes, which is currently at an early stage [28].

As an immediate future work, we aim to validate the model's practical applicability and the effectiveness of process mining to perform BDPD and uncover dark data against the strong selection of complementary business cases involved in the H2020 DataCloud project. This will enable us to mitigate the rather preliminary evaluation presented in this paper, which limits the generalizability of our findings on the effectiveness of process mining (cf. RQ3) only to the data pipeline analyzed in the use case.

Acknowledgments. This work is supported by the H2020 project DataCloud (Grant number 101016835), and the Sapienza project DISPIPE.

References

1. van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. Abb, L., Rehse, J.: A reference data model for process-related user interaction logs. In: 20th International Conference on Business Process Management (BPM 2022) (2022)
3. Acampora, G., Vitiello, A., Di Stefano, B., van der Aalst, W.M.P., Günther, C., Verbeek, E.: IEEE 1849: the XES standard. In: IEEE Computational Intelligence Magazine (2017)
4. Agostinelli, S., Benvenuti, D., De Luzi, F., Marrella, A.: Big data pipeline discovery through process mining: challenges and research directions. In: 1st Italian Forum on Business Process Management, Co-located with BPM 2021 (2021)
5. Augusto, A., et al.: Automated discovery of process models from event logs: review and benchmark. IEEE TKDE **31**(4), 686–705 (2018)
6. Benvenuti, D., Falleroni, L., Marrella, A., Perales, F.: An interactive approach to support event log generation for data pipeline discovery. In: 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC 2022) (2022)
7. Carmona, J., van Dongen, B.F., 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
8. Corallo, A., Crespino, A.M., Vecchio, V.D., Lazoi, M., Marra, M.: Understanding and defining dark data for the manufacturing industry. IEEE Trans. Eng. Manag. **70**(2), 700–712 (2021)
9. van Dongen, B.F., Shabani, S.: Relational XES: data management for process mining. In: CAiSE Forum 2015, pp. 169–176 (2015)
10. Gimpel, G.: Bringing dark data into the light: illuminating existing IoT data lost within your organization. Bus. Horiz. **63**(4), 519–530 (2020)
11. Humayoun, S.R., et al.: Designing mobile systems in highly dynamic scenarios: the WORKPAD methodology. Knowl. Tech. Pol. **22**, 25–43 (2009)
12. Johannesson, P., Perjons, E.: An Introduction to Design Science. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10632-8
13. Leemans, M., Liu, C.: XES Software Telemetry Extension. XES W. Group (2017)

14. de Leoni, M., Marrella, A.: Aligning real process executions and prescriptive process models through automated planning. Exp. Syst. App. **82**, 162–183 (2017)
15. Marrella, A., Mecella, M., Russo, A.: Collaboration on-the-field: suggestions and beyond. In: 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2011) (2011)
16. Marrella, A., Mecella, M., Russo, A.: Featuring automatic adaptivity through workflow enactment and planning. In: 7th International Conference on College Composition: Networking, Applications and Worksharing (CollaborateCom 2011), pp. 372–381. IEEE (2011)
17. Mernik, M., Heering, J., Sloane, A.M.: When and how to develop domain-specific languages. ACM Comput. Surv. **37**(4), 316–344 (2005)
18. Munappy, A.R., Bosch, J., Olsson, H.H.: Data pipeline management in practice: challenges and opportunities. In: International Conference on Product-Focused Software Process Improvement (PROFES 2020), pp. 168–184 (2020)
19. Nikolov, N., et al.: Conceptualization and scalable execution of big data workflows using domain-specific languages and software containers. Internet Things **16**, 100440 (2021)
20. Oleghe, O., Salonitis, K.: A framework for designing data pipelines for manufacturing systems. Procedia CIRP **93**, 724–729 (2020)
21. Pegoraro, M., Uysal, M.S., van der Aalst, W.M.: An XES extension for uncertain event data. arXiv preprint 2204.04135 (2022)
22. Plale, B., Kouper, I.: The centrality of data: data lifecycle and data pipelines. In: Data Analytics for Int. Transportation System. Elsevier (2017)
23. Rabl, T., Jacobsen, H.-A.: Big data generation. In: Rabl, T., Poess, M., Baru, C., Jacobsen, H.-A. (eds.) WBDB -2012. LNCS, vol. 8163, pp. 20–27. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-53974-9_3
24. Rafiei, M., van der Aalst, W.M.P.: Privacy-preserving data publishing in process mining. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNBIP, vol. 392, pp. 122–138. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58638-6_8
25. Roman, D., Prodan, R., Nikolov, N., Soylu, A., Matskin, M., Marrella, A., et al.: Big data pipelines on the computing continuum: tapping the dark data. Computer **55**(11), 74–84 (2022)
26. Schönig, S., Rogge-Solti, A., Cabanillas, C., Jablonski, S., Mendling, J.: Efficient and customisable declarative process mining with SQL. In: 28th International Conference on Advanced Information Systems Engineering (CAiSE 2016) (2016)
27. Schönig, S.: SQL queries for declarative process mining on event logs of relational databases. arXiv preprint 1512.00196 (2015)
28. Steinau, S., Marrella, A., Andrews, K., Leotta, F., Mecella, M., Reichert, M.: DALEC: a framework for the systematic evaluation of data-centric approaches to process management software. Software Syst. Model. **18**(4), 2679–2716 (2019)
29. Syamsiyah, A., van Dongen, B.F., van der Aalst, W.M.P.: DB-XES: enabling process discovery in the large. In: Sixth International Symposium on Data-Driven Process Discovery and Analysis SIMPDA 2016, vol. 1757, pp. 63–77 (2016)
30. Teymourlouei, H., Jackson, L.: Dark data: managing cybersecurity challenges and generating benefits. In: Arabnia, H.R., et al. (eds.) Advances in Parallel & Distributed Processing, and Applications. TCSCI, pp. 91–104. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-69984-0_9

Foundations of Collaborative DECLARE

Luca Geatti[1], Marco Montali[2], and Andrey Rivkin[3](✉)

[1] University of Udine, Udine, Italy
luca.geatti@uniud.it
[2] Free University of Bozen-Bolzano, Bolzano, Italy
montali@inf.unibz.it
[3] Technical University of Denmark, Kgs. Lyngby, Denmark
ariv@dtu.dk

Abstract. Collaborative work processes are widespread, and call for sophisticated modelling techniques to guarantee that the in-focus process is able to suitably handle all the relevant ways in which external, uncontrollable participants can influence the overall behaviour. In the presence of external actors, one needs to distinguish the internal, controllable nondeterminism of the in-focus process from the uncontrollable nondeterminism of external participants. While collaborative processes have been previously studied in the context of declarative processes, where specifications distinguish how different sources of control interact, no study along this line exists in the context of the DECLARE declarative process modeling framework. To this end, we introduce "collaborative DECLARE" (coDECLARE), where activities are assigned to the internal orchestrator or to external participants, and constraints are partitioned into conditions on how the external participants can interact with the in-focus process, and conditions that must be guaranteed by the in-focus process itself, framing the resulting specifications in style of assume-guarantee (behavioral) contracts. We discuss the conceptual and explain how central tasks such as that of DECLARE consistency and enactment have to be revised for coDECLARE. Moreover, we show how the resulting tasks can be encoded into corresponding realisability and reactive synthesis tasks for LTL specifications on finite traces.

Keywords: declarative process modelling · collaborative processes · model analysis · reactive synthesis · LTL on finite traces

1 Introduction

If we consider the very core definition of what a business or work process is, we see that *collaboration* is an essential aspect. In [22], a process is defined as:

> *"a collection of inter-related events, activities and decision points that involve a number of actors and objects, and that collectively lead to an outcome that is of value to at least one customer".*

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 55–72, 2023.
https://doi.org/10.1007/978-3-031-41623-1_4

The definition highlights, in fact, that a process involves a number of actors, of which at least one is external to the organization. This generalizes to multiple external actors that may be involved in the process execution: from providers furnishing input material, resources, and/or services to the organization, to customers and other external stakeholders that are interested in the value produced through the process.

Collaborative work processes are hence widespread. In this work, we consider how collaboration is handled by taking the subjective perspective of one of interacting parties, singling out the process orchestrated by the in-focus party and how it relates to the behavior of other parties. Other approaches are obviously possible: for example, collaboration can be globally captured in a choreography model, from which local processes can be derived.

In the context of service interaction and execution, these aspects have been tackled when formalizing and analyzing collaborative processes [1,5,46], and in the context of realizability for multi-party choreographic processes, considering that they must be enacted in a decentralized way through interaction of the parties, which in turn call for ensuring that they locally have enough visibility of the current state of affairs when required to perform some task [20,36]. More recently, similar notions have been applied to BPMN collaborative processes, sometimes indeed reflecting the two sources of nondeterminism [13], and sometimes instead blurring them [33]. Such a blurring can be distinctively seen in widely employed Petri net encodings of BPMN [21,37], where radically different constructs such as internal decisions, event-based gateways, and boundary events are all captured as free choices in the Petri nets.

A flourishing line of research on this matter exists also for declarative processes. Several approaches have brought forward formal models for collaborative/distributed process specifications, with two main lines respectively focused on Dynamic Condition Response (DCR) Graphs [31,32] and artifact-centric formalisms [4,23]. The common theme of these works is to ensure the overall correct design and executability of distributed, collaborative processes that emerge from the composition of local declarative components. Local processes are typically composed under a cooperative assumption, where their internal non-determinism related to the choices they take is aligned with that of the others.

In our approach, we consider a different setting, where collaboration is in a way approached "cautiously", considering that when the process of an in-focus party interacts with external participants, those external participants enact their own, at least not fully controllable independent processes. This interpretation dates back to seminal contributions on open, reactive systems [30,42], where two sources of non-determinism have to be considered. On the one hand, there is non-determinism resolved by the choices internally taken by the in-focus process; on the other hand, there is non-determinism arising from the external choices picked by the other parties. While the first is within the scope of the local orchestration, and hence can be resolved existentially, the second is uncontrollable, and thus all possibilities have to be taken into account by the process.

This delicate interplay is apparent by looking into de-facto process modelling notations, such as BPMN. There, internal control-flow structures are paired with corresponding structures dealing with external actors and events, leading to constructs such as message exchange, different types of unsolicited events, boundary events attached to exception flows, and event-driven gateways. All such structures are used to indicate routes that are taken by the process not due to internal decisions of its orchestrator, but based on which of the external events (from the set of o available ones) occurs first.

Surprisingly, collaborative processes have never been studied in the context of DECLARE, where constraints are in fact all combined together without distinguishing how they interact with the different sources of control. In fact, DECLARE [40,41] simply defines a process as a monolithic set of activities equipped with a set of constraints composed in a conjunction. No distinction is given based on which parties (or agents) control which actions. Similarly, constraints are not differentiated among those that define the context of execution (e.g., indicating under which conditions external partners can interact with the process), and those that capture the expected behavior of the in-focus process (e.g., expressing what the process should do in response to external stimuli).

In this work, we close this gap by introducing *collaborative* DECLARE (coDECLARE), where activities are assigned to the internal orchestrator or to external participants, and constraints are partitioned into conditions capturing *assumptions* on how the external participants can interact with the in-focus process, and conditions that express *guarantees* provided by the in-focus process. This reflects a peculiar characteristics when dealing with work processes or, more in general, information systems, which makes them different from general reactive systems where the environment is often assumed to be completely uncontrollable. In fact, when external stakeholders interact with a process enacted by a single party, they freely decide which task to select next among the possible ones made available to them. In this sense, the presence of external actors requires the process to define *the context of interaction*, and then to *suitably react to all possible interactions within this context*. On the one hand, the process defines the *assumptions* under which its external parties can send messages/generate events that have to be handled. On the other hand, external actors are uncontrollable, thus calling the in-focus process to be able to *guarantee* a suitable management of all possible situations within the space of possibilities defined by the context.

To handle these features, we resort to the long-standing literature on *realizability and synthesis* for temporal specifications, widely studied in AI and formal methods, and we show how to lift it for defining coDECLARE and solve key tasks related to correctness and enactment. Our main contributions are as follows:

1. We discuss why it is essential in this collaborative setting to distinguish responsibility over activities and separate constraints into assumption and guarantee constraints.
2. We define coDECLARE and discuss how central tasks such as that of DECLARE consistency and enactment have to be redefined in the light of collaborative, declarative processes.

3. We show how consistency and enactment of coDECLARE specifications can be reduced to realizability and synthesis for Linear Temporal Logic over finite traces (LTLf) , for which very mature implementations exist [6,10,27,29,48,50].

The paper is organized as follows. In Sect. 2 we review the necessary background. In Sect. 3 we introduce the framework of collaborative DECLARE. Section 4 formally defines the consistency and enactment for coDECLARE, whereas Sect. 5 shows how those can be effectively checked and computed using the existing reactive synthesis techniques. Conclusions and future directions follow.

2 A Bird-Eye-View on LTLf and DECLARE

We recall the necessary background on LTL on finite traces (LTLf), and how it is used to formalize the DECLARE language.

2.1 LTLf

LTLf is a temporal logic to express properties of finite traces.

Definition 1 (Syntax of LTLf). Given a set Σ of proposition letters, a formula ϕ of LTLf is defined as follows [18]:

$$\phi \coloneqq p \mid \neg p \mid \phi \vee \phi \mid \phi \wedge \phi \mid \mathsf{X}\phi \mid \widetilde{\mathsf{X}}\phi \mid \phi \mathsf{U} \phi$$

where $p \in \Sigma$. Formulas of LTLf over the alphabet Σ are interpreted over *finite traces* (or state sequences, or words), *i.e.*, sequences in the set $(2^{\Sigma})^{+}$. ◁

Intuitively, $\mathsf{X}\phi$ indicates *strong next*, postulating that there must exist a next state in the trace, and ϕ must hold therein. Instead, $\widetilde{\mathsf{X}}\phi$ indicates *weak next*, capturing that if a next state exists, then ϕ must hold therein. Hence, $\mathsf{X}\phi$ evaluates to false in the last state of the trace, while $\widetilde{\mathsf{X}}\phi$ evaluates to true, both regardless of ϕ. Formula $\phi_1 \mathsf{U} \phi_2$ indicates instead *until*, capturing that later in the trace (obviously, before the end of the trace) ϕ_2 must hold, and in all the states between the current one and the one where ϕ_2 holds, ϕ_1 must hold.

In the following, we will write *general finite trace semantics* to denote the interpretation under these structures. Let $\sigma = \langle \sigma_0, \ldots, \sigma_{n-1} \rangle \in (2^{\Sigma})^{+}$ be a finite trace. We define the *length* of σ as $|\sigma| = n$. With $\sigma_{[i,j]}$ (for some $0 \leq i \leq j < |\sigma|$) we denote the subinterval $\langle \sigma_i, \ldots, \sigma_j \rangle$ of σ.

Definition 2 (LTLf satisfaction relation, model, language, equivalence). Given an LTLf formula ϕ over Σ, the *satisfaction relation* of Σ by trace $\sigma \in (2^{\Sigma})^{+}$ at time $0 \leq i < |\sigma|$, denoted by $\sigma, i \models \phi$, is inductively defined as:

- $\sigma, i \models p$ iff $p \in \sigma_i$;
- $\sigma, i \models \neg p$ iff $p \notin \sigma_i$;

- $\sigma, i \models \phi_1 \vee \phi_2$ iff $\sigma, i \models \phi_1$ or $\sigma, i \models \phi_2$;
- $\sigma, i \models \phi_1 \wedge \phi_2$ iff $\sigma, i \models \phi_1$ and $\sigma, i \models \phi_2$;
- $\sigma, i \models \mathsf{X}\phi$ iff $i+1 < |\sigma|$ and $\sigma, i+1 \models \phi$;
- $\sigma, i \models \widetilde{\mathsf{X}}\phi$ iff either $i+1 = |\sigma|$ or $\sigma, i+1 \models \phi$;
- $\sigma, i \models \phi_1 \mathsf{U} \phi_2$ iff there exists $i \leq j < |\sigma|$ such that $\sigma, j \models \phi_2$, and $\sigma, k \models \phi_1$ for all k, with $i \leq k < j$.

Table 1. DECLARE patterns together with their LTLf formalization

Pattern	LTLf **formalization**
`existence`(p)	$\mathsf{F}(p)$
`coexistence`(p,q)	$\mathsf{F}(p) \leftrightarrow \mathsf{F}(q)$
`resp-existence`(p,q)	$\mathsf{F}(p) \rightarrow \mathsf{F}(q)$
`not-coexistence`(p,q)	$\neg(\mathsf{F}(p) \wedge \mathsf{F}(q))$
`absence2`(p)	$\neg\mathsf{F}(p \wedge \mathsf{XF}(p))$
`response`$(p, q_1, \ldots, q_n)$	$\mathsf{G}(p \rightarrow \mathsf{F}(\bigvee_{i=1}^{n} q_i))$
`alt-response`(p,q)	$\mathsf{G}(p \rightarrow \mathsf{X}((\neg p)\mathsf{U}q))$
`chain-response`(p,q)	$\mathsf{G}(p \rightarrow \mathsf{X}(q))$
`precedence`(p,q)	$(\neg q)W(p)$
`alt-precedence`(p,q)	$((\neg q)Wp) \wedge \mathsf{G}(q \rightarrow \widetilde{\mathsf{X}}((\neg q)Wp))$
`chain-precedence`(p,q)	$\mathsf{G}(\mathsf{X}(q) \rightarrow p)$
`succession`(p,q)	$\mathsf{G}(p \rightarrow \mathsf{F}(q)) \wedge (\neg q)W(p)$
`alt-succession`(p,q)	$\mathsf{G}(p \rightarrow \mathsf{X}((\neg p)Uq)) \wedge ((\neg q)Wp) \wedge \mathsf{G}(q \rightarrow \widetilde{\mathsf{X}}((\neg q)Wp))$
`chain-succession`(p,q)	$\mathsf{G}(p \leftrightarrow \mathsf{X}(q))$
`neg-succession`(p,q)	$\mathsf{G}(p \rightarrow \neg\mathsf{F}(q))$
`neg-chain-succession`(p,q)	$\mathsf{G}(p \rightarrow \widetilde{\mathsf{X}}(\neg q)) \wedge \mathsf{G}(q \rightarrow \widetilde{\mathsf{X}}(\neg p))$
`choice`(p,q)	$\mathsf{F}(p) \vert \mathsf{F}(q)$
`exc-choice`(p,q)	$(\mathsf{F}(p) \vert \mathsf{F}(q)) \wedge \neg(\mathsf{F}(p) \wedge \mathsf{F}(q))$

We say that σ is a *model* of ϕ (written as $\sigma \models \phi$) iff $\sigma, 0 \models \phi$. The *language* (over finite trace) of ϕ, denoted by $\mathcal{L}(\phi)$, is the set of traces $\sigma \in (2^{\Sigma})^{+}$ such that $\sigma \models \phi$. We say that two formulas $\phi, \psi \in$ LTLf are *equivalent* iff $\mathcal{L}(\phi) = \mathcal{L}(\psi)$. $\triangleleft$

As customary, we define the following abbreviations: $\mathsf{F}\phi = \top\mathsf{U}\phi$ (eventually) captures that ϕ holds at some moment in the future, $\mathsf{G}\phi = \neg\mathsf{F}\neg\phi$ (globally) captures that ϕ holds from the current state to the end of the trace, and $\phi_1 W \phi_2 = \phi_1 \mathsf{U} \phi_2 \vee \mathsf{G}\neg\phi_2$ weakens U by not necessarily requiring that ϕ_2 becomes true. Also notice that $\widetilde{\mathsf{X}}\phi = \neg\mathsf{X}\neg\phi$.

2.2 DECLARE

DECLARE is a framework [41] and a language [40] for the declarative specification of processes, enjoying flexibility by design [44]. We refer to [39] for a thorough treatment of declarative processes.

A DECLARE specification consists of a finite set of *patterns* used for constraining the allowed execution traces of the process. Each pattern is defined over a set of (atomic) actions, and has a semantics based on LTLf. Table 1 recalls the typical DECLARE patterns and their LTLf formalization.

In the context of this work, we actually support arbitrary patterns in LTLf, going beyond the patterns of Table 1. We therefore directly use LTLf formulas in place of constraint patterns.

Definition 3 (DECLARE specification). A DECLARE *specification* is a pair $\langle \mathcal{A}, \mathcal{C} \rangle$, where $\mathcal{A}$ is a finite set of *activities*, and $\mathcal{C}$ is a finite set of LTLf formulas over $\mathcal{A}$, called *constraints*. $\triangleleft$

DECLARE also comes with a graphical notation. We illustrate next a DECLARE specification, which will be used throughout the entire paper.

Example 1. Consider the DECLARE specification from Fig. 1(a), shown in the standard graphical notation associated to DECLARE. The specification is a fragment of an order-to-cash process. The specification dictates that an order can be paid or canceled at most once (constraint 0..1 on `cancel` and `pay`). Whenever an order is paid, then the customer address has to be set at least once, wither before or after the payment (resp-existence(`pay`,`set`)). Upon payment either shipment or refund should eventually occur (response(`pay`,`ship` $\vee$ `refund`)). In turn, shipment and refund can only occur after payment (precedence(`pay`,`ship`) and precedence(`pay`,`refund`)). Shipment is only possible if the address has been set (precedence(`pay`,`ship`)), and the address cannot be updated anymore once shipment occurs (neg-response(`ship`,`set`)). Finally, shipment and cancelation are mutually exclusive (not-coexistence(`cancel`,`ship`)). $\triangleleft$

Differently from arbitrary LTLf formulas, DECLARE assumes that every state corresponds to the execution of one and only one activity, that is, that exactly one proposition is true therein [16,25]. This leads to a semantics based on so-called *simple finite traces*.

Definition 4 (Simple finite trace). A simple finite trace over $\mathcal{A}$ is a finite trace $\sigma = \langle \sigma_0, \ldots, \sigma_n \rangle \in \mathcal{A}^+$, such that for every $i \in \{0, \ldots, |\sigma - 1|\}$, we have $|\sigma_i| = 1$. $\triangleleft$

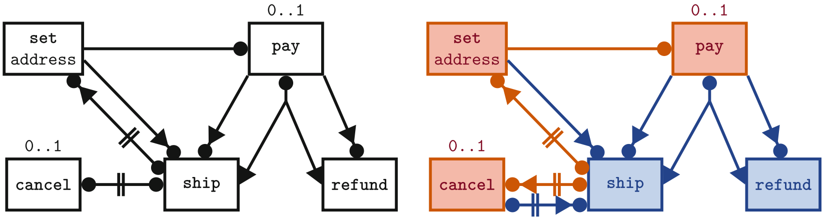
Definition 5 (DECLARE model trace). Given a DECLARE specification $\mathcal{D} = \langle \mathcal{A}, \mathcal{C} \rangle$, a simple trace σ over $\mathcal{A}$ is a *model trace* of $\mathcal{D}$ if $\sigma \models \bigwedge_{\varphi_i \in \mathcal{C}} \varphi_i$. $\triangleleft$

We close by recalling that, without loss of generality, one can redefine the notion of model trace by taking as input an arbitrary LTLf trace, proviso altering the DECLARE specification with an additional special formula forcing the arbitrary trace to be simple.

Remark 1 ([24,39]). Given a DECLARE specification $\mathcal{D} = \langle \mathcal{A}, \mathcal{C} \rangle$, to check wether an arbitrary model trace σ over $\mathcal{A}$ is indeed a *model trace* of $\mathcal{D}$ we check whether $\sigma \models \psi_{simple}(\mathcal{A}) \bigwedge_{\varphi_i \in \mathcal{C}} \varphi_i$, where for a set S of propositions we define $\psi_{simple}(S) \coloneqq \mathsf{G}(\bigvee_{p \in S} p \wedge \bigwedge_{p \neq q \in S} \neg(p \wedge q))$. $\triangleleft$

3 Collaborative DECLARE

We now critically assess the notion of simple traces (Definition 4), and that of DECLARE specifications (Definition 3), in the light of collaborative processes, such as actually the one introduced in Example 1.



(a) A DECLARE specification (b) A coDECLARE refinement of (a)

Fig. 1. A graphical representation of an order handling process in DECLARE (a), then refined in (b) as a coDECLARE specification. Rectangles denote actions, connectors constraints. Customer action s/constraints are in orange, those of the seller in blue. (Color figure online)

3.1 Executing a Collaborative Process

By inspecting Example 1, it is clear that the activities contained in the specification cannot be all ascribed to a single locus of execution. Instead, the process brings together two parties: a *seller* - responsible for orchestrating the order-to-cash process, and a *customer*, representing the external participant. In this light, the first important change we need to apply to the specification, is to actually assign the different activities to one of the two parties.

Example 2. In the order-to-cash process of Fig. 1, activities `set address` and `pay` are controlled by the customer, while `cancel`, `ship`, `refund` are controlled by the seller. ◁

In case of multiple external parties, for the purpose of this paper we can all model them as a single, external party, as it is typically done in the literature [34,43]. In fact, what matters is distinguishing the two different sources of nondeterminism when choosing which activity to execute: the one of the orchestrator, or that of external, uncontrollable actors. From now on, we refer to the orchestrator as *controller*, and to the external parties as *environment*.

Since the environment is uncontrollable, capturing a process execution as a simple trace (in the sense of Definition 4) is overly restrictive. Conceptually speaking, in fact, the process specification is given to controller, and therefore it is in the interest of controller to ensure that the resulting trace satisfies the specification.

In fact, controller has to *observe* what environment has done so far, so as to *suitably react*. For example, the seller may decide to behave differently depending on whether the customer has paid and later cancelled or not. This calls for moving from the notion of simple trace to that of *process strategy* for controller.

Definition 6 (Process strategy). Let $\mathcal{A}^E$ and $\mathcal{A}^C$ be two disjoint sets of activities, respectively denoting the environment and the controller activities. A *process strategy* is a function $s : (E)^+ \rightarrow P$ that for every finite sequence $\boldsymbol{e} = \langle e_0, \ldots, e_n \rangle$ of activities chosen by the environment, determines the next activity $P_n = s(\boldsymbol{e})$ executed by the controller. ◁

Notice that a process strategy respects the original DECLARE semantics of simple traces, as it enables only one activity to be executed per state.

One may wonder why the process strategy does not use the whole partial trace accumulated so far, including the activities of both the environment and the controller, but instead merely focuses on previous activities executed by environment. As we will see, more sophisticated notions of process strategies will not be needed to solve the key tasks of collaborative processes.

3.2 Satisfying the Constraints of a Collaborative Process

Now that we have a notion of process strategy, we can imagine that the orchestrator uses it to guarantee that the constraints of the DECLARE specification of interest are indeed all satisfied when concluding the execution (that is, the strategy yields, at completion time, a model trace). However, considering that the environment is uncontrollable, it turns out that this is impossible even for very trivial DECLARE specifications that assign at least one activity to environment.

Example 3. Consider the DECLARE specification of Fig. 1(a), with the activity partitioning from Example 2. The seller cannot define a process strategy that guarantees the satisfaction of all constraints, since the customer controls activities that are subject to unary constraints (0..1 on `cancel` and `pay`) or binary constraints all related to the customer (resp-existence(`pay`,`set`)). Since the customer is uncontrollable, they may issue two payments, or two cancelations, or may pay without ever setting an address. ◁

Example 3 witnesses that we cannot monolithically consider all specification constraints as being under the responsibility of the controller, while leaving environment free to generate an arbitrary sequence of activities under their responsibility. In fact, as stated in the introduction, external parties need to come with some context on their freedom of choice. This is concretely reflected inside BPM systems, which expose executable activities to the external parties, and define how to handle external events, only under certain circumstances.

To reflect this requirement in our declarative setting, we give constrained freedom to the environment, by partitioning the constraints into those that must be respected by the environment to properly interact with the process, and those that the controller has to satisfy. This reflects the paradigm of *assume-guarantee contracts* [7]: under the *assumption* that the environment behaves in such a way that their constraints are satisfied, the orchestrator is bound to *guarantee* the satisfaction of their own constraints.

With this intuition in mind, we define collaborative DECLARE specifications.

Definition 7 (coDECLARE specification). A *collaborative* DECLARE (coDECLARE) specification is a tuple $\mathcal{D} = \langle \mathcal{A}^E, \mathcal{A}^C, \mathcal{C}^E, \mathcal{C}^C \rangle$, where:

- $\mathcal{A}^E$ is a finite set of *environment activities*;
- $\mathcal{A}^C$ is a finite set of *controller activities*, with $\mathcal{A}^E \cap \mathcal{A}^C = \emptyset$;

- $\mathcal{C}^E$ is a set of LTLf constraints over $\mathcal{A}^E \cup \mathcal{A}^C$ representing the *environment constraints*;
- $\mathcal{C}^C$ is a set of LTLf constraints over $\mathcal{A}^E \cup \mathcal{A}^C$ representing the *controller constraints*;

We respectively call $\mathcal{D}^a = \langle \mathcal{A}^E \cup \mathcal{A}^C, \mathcal{C}^E \rangle$ and $\mathcal{D}^g = \langle \mathcal{A}^E \cup \mathcal{A}^C, \mathcal{C}^C \rangle$ the *assumption* and *guarantee specifications* of $\mathcal{D}$. $\lhd$

Example 4. We refine the DECLARE specification of Fig. 1(a), considering the activity partitioning from Example 2, into the coDECLARE specification $\mathcal{D}_{order}$ of Fig. 1(b). The partitioning is done according to the following idea. To participate to the order handling process, the customer has to commit to: (i) paying and cancelling at most once, (ii) ensuring that an address is set upon payment, (iii) not updating the address nor cancelling once the seller has shipped the order. At the same time, the seller commits to: (i) shipping only if the customer sets an address and pays, (ii) refunding only if a previous customer payment exists, (iii) ensuring that shipment or refund occur whenever the customer pays, (iv) not shipping if the customer cancels the order. One can see that the vast majority of constraints present in the original DECLARE specifications have been maintained and assigned either to the environment (customer) or the controller (seller). The only exception is the not coexistence constraint relating cancelation and shipment, which is now refined into two time-oriented constraints, one assuming that the customer does not cancel after a seller's shipment (neg-response(`ship`, `cancel`)), and the other guaranteeing that the seller does not ship after a customer's cancelation ((neg-response(`cancel`, `ship`)). $\lhd$

We can directly lift the notion of model trace (as per Definition 5) to the case of coDECLARE. To do so, we formalize the intuition given so far: the trace must be such that whenever it satisfies the assumption on the environment, then it must satisfy the guarantee on the orchestrator. This implicitly means that executions violating the assumption on the environment are all considered model traces, as these are traces over which the environment cannot claim any guarantee. This is in line with the notion of assume-guarantee contract [7], and that of assume-guarantee synthesis [8,11,38].

Definition 8 (coDECLARE model trace). Given a coDECLARE specification $\mathcal{D} = \langle \mathcal{A}^E, \mathcal{A}^C, \mathcal{C}^E, \mathcal{C}^C \rangle$, a simple trace σ over $\mathcal{A}^E \cup \mathcal{A}^C$ is a *model trace* of $\mathcal{D}$ if, whenever it is a model trace for $\mathcal{D}^a$ (in the sense of Definition 5), then it is also a model trace for $\mathcal{D}^g$ (again in the sense of Definition 5). $\lhd$

Example 5. Consider the coDECLARE specification $\mathcal{D}_{order}$ of Fig. 1(b). Four model traces for $\mathcal{D}_{order}$ are $\langle$`set`, `pay`, `ship`$\rangle$, $\langle$`pay`, `set`, `refund`$\rangle$, $\langle$`pay`, `set cancel refund`$\rangle$, and $\langle$`pay set ship cancel`$\rangle$. They respectively denote a good execution where the order is shipped, an execution where the seller decides to refund for an internal problem, and an execution where the seller refunds due to a customer cancellation, and one where no refund is given in spite of cancellation since the order has been already shipped. The trace $\langle$`pay set cancel`$\rangle$ is instead not a model trace, as the seller should refund. $\lhd$

The main open question now is: how can the orchestrator ensure that an ongoing execution for a coDECLARE model eventually leads to a proper, model trace? The challenge here is that, even if the environment is constrained by their assumption specification, it still has a (constrained) freedom to decide which environment activities are executed, and in which order. Hence, to be able to properly execute the specification, the orchestrator must have a strategy (in the sense of Definition 6) to guarantee that no matter how the environment behaves within the space given by the assumption specification, then the execution is progressed and finally stopped by satisfying the guarantee specification. This is tackled in the next section.

4 Consistency and Enactment of coDECLARE

To tackle the problem of enactment of coDECLARE specifications, we start pointing out the striking similarity with the long-standing problem of *synthesis from declarative specification*, which dates back to Church [12]. In summary, given a declarative specification, one can define two distinct problems. The first concerns *verification*, and is about checking the correctness of the specification, namely whether the specification has a satisfying assignment (which, in the case of linear temporal specifications, means a trace). A different problem is that of *synthesis*, which deals with deriving a correct-by-construction program (in the shape, *e.g.*, of a Mealy or Moore machine, I/O-transducer, or circuit) that realizes the specification and makes it possible to execute it. Extensive research has been conducted on different synthesis settings, considering in particular closed and *open* (also called *reactive*) systems, starting from the seminal contributions by Harel, Pnueli and Rosner [30,42]. In the reactive setting, using the same terminology adopted here, the system (referred to as controller) interacts with an uncontrollable environment, which, in turn, can affect the behavior of the controller. Reactive synthesis is hence modeled as a two-player game between Controller, whose aim is to satisfy the formula, and Environment, who tries to violate it. The objective of the synthesis task is then to synthesize a program for Controller indicating which actions the Controller should take to guarantee the satisfaction of the declarative specification of interest, no matter what are the actions taken by Environment. This problem was originally studied in [12] and solved in [9], and for LTL specifications in particular it was shown to be 2EXPTIME-complete [43,45]. The high theoretical complexity and practical infeasibility of the original approach, led to a plethora of studies focused on settings more amenable to effective synthesis algorithms, one of the most important being synthesis for LTLf- thus considering *finite traces* [28].

In this section, we connect such long-standing literature with coDECLARE, in particular defining consistency and (automatic) enactment for coDECLARE by adapting to our setting the well-established notions of realizability and synthesis from declarative specifications.

4.1 Realizability over Simple Traces

We start by considering the *realizability* task [43] for LTLf formulas, in the setting where executions correspond to simple traces (cf. Definition 4). Intuitively, given an LTLf formula ϕ over two sets of controllable $\mathcal{C}$ and uncontrollable $\mathcal{U}$ variables (s.t. $\mathcal{C} \cap \mathcal{U} = \emptyset$), we have that ϕ is realizable if there exists a strategy for controller that, no matter the choices made by the environment regarding the variables in $\mathcal{U}$ to set true, chooses truth assignments to variables in $\mathcal{C}$ so that ϕ is satisfied. Hereinafter, we talk about variables in the context of general definitions, and activities in the context of coDECLARE and processes, and use $\mathcal{A}^E$ and $\mathcal{A}^C$ as uncontrollable and controllable variables, respectively.

To adapt realizability to our setting, we use process strategies from Definition 6. Since realizability is usually tested using a two-player game between the controller and environment, we postulate that such strategies are applied in the strictly alternating way, and that the environment always starts first. These assumptions will be clarified later.

Definition 9 (Realizability over simple traces). Let ϕ be an LTLf formula over $\mathcal{A}$. ϕ is *realizable over simple finite traces* iff there exists a process strategy $s : (\mathcal{U})^+ \to \mathcal{C}$ such that, for any infinite sequence $\mathcal{U} = \langle \mathcal{U}_0, \mathcal{U}_1, \ldots \rangle \in (\mathcal{U})^\omega$ of actions chosen by the environment, there exists $k \in \mathbb{N}$ such that $\mathtt{simres}(s, \mathcal{U})_{[0,k]} \models \phi$.[1] $\triangleleft$

First and foremost, notice that the way the process strategy starts and completes is perfectly compatible with the notion of a business process. On the one hand, every process instance starts because of an activity triggered by the environment. On the other hand, the power to decide when an execution should be stopped is of the controller: it is in fact the internal orchestration mechanism that defines when a process instance reaches a final state. We now comment on strict alternation. First, the fact that every step comes with just a single chosen activity is in line with the notion of simple trace. Second, imposing alternation does not incur in any loss of generality, as we can equip both actors with a *no-op activity* whose purpose is simply to relinquish control back to the other actor. In our running example (cf. 1(b)), this is for example useful for controller to wait that the customer sets their address when the customer indeed triggers a payment without a prior execution of `set`.

4.2 Consistency and Orchestration

To ensure that a coDECLARE specification is consistent, we need to ensure that controller can define a strategy that yields a model trace – as per Definition 8. Considering Definition 10, we thus get the following.

[1] Here, $\mathtt{simres}(s, \mathcal{U}) = \langle \mathcal{U}_0, s(\langle \mathcal{U}_0 \rangle), \mathcal{U}_1, s(\langle \mathcal{U}_0, \mathcal{U}_1 \rangle), \ldots \rangle$ is the state sequence resulting from the strict alternation between the choices made by the environment and those made by the strategy s.

Definition 10 (Consistency).
A coDECLARE specification $\mathcal{D} = \langle \mathcal{A}^E, \mathcal{A}^C, \mathcal{C}^E, \mathcal{C}^C \rangle$ is *consistent* if the LTLf formula $\bigwedge_{\varphi_i \in \mathcal{C}^E} \varphi_i \rightarrow \bigwedge_{\psi_j \in \mathcal{C}^C} \psi_j$ is realizable over simple finite traces. ◁

A process strategy witnessing consistency can be effectively seen as an *orchestration mechanism* for controller: it defines a specific behaviour for controller ensuring that, whenever environment behaves in accordance to the assumption specification, the resulting reactions yield a simple trace satisfying the guarantee specification. Obviously, for a consistent specification, many different process strategyes may exist, resulting in different orchestration mechanisms for the controller. We show this in our running example.

Example 6. Consider the coDECLARE specification $\mathcal{D}_{order}$ from Fig. 1(b). Multiple process strategies exist for the seller. We show two. The first is an *always refund* strategy:

- The seller simply generates a no-op, unless the customer pays.
- As soon as the customer pays, the seller immediately reacts by refunding.

The second is a *ship as soon as possible* strategy:

- The seller simply generates a no-op, unless the customer pays.
- If the customer pays, the seller checks whether the customer has already set an address. If so, then the seller immediately ships. If not, then the seller waits for further activities executed by the seller. In particular, since the customer operates under the assumption that an address must eventually be set:
  - if the customer sets the address, the seller immediately ships afterwards;
  - if the customer cancels and only later sets the address, the seller reacts to the cancelation by refunding.

Obviously, many other process strategies exist. For example, *ship as soon as possible* may be turned into a more cautious strategy where, instead of immediately shipping whenever there are the conditions for doing so, seller waits for a while to see whether customer intends to cancel. ◁

Example 6 may show that some of the process strategies for the controller (such as the *always refund* strategy) are unintended. This should not be seen as a technical limitation of our approach, but rather as the same issue of typical *under-specification* problems arising in standard DECLARE, a well-known problem that actually pervades declarative modelling languages in general. In fact, additional constraints could be added to cut off some unintended process strategies for the controller, as discussed next.

Example 7. Consider Example 6. We may want to ensure that the seller cannot use *always refund* as an orchestration mechanism. A possible way to do so would be to constrain that the seller only refunds upon an explicit cancelation of the customer, in particular when this is triggered after a payment. This could be done by adding to the guarantee specification a further constraint resp-existence(refund, cancel). Interestingly, in every possible execution strategy for seller, this constraint will be interpreted as the more restrictive constraint precedence(cancel, refund); in fact, since cancel is under the control of the customer, the seller can only guarantee to satisfy resp-existence(refund, cancel) by first waiting that the customer indeed cancels, as refunding before a cancellation may lead to a violation of the constraint (if the customer decides not to cancel, which they can legitimately do). $\triangleleft$

5 Encoding into LTLf Realizability

In this section, we show how coDECLARE consistency can be checked using the standard decision procedure from the literature on LTLf realizability [19], also using the same technique to extract actual process strategies for orchestration.

To do so, we have to resolve a mismatch between the definition of realizability over simple traces (in the sense of Definition 10), and the general notion of LTLf realizability. The mismatches are that in LTLf realizability: a there is no simple trace semantics, and thus strategies are deciding on sequences of sets of actions; b strict alternation is not required and there is no need to secure exclusive control of only one player over a state in the game runs.

Definition 11 (LTLf strategy, realizability [19]**).** Given sets $\mathcal{C}$ and $\mathcal{U}$ as in the previous section, a *strategy* is a function $s : (2^{\mathcal{U}})^+ \rightarrow 2^{\mathcal{C}}$. An LTLf formula ϕ over $\mathcal{U} \cup \mathcal{C}$ is *realizable* if there is a strategy s s.t. for every infinite sequence $\mathcal{U} = \langle \mathcal{U}_0, \mathcal{U}_1, \ldots \rangle \in (2^{\mathcal{U}})^{\omega}$, there exists $k \in \mathbb{N}$ s.t. $\mathtt{res}(s, \mathcal{U})_{[0,k]} \models \phi$, where $\mathtt{res}(s, \mathcal{U}) = \langle \mathcal{U}_0 \cup s(\langle \mathcal{U}_0 \rangle), \mathcal{U}_1 \cup s(\langle \mathcal{U}_0, \mathcal{U}_1 \rangle), \ldots \rangle$ is the trace resulting from reacting to $\mathcal{U}$ according to s. $\triangleleft$

Given an LTLf formula ϕ, the decision procedure for checking realizability of ϕ performs the following steps [19]. First, build a non-deterministic finite automaton § for ϕ and determinise it. Then, play a reachability game using § as the arena so as to check whether the process can reach a final state of the automaton (see [34] for more details on the actual procedure for this step). If this is the case, then ϕ is realizable, otherwise it is not.

Notice that the first step, in the worst case, takes double exponential time: § has at most exponentially many states [19] and the standard subset construction algorithm for determinization of § will require potentially lead to exponential blow-up in the number of states of the resulting deterministic automaton. The reachability game can be solved in polynomial time in the size of automaton [14] and thus does not affect the overall complexity. Notice also that if ϕ is shown to be realizable, then a strategy s can be extracted [34]. In practice, using various heuristics, the first step can be computed efficiently for DECLARE [15,47].

An important step towards reducing the consistency check (and in turn synthesis of process strategies for orchestration) to realizability (and synthesis) of LTLf over finite traces is in enforcing the assumptions made in Sect. 4 on the two-player game used for realizability checking. To this end, we define the following auxiliary LTLf formulas, making sure that (i) the game of the play is a simple trace; (ii) the environment plays at all even states; (iii) the controller plays at all odd states. The formulas are as follows:

$$\psi_{env}(\mathcal{U}) \coloneqq \bigvee_{u\in\mathcal{U}} u \wedge \mathsf{G}(\bigvee_{u\in\mathcal{U}} u \rightarrow (\bigwedge_{u\neq u'\in\mathcal{U}} \neg(u\wedge u') \wedge \widetilde{\mathsf{X}}(\bigwedge_{u\in\mathcal{U}} \neg u \wedge \widetilde{\mathsf{X}} \bigvee_{u\in\mathcal{U}} u)))$$

$$\psi_{con}(\mathcal{C}) \coloneqq \bigwedge_{c\in\mathcal{C}} \neg c \wedge \mathsf{G}(\bigwedge_{c\in\mathcal{C}} \neg c \rightarrow \widetilde{\mathsf{X}}(\bigvee_{c\in\mathcal{C}} c \wedge \bigwedge_{c\neq c'\in\mathcal{C}} \neg(c\wedge c') \wedge \widetilde{\mathsf{X}} \bigwedge_{c\in\mathcal{C}} \neg c)))$$

We use here the weak next (i.e., $\widetilde{\mathsf{X}}$) operator to ensure that the two-player game used for checking the realizability of these formulas can stop at any iteration.

The following theorem from [26] shows how we can recast the realizability technique discussed above to the case if coDECLARE specifications.

Theorem 1 **([26]).** *Let $\mathcal{D} = \langle \mathcal{A}^E, \mathcal{A}^C, \mathcal{C}^E, \mathcal{C}^C\rangle$ be a* coDECLARE *specification. It holds that $\mathcal{D}$ is consistent iff the* LTLf *formula $\psi_{simple}(\mathcal{A}^E \cup \mathcal{A}^C) \wedge \psi_{con}(\mathcal{A}^C) \wedge ((\psi_{env}(\mathcal{A}^E) \wedge \mathcal{C}^E) \rightarrow \mathcal{C}^C)$ is realizable.* ◁

Tool Support. The formula from Theorem 1 can be given as input to the realizability algorithm discussed above. This makes it possible to use any off-the-shelf tool for LTLf synthesis for the purpose of coDECLARE consistency and orchestration. This paves the way towards a direct implementation of our approach. In fact, since the introduction of the LTLf reactive synthesis problem [19], several optimized tools have been developed for solving this problem. Among all, we mention the `Syft` [51] and the `Cynthia` tools [17]. LTLf reactive synthesis is an active area of research: this is witnessed also by the organization of an annual competition (SYNTCOMP [34,35]). We thus expect that the practical efficiency of LTLf synthesis tools reflect also to coDECLARE enactment.

6 Conclusions

We have introduced a novel framework to capture *collaborative, declarative processes* specified in DECLARE, where the process orchestrator (i.e., the controller) must be able to suitably handle the interaction with uncontrollable external parties (i.e., the environment). We have described how this framework, called coDECLARE, can be naturally framed in an *assume-guarantee* style, where the following behavioral contract is stipulated by the controller and environment: under the assumption that the environment behaves according to an *assumption* DECLARE specification, the controller ensures to react by satisfying a *guarantee* DECLARE specification. We have shown, both foundationally and in terms of algorithmic support, how this framework can be connected to the well-studied framework of realizability and synthesis for LTLf specifications.

The natural, next step is to leverage this connection and provide a proof-of-concept implementation for consistency checking and process strategy generation for orchestration in coDECLARE, calling different back-end LTLf realizability/synthesis tools. We are also studying that when the LTLf formulas of interest have the shape of DECLARE patterns, better complexity bounds for solving these problems can be obtained [26]. We also foresee three interesting foundational lines of research starting from the basis provided here. The first concerns the definition of variants of consistency/orchestration for coDECLARE, in the case where the overall specification turns out to be unrealizeable. To this end, so-called best-effort strategies have been introduced [2]. However, while in our setting assume-guarantee specifications can be treated by constructing an implication formula, this is not true anymore in the case of best-effort strategies, and more sophisticated notions of synthesis under assumption have to be studied [3]. The second line concerns the actual process strategies generated from a coDECLARE specification. As discussed in the paper, each strategy defines *a particular way* for controller to ensure that whenever the environment behaves according to the assumption specification, then the guarantee specification is satisfied. In a BPM context, it would be definitely of interest to lift synthesis to a more general orchestration mechanism, where all possible strategies are combined, allowing the controller to decide at runtime, step-by-step, which specific strategy to follow. This is akin to maximally permissive strategies [49], but novel research is needed to represent them by natively dealing with concurrency and other typical control-flow patterns. A last line is to consider different notions of collaboration when dealing with DECLARE specifications. A significant setting, which departs from the one tackled here, is that where collaboration is approached in a choreographic way, observing interaction from an external point of view, and considering all the interacting parties are equally standing. This was partly studied in some seminal works [39, Chapter 8], [40], but not further developed so far.

Acknowledgements. This research has been partially supported by the UNIBZ project ADAPTERS and PRIN MIUR project PINPOINT Prot. 2020FNEB27.

References

1. van der Aalst, W.M.P., Lohmann, N., Massuthe, P., Stahl, C., Wolf, K.: Multiparty contracts: agreeing and implementing interorganizational processes. Comput. J. **53**(1), 90–106 (2010)
2. Aminof, B., De Giacomo, G., Rubin, S.: Best-effort synthesis: doing your best is not harder than giving up. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI), pp. 1766–1772. ijcai.org (2021)
3. Aminof, B., Giacomo, G.D., Murano, A., Rubin, S.: Planning under LTL environment specifications. In: Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling (ICAPS), pp. 31–39. AAAI Press (2019)
4. Badouel, É., Hélouët, L., Kouamou, G.E., Morvan, C.: A grammatical approach to data-centric case management in a distributed collaborative environment. In: Wainwright, R.L., Corchado, J.M., Bechini, A., Hong, J. (eds.) Proceedings of SAC 2015, pp. 1834–1839. ACM (2015)
5. Baldoni, M., Baroglio, C., Martelli, A., Patti, V.: A priori conformance verification for guaranteeing interoperability in open environments. In: Dan, A., Lamersdorf,

W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 339–351. Springer, Heidelberg (2006). https://doi.org/10.1007/11948148_28
6. Bansal, S., Li, Y., Tabajara, L.M., Vardi, M.Y.: Hybrid compositional reasoning for reactive synthesis from finite-horizon specifications. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 9766–9774. AAAI Press (2020)
7. Benveniste, A., et al.: Contracts for System Design. Now Foundations and Trends (2018)
8. Bloem, R., Jobstmann, B., Piterman, N., Pnueli, A., Saar, Y.: Synthesis of reactive (1) designs. J. Comput. Syst. Sci. **78**(3), 911–938 (2012)
9. Buchi, J.R., Landweber, L.H.: Solving sequential conditions by finite-state strategies. In: Mac Lane, S., Siefkes, D. (eds) The Collected Works of J. Richard Büchi. Springer, New York (1990). https://doi.org/10.1007/978-1-4613-8928-6_29
10. Camacho, A., Baier, J.A., Muise, C.J., McIlraith, S.A.: Finite LTL synthesis as planning. In: Twenty-Eighth International Conference on Automated Planning and Scheduling (ICAPS 2018), pp. 29–38. AAAI Press (2018)
11. Chatterjee, K., Henzinger, T.A.: Assume-guarantee synthesis. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 261–275. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71209-1_21
12. Church, A.: Logic, arithmetic, and automata. In: Proceedings of International Congress of Mathematicians. vol. 1962, pp. 23–35 (1962)
13. Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F., Vandin, A.: A formal approach for the analysis of BPMN collaboration models. J. Syst. Softw. **180**, 111007 (2021)
14. De Alfaro, L., Henzinger, T.A., Kupferman, O.: Concurrent reachability games. Theor. Comput. Sci. **386**(3), 188–217 (2007)
15. De Giacomo, G., De Masellis, R., Maggi, F.M., Montali, M.: Monitoring constraints and metaconstraints with temporal logics on finite traces. ACM Trans. Soft. Eng. and Method. **31**(4), 1–44 (2022)
16. 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 (AAAI), pp. 1027–1033. AAAI Press (2014)
17. De Giacomo, G., Favorito, M., Li, J., Vardi, M.Y., Xiao, S., Zhu, S.: LTLF synthesis as and-or graph search: Knowledge compilation at work. In: Thirty-First International Joint Conference on Artificial Intelligence (IJCAI) (2022)
18. De Giacomo, G., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. Association for Computing Machinery IJCAI 2013, pp. 854–860. IJCAI/AAAI (2013)
19. De Giacomo, G., Vardi, M.Y.: Synthesis for LTL and LDL on finite traces. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (AAAI), pp. 1558–1564. AAAI Press (2015)
20. Decker, G., Weske, M.: Local enforceability in interaction petri nets. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 305–319. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_22
21. Dijkman, R.M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in BPMN. Inf. Softw. Technol. **50**(12), 1281–1294 (2008)
22. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, Second Edition. Springer (2018). https://doi.org/10.1007/978-3-662-56509-4

23. Eshuis, R., Hull, R., Sun, Y., Vaculín, R.: Splitting GSM schemas: a framework for outsourcing of declarative artifact systems. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 259–274. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_22
24. Fionda, V., Greco, G.: LTL on finite and process traces: complexity results and a practical reasoner. J. Artif. Intell. Res. **63**, 557–623 (2018)
25. Fionda, V., Guzzo, A.: Control-flow modeling with declare: behavioral properties, computational complexity, and tools. IEEE Trans. Knowl. Data Eng. **32**(5), 898–911 (2019)
26. Geatti, L., Montali, M., Rivkin, A.: Reactive synthesis for DECLARE via symbolic automata. CoRR abs/2212.10875 (2022)
27. Giacomo, G.D., Favorito, M., Li, J., Vardi, M.Y., Xiao, S., Zhu, S.: LTLF synthesis as AND-OR graph search: Knowledge compilation at work. In: Thirty-First International Joint Conference on Artificial Intelligence (IJCAI 2022), pp. 2591–2598. ijcai.org (2022)
28. Giacomo, G.D., Masellis, R.D., Montali, M.: Reasoning on LTL on finite traces: Insensitivity to infiniteness. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 1027–1033. AAAI Press (2014)
29. Giacomo, G.D., Stasio, A.D., Vardi, M.Y., Zhu, S.: Two-stage technique for LTLf synthesis under LTL assumptions. In: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR 2020), pp. 304–314 (2020)
30. Harel, D., Pnueli, A.: On the development of reactive systems. In: Apt, K.R. (eds) Logics and Models of Concurrent Systems. NATO ASI Series, vol 13, pp. 477–498. Springer, Berlin, Heidelberg (1985). https://doi.org/10.1007/978-3-642-82453-1_17
31. Hildebrandt, T.T., Mukkamala, R.R., Slaats, T.: Designing a cross-organizational case management system using dynamic condition response graphs. In: 2011 IEEE 15th International Enterprise Distributed Object Computing Conference (EDOC 2011), pp. 161–170. IEEE Computer Society (2011)
32. Hildebrandt, T.T., Slaats, T., López, H.A., Debois, S., Carbone, M.: Declarative choreographies and liveness. In: Pérez, J.A., Yoshida, N. (eds.) FORTE 2019. LNCS, vol. 11535, pp. 129–147. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21759-4_8
33. Houhou, S., Baarir, S., Poizat, P., Quéinnec, P.: A first-order logic semantics for communication-parametric BPMN collaborations. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 52–68. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_6
34. Jacobs, S., et al.: The first reactive synthesis competition. Int. J. Soft. Tools for Tech. Transf. **19**(3), 367–390 (2017)
35. Jacobs, S., et al.: The reactive synthesis competition (syntcomp): 2018–2021. arXiv preprint arXiv:2206.00251 (2022)
36. Lohmann, N., Wolf, K.: Realizability is controllability. In: Laneve, C., Su, J. (eds.) WS-FM 2009. LNCS, vol. 6194, pp. 110–127. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14458-5_7
37. Lomidze, G., Schuster, D., Li, CY., van Zelst, S.J.: Enhanced transformation of bpmn models with cancellation features. In: Almeida, J.P.A., Karastoyanova, D., Guizzardi, G., Montali, M., Maggi, F.M., Fonseca, C.M. (eds) Enterprise Design, Operations, and Computing. EDOC 2022. Lecture Notes in Computer Science. vol 13585, pp. 128–144. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17604-3_8

38. Maoz, S., Sa'ar, Y.: Assume-guarantee scenarios: semantics and synthesis. In: France, R.B., Kazmeier, J., Breu, R., Atkinson, C. (eds.) MODELS 2012. LNCS, vol. 7590, pp. 335–351. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33666-9_22
39. Montali, M.: Specification and Verification of Declarative Open Interaction Models. LNBIP, vol. 56. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14538-4
40. Montali, M., Pesic, M., van der Aalst, W.M.P., Chesani, F., Mello, P., Storari, S.: Declarative specification and verification of service choreographies. ACM Trans. Web **4**(1), 1–62 (2010)
41. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC), pp. 287–300. IEEE Computer Society (2007)
42. Pnueli, A., Rosner, R.: On the synthesis of a reactive module. In: Proceedings of the 16th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), pp. 179–190. ACM Press (1989)
43. Pnueli, A., Rosner, R.: On the synthesis of an asynchronous reactive module. In: Ausiello, G., Dezani-Ciancaglini, M., Della Rocca, S.R. (eds.) ICALP 1989. LNCS, vol. 372, pp. 652–671. Springer, Heidelberg (1989). https://doi.org/10.1007/BFb0035790
44. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods. Springer, Technologies (2012)
45. Rosner, R.: Modular synthesis of reactive systems. Ph.D. thesis, PhD thesis, Weizmann Institute of Science (1992)
46. Stahl, C., Wolf, K.: Deciding service composition and substitutability using extended operating guidelines. Data Knowl. Eng. **68**(9), 819–833 (2009)
47. 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
48. Xiao, S., Li, J., Zhu, S., Shi, Y., Pu, G., Vardi, M.Y.: On-the-fly synthesis for LTL over finite traces. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 6530–6537. AAAI Press (2021)
49. Zhu, S., De Giacomo, G.: Synthesis of maximally permissive strategies for ltlf specifications. In: Thirty-First International Joint Conference on Artificial Intelligence (IJCAI), pp. 2783–2789. ijcai.org (2022)
50. Zhu, S., Tabajara, L.M., Li, J., Pu, G., Vardi, M.Y.: A symbolic approach to safety LTL synthesis. In: HVC 2017. LNCS, vol. 10629, pp. 147–162. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70389-3_10
51. Zhu, S., Tabajara, L.M., Li, J., Pu, G., Vardi, M.Y.: Symbolic LTLf synthesis. arXiv preprint arXiv:1705.08426 (2017)

Declarative Choreographies with Time and Data

Thomas T. Hildebrandt[1], Hugo A. López[2](✉), and Tijs Slaats[1]

[1] Department of Computer Science, Copenhagen University, København, Denmark
{hilde,slaats}@di.ku.dk

[2] DTU Compute, Technical University of Denmark, Lyngby, Denmark
hulo@dtu.dk

Abstract. Choreographic models express coordination between business roles, in contrast to standard process models that merge local control flow and communication between parties. A choreography is realizable, also known as endpoint projectable, if the independent behaviour of each role in composition with other roles, only behaves exactly in the same way as described in the choreography. We introduce a novel choreographic language expressing safety and liveness properties, incorporating multi-perspective constraints in communication flows, data, and time. This language builds upon recent developments in declarative choreographies using the Dynamic Condition Response (DCR) graph formalism and extends it to accommodate data and time. The interaction between multiple dimensions can render models unrealizable, so we determine the conditions required for realizability through causal relationships within multi-perspective declarative choreographies. This way, realizable choreographies are guaranteed freedom of conflicts in the message exchanges that otherwise can lead to deadlocks.

Keywords: Choreographies · Declarative Models · Multi-Perspective Process Modelling

1 Introduction

The modelling and implementation of cross-organizational business processes introduce additional complexity compared to intra-organizational processes, and each organization needs to agree on a protocol defining the ordering of message exchanges. *Choreographies* provide a way to describe how participants coordinate their interactions in a cross-organizational business process from a global vantage point, by making interactions and not actions the basic building blocks. In this way, choreographies can be viewed as a type of contract between two or more organizations [22], which makes sure that every participant agrees on the protocol for exchanging messages.

However, despite this apparent advantage, choreographies have not reached the same level of adoption as the use of collaboration diagrams, where each

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 73–89, 2023.
https://doi.org/10.1007/978-3-031-41623-1_5

participant is modelled individually. In this work, we will focus on studying 3 limitations of classical choreographies as defined in the BPMN choreography language [22]: adequacy, expressiveness and realizability. First, BPMN, choreographies are imperative: they define an explicit control flow that might not be the most adequate for all types of interaction protocols, e.g. if flexibility in the order of execution is wanted or if one indeed wants to specify a contract, where declarative rules are often more common. Second, protocols in distributed systems often depend both on data and time, but as also pointed out in [23], the BPMN choreography language specification is vague about dependencies on timing constraints, which means that one can not immediately use the notation for formal reasoning about timed systems. Indeed, we are not aware of any formalisation of BPMN choreographies that takes time and data into account. Finally, the real value of choreographies is their realizability. The protocol described by the global view should be implementable as communicating endpoint processes [20]. As also described in [22], realizability requires some additional constraints on the control flow, making sure that an endpoint is not expected to perform a decision based on non-available data.

In earlier work [16] we presented how to use the declarative Dynamic Condition Response (DCR) graphs [13] formalism as a choreography language by using interactions as labels instead of actions, thereby addressing the adequacy limitation of BPMN Choreographies for flexible processes or the specification of contracts. We also provided sound and effectively computable criteria for the realizability of DCR Choreographies. Choosing DCR is not incidental: out of the existing declarative notations (e.g. Declare), DCR is a notation supported by industries, with multiple design and simulation tools, the notation has been employed in major industrial case management systems used in the public sector in Denmark, Australia and Japan [12].

Figure 1 illustrates DCR choreographies with a simple contract between a Buyer and a Seller that we will use as a running example. It contains three interactions: 1) Request (Buyer asks Seller for a quote), 2) Quote (Seller sends a quote to Buyer), 3) Decide (Buyer announces a decision to Seller). Figure 1 also specifies 4 rules: Rule 1 is modelled with two response relations (blue arrows) from Request to Quote and Decision. They define commitments (the interactions must happen after Quote is executed). Rule 2 is modelled by a condition relation (orange arrow) from Request to Decide. Rule 3 is modelled by a milestone relation (purple arrow) from Quote to Decide, which means that if Quote is currently pending, i.e. required to be executed, then Decide cannot be executed. Finally, rule 4 is modelled by an inclusion relation (green arrow) from Request to Quote and an exclusion relation (red arrow) from Decide to Quote and by making Quote initially excluded. This way, Quote can only be executed if it has been included after the execution of Request and not subsequently excluded due to an execution product of executing Decide. Note that the declarative notation allows for expressing the rules instead of the flows respecting the rules. In particular, the model allows the flexibility that the Buyer asks several times before Seller responds by sending a Quote, that the Buyer can ask again after having received a quote, and that the Seller can also send more than one Quote before the Buyer decides.

However, in a realistic system, the Buyer has a time limit and won't wait indefinitely for a response. Once the time-out is reached, the Seller can no longer give a quote. Additionally, the Buyer communicates the requested product to the Seller, and their decision depends on the quote received from the Seller and certain decision logic based on the Buyer's locally specified maximum acceptable quote. These aspects cannot be expressed in our earlier versions of declarative choreographies [16], but become expressible in this contribution.

Our main contribution is to add language expressiveness to DCR choreographies so they can express message payloads and data and time constraints by using timed DCR graphs with data introduced in [15]. The extension of DCR graphs with time generalises the condition relation to allowing modelling of delays and the response relation to allow modelling deadlines. As we will see in the running example, timers can be modelled by combining delays and deadlines[1].

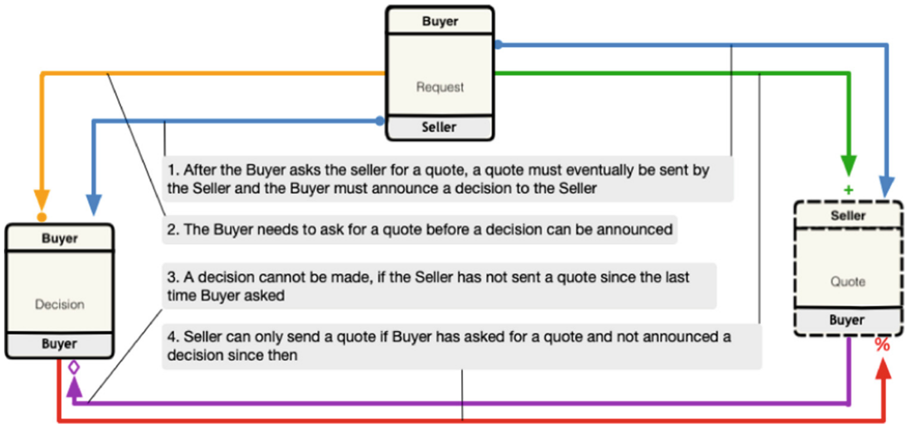


Fig. 1. A declarative DCR choreography for a Buyer Seller protocol.

Figure 2 shows a first attempt at extending the running example to include a timeout on the Quote interaction, a local input activity allowing the Buyer to define which product is requesting, and a data decision that depends on a local maximum value at the Buyer. The timeout is modelled by a delayed condition and a response with a deadline from Request to a new activity TimeOut, local to Buyer and that excludes both the Quote and the Decision interactions if it is executed. The combination of a delay and a deadline with the same period (7 d[2]),

[1] For simplicity, we will not consider explicit dates in expressions, however, this is supported in the process engine available for free academic use at dcrsolutions.net.
[2] specified as P7D in the ISO 8601 standard.

means that TimeOut acts as a timer executed exactly 7 d after the last execution of Request. The maximum accepted Quote is represented by a local data input activity Maximum at the Buyer. This attempt is however *not realizable*. The Seller has no way of knowing that the TimeOut timer action is executed since it is executed locally by the Buyer. Making the TimeOut action local to the Seller just moves the problem to the Buyer, which has then no way of knowing that the timer action has been executed and the Decide action should be excluded. The solution is to make the TimeOut timer action an interaction, which means that the execution of the timer is sent to the Seller. With this change, the choreography can be proven to be realizable.

To illustrate how the data dependencies come into play with respect to realizability, consider a protocol where the Seller is cheating and sends a Quote which is one Euro less than the maximum computed by an expression referring to the input value by the data activity Maximum. Again, this choreography will not be realizable, since expressions will not be allowed to refer to non-local activities.

Our second contribution is an updated set of criteria to determine whether declarative choreographies with data and time constraints are realizable. This requires us to revise the original Definitions of projectability in [16] and extend for the causal dependencies created by local computations and the evaluation of guards. Concretely, we add additional constraints, guaranteeing that only locally available data is used in expressions. In **our last contribution**, we showcase that endpoint projection of realizable choreographies behave bisimilarly to DCR choreographies with Time and Data.

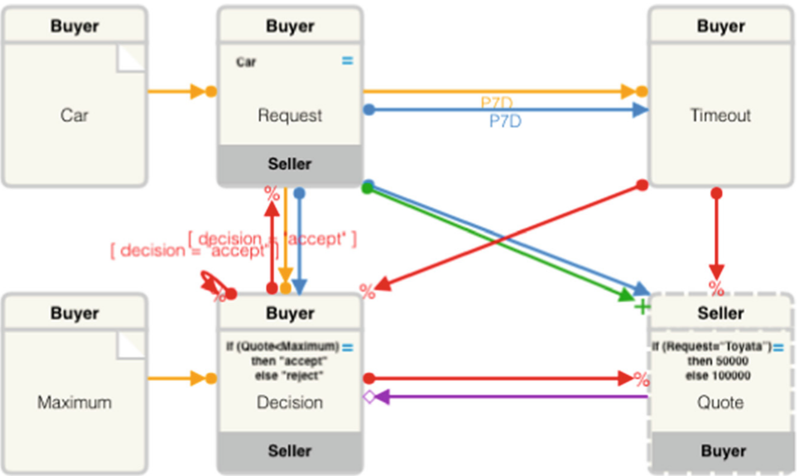


Fig. 2. Timed Buyer Seller DCR Choreography with Data.

Paper Outline: In Sect. 2 we introduce mixed timed DCR Choreographies with data. In Sect. 3 we provide criteria for realizability, a definition of how to project a choreography to endpoints and a proof of semantic correspondence, as well as implementation considerations. Section 4 summarizes related work. Finally, Sect. 5 concludes.

2 Mixed Timed DCR Choreographies with Data

A timed DCR graph with data [15] is a labelled, directed multigraph together with a marking defining the state of the graph. The nodes of the graph represent labelled events/actions and the edges represent relations that either constrain the execution of an event or define an effect on another event of the execution of an event. We assume a set $\mathsf{Exp_E}$ of computational expressions, and $\mathsf{BExp_E} \subset \mathsf{Exp_E}$ the subset of boolean expressions over E. V denotes the set of all possible values. Finally, we will let ω denote the natural numbers including zero, let $\infty = \omega \cup \{\omega\}$, i.e. the natural numbers including zero and the value infinity, let $A \rightharpoonup B$ denote the partial functions from A to B and let $\mathcal{P}_f(E)$ denote the set of all finite subsets of the set E. Timed DCR graphs with data are defined as follows.

Definition 1. *A* timed DCR Graph with data *G is given by the tuple* $(E, L, D, M, \ell, \rightarrow\!\bullet, \bullet\!\rightarrow, \rightarrow\!\diamond, \rightarrow\!+, \rightarrow\!\%)$ *where:*

1. *E is a finite set of* events*,*
2. *L is the set of labels,*
3. *$D : E \to \mathsf{Exp_E} \uplus \{?\}$ is the* data function*,*
4. *$l\colon E \to L$ is a labelling function between events and labels,*
5. *$M = (\mathsf{Ex}, \mathsf{Re}, \mathsf{In}, \mathsf{Va}) \in \big((E \rightharpoonup \omega) \times (E \rightharpoonup \infty) \times \mathcal{P}_f(E) \times (E \rightharpoonup V)\big)$ is the* timed marking with data*,*
6. *$\rightarrow\!\bullet \subseteq E \times \omega \times \mathsf{BExp_E} \times E$, is the* guarded timed condition relation*,*
7. *$\bullet\!\rightarrow \subseteq E \times \infty \times \mathsf{BExp_E} \times E$, is the guarded timed response relation,*
8. *$\rightarrow\!\diamond, \rightarrow\!+, \rightarrow\!\% \subseteq E \times \mathsf{BExp_E} \times E$ are the guarded milestone, include and exclude relations respectively.*

If $D(e) = ?$, we say that the event is an input action *otherwise it is a* computation*.*

The marking $M = (\mathsf{Ex}, \mathsf{Re}, \mathsf{In}, \mathsf{Va})$ consists of three functions and a set of events, which define the state of the process. The function Ex records for each event the time since it was last **Ex**ecuted*. The function Re records the deadline for when a* pending **Re**sponse *event is required to happen (if it is included). The set In records the currently* **In***cluded events. Finally, the* **Va**lue function *$\mathsf{Va} : E \rightharpoonup V$ assigns the current value of an event, if defined. We require that if $Ex(e)$ is defined then $\mathsf{Va}(e)$ is defined. Let $[[d]]_M$ be the partial evaluation relation of expressions $d \in \mathsf{Exp_E}$ relative to the marking M. We assume that for every event $e \in E$ there is a corresponding expression $e \in \mathsf{Exp_E}$ that evaluates to the value $\mathsf{Va}(e)$.*

We define Mixed Timed DCR Choreographies with Data as Timed DCR graphs with data where labels can be interactions or actions. Assume a fixed set of action names A, ranged over by $a, b, \ldots$, and a fixed set of participant roles R, ranged over by $r, r', r_1, r_2, \ldots$. We define interactions and actions as follows.

Definition 2. *An* interaction *is a triple* $(a, r \rightarrow r')$*, in which the action* $a \in \mathsf{A}$ *is* initiated *by the role* $r \in \mathsf{R}$ *and* received *by the role* $r' \in \mathsf{R}\backslash\{r\}$*, i.e. a role distinct from* r*. We denote by* I_A *the set of all interactions over action names* A*. A local* input action *and* internal action *with the label* a *for role* r *is written as* $?(a, r)$ *and* (a, r) *respectively. We denote by* Inp_A *the set of all local input and internal actions over action names* A*. Moreover, we define* $\mathsf{Initiator}(\cdot)$ *(respectively* $\mathsf{Receivers}(\cdot)$ *and* $\mathsf{Participants}(\cdot)$*) as the function returning roles from an interaction, local input or internal action, where:*

$$\mathsf{Initiator}((a, r \rightarrow r')) = \mathsf{Initiator}(?(a, r)) = \mathsf{Initiator}((a, r)) = r$$

$$\mathsf{Receivers}((a, r \rightarrow r')) = \{r'\} \qquad \mathsf{Receivers}(?(a, r)) = \mathsf{Receivers}((a, r)) = \emptyset$$

$$\mathsf{Participants}(\phi) = \{\mathsf{Initiator}(\phi)\} \cup \mathsf{Receivers}(\phi) \qquad \textit{for } \phi \in \mathsf{I}_\mathsf{A} \cup \mathsf{Inp}_\mathsf{A}$$

Sometimes we refer to internal actions as *computations*. We are now ready to define Mixed Timed DCR Choreographies with Data.

Definition 3. *A triple* (G, A, R) *is a* Mixed Timed DCR Choreography with Data *when* G *is a timed DCR graph with data with labels* $L \subseteq \mathsf{I}_\mathsf{A} \cup \mathsf{Inp}_\mathsf{A}$*, and for any event* e*,* $D(e) = ?$ *if and only if* $l(e) = ?(a, r)$ *for some* a *and* r*.*

Figure 2 shows a timed DCR Choreography with data with six events represented by boxes. Interactions show the initiator's role is at the top and the receiver's role is at the bottom. Input actions are denoted by boxes with a flipped top right corner. There are two input actions: Car (for specifying the desired car for a quote) and Maximum (for the maximum acceptable quote). Computations are marked by a "=" sign, accompanied by an expression. There are three computations: Request (with the expression Car, that takes the value of the input action Car), Decision (with the expression if $(Quote < Maximum)$ then "accept" else "reject"), and Quote (with the expression if $(Request = \text{“}Toyota''\text{)}$ then 50000 else 100000).

Whenever a computation e is executed in a graph with marking M, the value of computing the expression $D(e)$ will be assigned to the event by updating the value function Va in the marking such that $\mathsf{Va}'(e) = [[D(e)]]_M$. An input action will be assigned a data value provided as input when it is executed.

The relations define the effects of events execution and they constrain the executions of the process described by a DCR graph. Generally any guarded relation $\xrightarrow{[g]}$ only applies if g evaluates to true. A *condition* relation $e \rightarrow\!\bullet_d e'$ means that e is a condition for e', i.e. if e is included, then e must have been executed at least d time steps ago for e' to be enabled for execution. A *response* relation $e \bullet\!\!\rightarrow_k e'$ means that whenever e is executed, e' becomes a pending response with deadline $k \in \infty$. During process execution, a pending event with a deadline k must either be executed within k time steps, be made not pending by the no-response relation explained next, or be permanently excluded. Note that if $k = \omega$ then the event e' must eventually be executed, and in this case, we often simply omit the deadline k. A *milestone* relation $e \rightarrow\!\diamond e'$ means that if e is included, it must not be pending for e' to be enabled for execution. We refer to e as a milestone for e'. Finally, an *inclusion* (respectively *exclusion*) relation $e \rightarrow\!+ e'$ (respectively $e \rightarrow\!\% e'$) means that if e is executed, then e' is included in (respectively excluded from) the marking. An example of a guarded exclusion relation can be seen from the Decision event to the Request event. The guard, $[decision = \text{“}accept\text{”}]$ means that only if the expression in the Decision event evaluates to *accept* the event Request is excluded.

To define the semantics of a timed DCR graph with data we first define when events are enabled and time can be advanced.

Notation 4 *When $f : X \to Y$ is a (possibly partial) function, we write $f[x \mapsto y]$ for the function $f' : X \to Y$ which is identical to f, except $f'(x) = y$. We apply this notation also to sets, taking $f[x \mapsto y \mid P(x, y)]$ to be the function f' which is identical to f except that $f'(x) = y$ for all x, y satisfying the given predicate $P(x, y)$. Finally, we let $f(x) = \bot$ denote that function f is undefined for x.*

Definition 5 (Event and time step enabling). *Let G be a timed DCR graph with data containing the set of events E and markings $M = (\mathsf{Ex}, \mathsf{Re}, \mathsf{In}, \mathsf{Va})$. An event $e \in E$ is* enabled with value v for the marking M, *writing* $\mathsf{enabled}(M, (e, v))$ *iff:*

1. $e \in \mathsf{In}$,
2. $\forall e' \in \mathsf{In}.\ e' \xrightarrow{[g]}\!\bullet_k e \wedge [[g]]_M = \mathtt{tt} \implies k \leq Ex(e')$, *and*
3. $\forall e' \in \mathsf{In}.\ e' \xrightarrow{[g]}\!\diamond e \wedge [[g]]_M = \mathtt{tt} \implies Re(e')$*is undefined.*
4. $\forall e' \in E.e \xrightarrow{[g]}\!+ e' \wedge [[g]]_{M'} = \mathtt{tt} \implies \neg(Re(e') < 0)$
5. $v = [[D(e)]]_M$, *if* $D(e) \neq\ ?$.
6. $M' = (\mathsf{Ex}, \mathsf{Re}, \mathsf{In}, \mathsf{Va}[e \mapsto v])$, *if* $D(e) \neq\ ?$ *and* $M' = M$, *otherwise*

For $n \in \omega$ we say that the time step n is enabled, *written* $\mathsf{enabled}(M, n)$, *if* $\forall e \in \mathsf{In}.(\mathsf{Re}(e) = k \implies n \leq k)$.

Basically, for an event e to be enabled it (1) must be included, (2) whenever e has a condition with a guard evaluated to true in the current marking from an included event e' with delay k, then e' was executed at least k time steps ago, (3) No included e', with a milestone relation to e with a guard that evaluates to true in the current marking, is pending, (4) If a pending event is included, the deadline cannot be passed [3], and (5) For a computation e, the value v is the result of computing the expression $D(e)$ in the marking M and otherwise the value v is any value provided by the environment. Finally, a time step n denotes that time advances n steps and can only happen if all included pending events have a deadline greater or equal to n. We now proceed to define event effects.

Definition 6. *Let G be a timed DCR graph with data and a marking $M =$* $(\mathsf{Ex}, \mathsf{Re}, \mathsf{In}, \mathsf{Va})$. *The effect of executing an enabled event e with value v is the*

[3] Note that this differs from the original definition of timed DCR Graphs.

marking $M \cdot e = (\mathsf{Ex}', \mathsf{Re}', \mathsf{In}', \mathsf{Va}')$ *where:*

$$\begin{aligned}
\mathsf{Ex}' &= \mathsf{Ex}[e \mapsto 0] \\
\mathsf{Re}' &= \mathsf{Re}[e \mapsto \bot][e' \mapsto k \mid \exists k'.\ e \bullet\!\xrightarrow{[g]}_{k'} e' \wedge [[g]]_{M'} = \mathtt{tt} \wedge k = min\{k' \mid e \bullet\!\xrightarrow{[g]}_{k'} e' \wedge [[g]]_{M'} = \mathtt{tt}\}] \\
\mathsf{In}' &= \mathsf{In} \setminus (\{e' \in E \mid e \xrightarrow{[g]}\!\% e'.[[g]]_{M'} = \mathtt{tt}\} \cup \{e' \mid e \xrightarrow{[g]}\!+ e'.[[g]]_{M'} = \mathtt{tt}\}) \\
\mathsf{Va}' &= \mathsf{Va}[e \mapsto v], \\
M' &= (\mathsf{Ex}, \mathsf{Re}, \mathsf{In}, \mathsf{Va}')
\end{aligned}$$

The effect of executing an enabled time step $n \in \omega$ *in the marking* M *is the marking* $M \cdot e = (\mathsf{Ex} \oplus n, \mathsf{Re} \ominus n, \mathsf{In}, \mathsf{Va})$ *where* $(\mathsf{Ex} \oplus n)(e) = \mathsf{Ex}(e) + n$ *if* $\mathsf{Ex}(e)$ *is defined and undefined otherwise, and* $(\mathsf{Re} \ominus n)(e) = \mathsf{Re}(e) - n$ *if* $\mathsf{Re}(e)$ *is defined and undefined otherwise.*

The result of executing an enabled event e with value v is a new marking where the time since the last execution of the event e is set to 0 and the value is set to v. The response status of e is first set to be non-pending, and then all events e' with a response relation where the guard g evaluates to true is set to pending with the deadline being the minimal deadline of such a response relation. Finally, all events e' with an exclude relation where the guard g evaluates to true is set to excluded, and then all events e' with an include relation where the guard g evaluates to true and is set to included. The execution of a time step of length n increases the time since the last execution of all events by n and decreases the deadline on responses by n.

3 Timed DCR Processes with Data and Realizability

Here we provide sufficient and effectively computable criteria for the realizability of timed DCR Choreographies with data as a synchronous composition of timed DCR endpoint processes with data and prove operational correspondence. Finally, we comment on how synchronous composition can be justified by using distributed locking to ensure that two conflicting interactions cannot interfere.

We first characterise when the execution of an event in a choreography may influence, i.e. change the marking or enabledness of, another event. To this end, we define the notion of *direct dependency*.

Definition 7. *Let* G *be a timed DCR graph with data and let* $e, e' \in E$ *be events in* G. *Then there is a* direct dependency $e' \preceq e$ *from* e' *to* e *iff either one of the following conditions is true:*

1. $e' = e$,
2. $e'(\rightarrow\!\bullet \cup \bullet\!\rightarrow \cup \rightarrow\!+ \cup \rightarrow\!\% \cup \rightarrow\!\diamond)e$,
3. $\exists e''.\ e'(\rightarrow\!+ \cup \rightarrow\!\%)e''(\rightarrow\!\bullet \cup \rightarrow\!\diamond)e$,
4. $\exists e''.\ e' \bullet\!\rightarrow e'' \rightarrow\!\diamond e$.

That is, $e' \preceq e$ iff either (1) e and e' is the same event, (2) there is a relation from e' to e, (3) e' includes or excludes an event which is itself a condition/milestone for e, or (4) e' has a response to a milestone for e.

The following proposition states that an event e *must* be directly dependent on any event e' whose execution may change the marking or enabledness of e.

Proposition 8. *Let G be a DCR graph with marking $M = (\mathsf{Ex}, \mathsf{Re}, \mathsf{In})$. Suppose $e' \in \mathsf{enabled}(G)$, and let $G' = \mathsf{execute}(G, e')$ and $M' = (\mathsf{Ex}', \mathsf{Re}', \mathsf{In}')$ be the marking of G'. The relation $e' \preceq e$ holds if either of the following holds:*

1. $e \in \mathsf{enabled}(G) \not\Leftrightarrow e \in \mathsf{enabled}(G')$,
2. $e \in \mathsf{Ex} \not\Leftrightarrow e \in \mathsf{Ex}'$,
3. $e \in \mathsf{Re} \not\Leftrightarrow e \in \mathsf{Re}'$,
4. $e \in \mathsf{In} \not\Leftrightarrow e \in \mathsf{In}'$.

Proposition 8 implies that the relation $e' \preceq e$ is a sufficient but not necessary criterion for whether the execution of e' affects the state or enabledness of e'. For example, in a graph with only two events, e and f, and a single relation between them, $e \rightarrow\bullet f$, then $e \preceq f$ (as per Definition 7(2)). However, executing e does not change the status of f in terms of whether it is enabled or marked.

We now formally define timed DCR endpoint processes with data.

Definition 9. *A tuple (G, A, R, r) is a* timed DCR endpoint process with data *for role $r \in R$, participants R and actions A, if G is a timed DCR graph with data and the labels L of G are of the form $(?a@r', r)$, $(!a@r', r)$, $(?a, r)$ or a for $a \in A$ and $r' \in R\backslash\{r\}$:*

- *$(?a@r', r)$ is an input action from role r' with action label a,*
- *$(!a@r', r)$ is an output action to role r' with action label a,*
- *$(?a, r)$ is a local input action with the action label a.*

Moreover, we define projections on interactions as:

$$(a, r \rightarrow r')|_{r'} = (?a@r, r') \qquad\qquad (a, r \rightarrow r')|_r = (!a@r', r),$$

and for local actions as:

$$?(a, r)|_r = (?a, r) \qquad\qquad (a, r)|_r = a.$$

Finally, we say $\mathsf{Initiator}((?a@r, r')) = \mathsf{Initiator}((!a@r', r)) = \mathsf{Initiator}((?a, r)) = \mathsf{Initiator}(a) = r$.

Intuitively, we will obtain the endpoint process for a participant r by keeping (a) events labelled with interactions involving r, *as well as* (b) the direct dependencies of the events for which r is the initiator. We then change the interactions to endpoint input or output actions. In order for the interactions to make sense as actions for the endpoint process at r, the role r must be involved in the direct dependencies of the events for which r is the initiator. Moreover, data expressions associated with events (or guards on relations between such events) that are either local events for r or interactions where r is the initiator must only refer to the values of events that are either local for r or interaction for which r is involved. We formalize this as follows.

Definition 10 (Realizability). *Let $C = (G, A, R)$ be a DCR choreography and ℓ the labelling function of G, and recall the definitions of* $\mathsf{Initiator}()$ *and* $\mathsf{Participants}()$ *in Def. 2. C is* realizable *iff*

1. *for all* e*, if* $e' \preceq e$ *then* $\mathsf{Initiator}(\ell(e)) \in \mathsf{Participants}(\ell(e'))$*,*
2. *if* $D(e) \in \mathsf{Exp}_\mathsf{E}$ *then* $\forall e' \in D(e).\mathsf{Initiator}(\ell(e)) \in \mathsf{Participants}(\ell(e'))$*,*
3. *if* g *is a guard on a relation between events* e, e' *and* r *a role which is the initiator for one of the events and a participant of the other, then* $\forall e'' \in g.r \in \mathsf{Participants}(\ell(e''))$*,*

Since the direct dependency relation is a local property between events connected by a path of length at most two relations, the 1st realizability criterion can be checked in linear time in the number of events. The second criterion can be checked at most at a quadratic time on the number of events.

Example 11. We find that the choreography in Fig. 2 is *not* realizable, as the event Timeout is a direct dependency for Quote, but Seller, being the initiator of Quote, is not among the participants of Timeout. This violates condition 1 of Def. 10. The solution to making the choreography realizable is to change Timeout to an interaction with Seller as a receiver.

We now define the projection of a timed DCR choreography (G, A, R) with data to an endpoint process for a given role $r \in R$.

Definition 12 (DCR endpoint projection). *Let* (G, A, R) *be a timed DCR Choreography with data, where* $G = (E, D, M, L, \ell, \to\bullet, \bullet\to, \to\diamond, \to+, \to\%)$*. For any* $r \in R$*, we define*

$$\delta = \{e \in E \mid \mathsf{Initiator}(l(e)) = r\} \text{ and } E' = \{e \in E \mid \mathsf{Receivers}(l(e)) = \{r\}\},$$

Then the endpoint projection *of* G *for role* r *is the DCR graph*

$$G|_r = (E|_\delta \cup E', D|_r, M|_r, L|_\delta, \ell|_\delta, \to\bullet|_\delta, \bullet\to|_\delta, \to\diamond|_\delta, \to+|_\delta, \to\%|_\delta)$$

where:

1. $E|_\delta = \{e \in E \mid \exists e' \in \delta.\ e \preceq e'\}$,
2. $D|_r(e) = D(e)$ *if* $e \in E|_\delta \setminus E'$
3. $D|_r(e) = e$, *if* $e \in E'$
4. $M|_r = (\mathsf{Ex}|_r, \mathsf{Re}|_\delta, \mathsf{In}|_\delta \cup E' \setminus (E_{|\delta} \setminus \mathsf{In}|_\delta))$ *where:*
   (a) $\mathsf{Ex}|_r(e) = \mathsf{Ex}(e)$, *if* $e \in E|_\delta \cup E'$
   (b) $\mathsf{Re}|_\delta(e) = \mathsf{Re}(e)$, *if* $e \in E|_\delta$
   (c) $\mathsf{In}|_\delta = \big(\mathsf{In} \cap ((\to\bullet\ \delta) \cup\ (\to\diamond\ \delta) \cup\ \delta)\big) \cup \big(E|_\delta \setminus ((\to\bullet\delta) \cup (\to\diamond\delta) \cup \delta)\big)$.
   (d) $\mathsf{Va}|_r = \mathsf{Va}|_r$, *if* $e \in E|_\delta \cup E'$
5. $\ell|_r(e) = \ell(e)|_r$,
6. $L|_\delta = \mathsf{img}(\ell)$
7. $\to|_\delta = \to \cap \big((\to\delta) \times \delta\big)$ *for* $\to \in \{\to\bullet, \to\diamond\}$
8. $\bullet\to|_\delta = \bullet\to \cap \big(((\bullet\to\to\diamond\delta) \times (\to\diamond\delta)) \cup ((\bullet\to\delta) \times \delta)\big)$
9. $\to|_\delta = \to \cap(((\to\ \to\bullet\delta) \times (\to\bullet\delta)) \cup ((\to\ \to\diamond\delta) \times (\to\diamond\delta)) \cup ((\to\ \delta) \times \delta))$ *for* $\to \in \{\to+, \to\%\}$

Two sources of complexity for these rules are that events are included in the endpoint for different reasons and that some events change their computational expression, which was not an issue in the work on DCR Choreographies without data [16]. Condition 1 makes sure that we include all events in the endpoint, that have a direct dependency on an event where r is initiator. In addition to these events, the endpoint will also contain the events E' where r is a recipient but do not have a direct dependency on any event where r is an initiator. Condition 2 states that all events where r is initiator keep their expression (or status as local input), while Condition 3 states that all events where r is receiver will get an expression which is simply the event itself. This is because the value will be received as a message and assigned to the event. Condition 4 describes how markings are defined for the events included in the endpoint. We keep the execution status and value assignment for all events. For events that have direct dependency relation to an event where r is the initiator, we keep the response status. This means that for events where r is the recipient but the event is not a direct dependency on any event where r is the initiator, the endpoint does not record its pending status. The inclusion status is complex: Here we keep the inclusion status for events if r is the initiator, or the event is a condition or milestone for an event where r is the initiator. We include all other events that are either direct dependencies on an event where r is the initiator or r is the receiver, but not direct dependencies.

Example 13 The endpoints projecting the choreography in Fig. 2 corrected to make the timeout action an interaction are shown in Fig. 3.

3.1 Correctness of Realizability

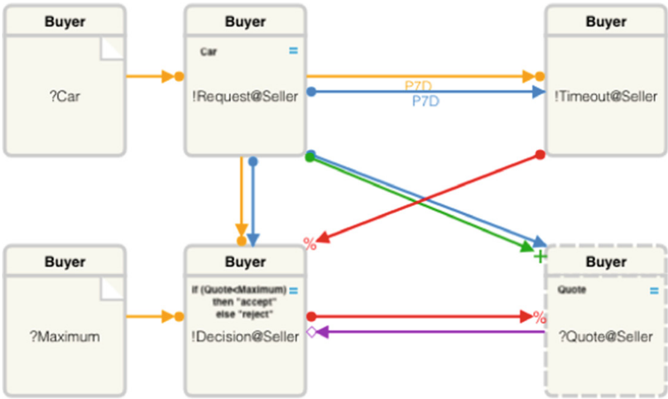
In this section, we prove the correctness of endpoint projections provided in the previous section. First we define a labelled event transition system with time, data and responses for a timed DCR graph with data.

Definition 14 (Transition semantics). *Let* $G = (E, D, M, \rightarrow\!\bullet, \bullet\!\rightarrow, \rightarrow\!\diamond, \rightarrow\!+, \rightarrow\!\%, L, \ell)$ *be a timed DCR graph with data. The Labelled Event Transition System with Data and Responses (LETSDR) for G is defined as*

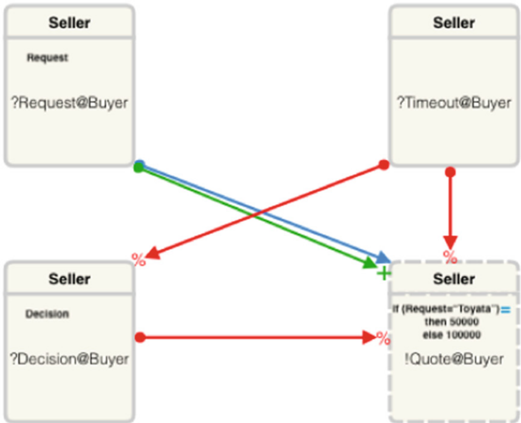
$$\mathcal{T}(G) = (\mathcal{M}, M, E_d, L, \ell, \rightarrow, \rho),$$

where the marking M of the DCR graph G is the initial state, $E_d = E \times V$ is the set of pairs of events and data values, L is the set of labels, ℓ is the labelling function, $\rightarrow \subseteq \mathcal{M} \times (E_d \cup \omega) \times \mathcal{M}$ is the transition relation, defined by $(M', \phi, M'') \in \rightarrow$ iff $\mathsf{enabled}(M', \phi)$, $M'' = \mathsf{execute}(M', \phi)$. Finally, $\mathcal{M} = \{M' \mid M \rightarrow^* M'\}$, the set of states, is the markings reachable from the initial marking M by execution of events or time steps, and finally ρ is the timed response function defined on the states of the transition system by $\rho(M') = \{(e, k) \mid e \in \mathsf{In} \wedge \mathsf{Re}(e) = k\}$, if $M' = (\mathsf{Ex}, \mathsf{Re}, \mathsf{In}, V)$.

We define the accepting runs of a timed DCR graph as follows.



(a) Endpoint projection for Buyer



(b) Endpoint projection for Seller

Fig. 3. Endpoint projections for Buyer and Seller in Example 11

Definition 15 (Accepting runs). *Let G be a timed DCR graph with data with events E. Let $\bar{\alpha}$ be an infinite transition sequence $M = M_0 \xrightarrow{\alpha_1} M_1 \xrightarrow{\alpha_2} \cdots$ of length $k \in \infty$ with $M_i = (Ex_i, Re_i, In_i)$. We say that $\bar{\alpha}$ is an* accepting run *iff*

1. *For all $i \leq k$ and all $e \in E$, if $(e, \omega) \in \rho(M_i)$ then for some $j > i$ we have $\alpha_j = (e, d, v)$ for some $d \in \mathsf{Exp}_\mathsf{E} \uplus \{?\}$ or $(e, \omega) \notin \rho(M_j)$*
2. *$\bar{\alpha}$ contains infinitely many time steps, i.e. there exists infinitely many indices i such that $\alpha_i = n_i$ for some $n_i \in \omega$.*

The condition 1 captures that if an event at some intermediate marking is included and pending and required eventually to happen, i.e. with deadline ω, it will eventually be executed or no longer pending with deadline ω and included.

This is the acceptance criteria for un-timed DCR graphs. The condition 2 captures that an accepting run is non-Zeno, i.e. time must keep progressing. As a consequence, only infinite traces are accepting in timed DCR graphs, but an accepting trace may contain only finitely many non-time steps.

Definition 16 (Trace). *A trace t of a graph G with initial marking M is an infinite sequence $\alpha_1, \alpha_2, \alpha_3, \cdots$ such that there exists an accepting run $M \xrightarrow{\alpha_1} M_1 \xrightarrow{\alpha_2} M_2 \xrightarrow{\alpha_3} \cdots$. We write $\mathsf{traces}(G)$ for the set of all possible traces in G.*

We say that two labelled Event Transition Systems with Data and Responses T and T' are isomorphic, written $T \equiv T'$ if there is an isomorphism between the sets of states preserving and respecting transitions and the response function. Since the definition of accepting runs only depends on the included pending responses in the markings of the graphs and the events and time steps being executed during a run, it is easy to see that if two DCR graphs have isomorphic transition systems with responses then they also have the same languages.

Proposition 17. *Let G and G' be DCR graphs. If $\mathcal{T}(G) \equiv \mathcal{T}(G')$ then $\mathsf{traces}(G) = \mathsf{traces}(G')$.*

Lemma 18. *Let (G, A, R) be a DCR choreography, let $r \in R$ be a role of R, and let $(G_{|r}, A_{|r}, R, r)$ be the projection of that choreography to r. If (G, A, R) is realizable then every label in $G_{|r}$ is an interaction with r as a participant.*

If an event is shared between two endpoints, then it must be the case that they share the same initiator.

Lemma 19. *Let $C = (G, A, R)$ be a realizable DCR choreography, $r, r' \in R$ be roles of R, and $(G_{|r}, A_{|r}, R, r)$ and $(G_{|r'}, A_{|r'}, R, r')$ be the projections of C to r and r'. If $e \in E \cap E'$, where E and E' are the events of $G_{|r}$ and $G_{|r'}$ respectively, then $\mathsf{Initiator}(\ell(e)) = \mathsf{Initiator}(\ell(e'))$.*

We now define the synchronous composition of a finite set of DCR endpoints. Intuitively, an event e is enabled in the synchronous composition, if it is enabled in all of the endpoints in which it occurs. The execution of an event is then defined simply by executing the event in all of the components it occurs. Finally, the label is the interaction obtained by taking the union of receivers.

Definition 20 (Synchronous composition of DCR endpoints). *For $R = \{r_1, \ldots, r_n\}$ and DCR endpoints $P_i = (G_i, A_i, R, r_i)$ for $i \in \{1, .., n\}$ we define the synchronous parallel composition as $P = \Pi_{i \in \{1,..,n\}} P_i$ as:*

- *$E = \bigcup_i \in \{1, .., n\} E_i$, where E_i is the events of G_i.*
- *$e \in \mathsf{enabled}(P)$ iff $e \in E_i \implies e \in \mathsf{enabled}(G_i)$ for all $i \in \{1, .., n\}$.*
- *$\mathsf{execute}(P, e) = \Pi_{i \in \{1,..,n\}} P'_i$, if $e \in \mathsf{enabled}(P)$ and $P_i = (G'_i, A_i, R, r_i)$ and $G'_i = \mathsf{execute}(G_i, e)$, if $e \in E_i$ and $P'_i = P_i$ otherwise.*
- *$\ell_P(e) = (a, r \to R')$ if $e \in E_i \implies \ell_i(e) = (a, r \to R'_i)$ and $R' = \bigcup_i \in I$, where $I = \{i \in \{1, \ldots, n\} \mid e \in E_i\}$.*

– $\rho_P(P') = \bigcup_{i\in\{1,\ldots,n\}} \mathsf{Re}_i \cap \mathsf{In}_i$ *if* $P' = \Pi_{i\in\{1,..,n\}} P'_i$, $P'_i = (G'_i, A_i, R, r_i)$ *and the marking of* G'_i *is* $(\mathsf{Ex}_i, \mathsf{Re}_i, \mathsf{In}_i)$.

We now define the LETSDR for P *by* $\mathcal{T}(P) = (\mathcal{P}, P, E, L, \ell_P, \rightarrow_P, \rho_P)$, *where* $(P', e, P'') \in \rightarrow_P$ *if* $e \in \mathsf{enabled}(P')$ *and* $P'' = \mathsf{execute}(P', e)$, *and* $\mathcal{P} = \{P' \mid P \rightarrow^* P'\}$.

The following theorem establishes the key property, that the synchronous composition of the endpoints yields a transition system with responses isomorphic to the transition system for the choreography.

Theorem 21. *Let* $C = (G, A, R)$ *be a realizable DCR choreography,* $R = \{r_1, \ldots, r_n\}$ *and* $P_i = (G_i, A_i, R, r_i)$ *for* $i \in \{1, \ldots, n\}$ *the DCR endpoints resulting from endpoint projection of* C*. Then* $\mathcal{T}(C) \equiv \mathcal{T}(\Pi_{i\in\{1,..,n\}} P_i)$.

Proof. (Sketch) The proof follows the same approach as the proof of Thm.5.1 in [14] where a bisimulation is constructed between the original graph (in this case the choreography) and the network of the synchronous parallel composition of projections.

We note that the isomorphism by Proposition 17 implies that the language of the choreography is the same as the language of the composition of the endpoints.

3.2 Implementing Synchronous Communication

The correspondence between the behaviour of the choreography and the behaviour of the synchronous composition proven in the previous section is only useful if we can implement synchronous composition. A way to do this is to use a distributed locking scheme: Firstly, we assume a fixed order relation given on the events in the choreography. Then, before execution of an event e in an endpoint process, the process can request to lock all events that have direct dependencies to e or that e has a direct dependency to - and if it is a send event, also the corresponding receiving event and its direct dependencies or dependants are locked. Here it is important that the requests for locks are done in the order given for events. If all locks are successfully obtained, the event e is executed. Any event that may change marking as a result of the execution is already locked - and no event in another endpoint process that can change the marking of any of the events can be executed at the same time (since it would then have obtained its locks at the same time). After execution and updating the marking, all locks are released. Note that deadlocks can not occur because of the global ordering of locks.

As for BPMN choreographies, the timings of events will be approximate, since timing constraints cross between different participants. For instance, the timeout event executed by the Buyer in our running example will only take effect at the Sender endpoint when the message has been received, thereby excluding the Quote event at a time after the timeout.

As for traditional choreographies, the correspondence of behaviour guarantees, that if the choreography is deadlock and livelock free, then the network of endpoints is also deadlock and livelock free. The timing constraints introduce possibilities of timelocks [14], i.e. where a deadline for an event is sooner than allowed by a delay to the event. If data domains are kept finite, DCR graphs map to finite Büchi-automata for which the absence of deadlocks, livelocks and timelocks can be proven [14, 21].

4 Related Work

The related work covers two major areas: model-driven engineering and analysis & verification of distributed systems. On the former, the work by Fdhila et al. [8] proposed techniques to generate endpoints from BPMN choreographies, where realizability is defined as a set of control-flow conditions without data or timing constraints. BPMN choreographies have been extended in [17] to support shared data objects such as those present in a blockchain. Similar imperative choreographies were extended in [19] with data primitives, but not endpoint projections. [5] presents a choreographic model based on the Klaim calculus [2] where communication is done via distributed tuple spaces, thus allowing both data and control-flow constraints. On the latter, [7] presents an encoding of the session π calculus into a calculus of linear temporal constraints allowing for richer communication patterns including communication flows, timing constraints and contextual information. In [3] an extension of session types to model asynchrony in timed protocols is proposed. The new Choral language [10] implements choreographic programming with new abstractions where choreographies are objects. Earlier than this, time extensions of the conversation calculus [6] were presented in [18] but these did not consider endpoint realizability. In a similar trend, imperative choreographies have been applied to synthesise correct distributed implementations where endpoints synchronise via handshakes of correlation data [9]. It is important to mention that in all the works cited (1) the guarantees for communication are expressed in terms of safety properties, thus liveness is not considered, and (2) the constructs for continuation are defined in terms of imperative flows, contrasting to continuations based on may/must relations such as those existing in DCR graphs. DCR choreographies have been used to monitor change requests in blockchains [4]. As our work presents an orthogonal extension to the control flow relations used in [16], we believe that monitors such as that in [4] will benefit from an additional expressive power where data and time constraints can be expressed.

5 Concluding Remarks and Related Work

We presented the first declarative choreography language taking control flow, time and data into account, providing criteria for realizability as communicating endpoint processes that considers multi-perspective constraints.

The assumed communication model is synchronous. If asynchronous communication is required, we will need an extension of the execution semantics with message queues, and it is left for future work. Other aspects for future work involve studying the understandability of the approach with real users, in a similar way that we have done with DCR graphs [1]. We have implemented a prototype tool able to analyze DCR choreographies and generate the projections and endpoints, showcasing the implementability of the approach [11]. A limitation of the approach is the inability to capture multi-instance subprocesses. A solution is to extend the notion of interaction with session initiation primitives [20], so each endpoint shares a fresh CaseId at each instance. We leave these extensions for future work. Finally, we would like to explore code generation from choreographies, as well as the generation of microservice architectures.

Acknowledgments. Work supported by the Danish Council for Independent Research project *PAPRICAS.org*

References

1. Abbad Andaloussi, A., Buch-Lorentsen, J., López, H.A., Slaats, T., Weber, B.: Exploring the modeling of declarative processes using a hybrid approach. In: Laender, A.H.F., Pernici, B., Lim, E.-P., de Oliveira, J.P.M. (eds.) ER 2019. LNCS, vol. 11788, pp. 162–170. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33223-5_14
2. Bettini, L., et al.: The Klaim Project: theory and practice. In: Priami, C. (ed.) GC 2003. LNCS, vol. 2874, pp. 88–150. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40042-4_4
3. Bocchi, L., Murgia, M., Vasconcelos, V.T., Yoshida, N.: Asynchronous timed session types. In: Caires, L. (ed.) ESOP 2019. LNCS, vol. 11423, pp. 583–610. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17184-1_21
4. Brahem, A., et al.: A trustworthy decentralized change propagation mechanism for declarative choreographies. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds) Business Process Management. BPM 2022. Lecture Notes in Computer Science. vol 13420. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16103-2_27
5. Bruni, R., Corradini, A., Gadducci, F., Melgratti, H., Montanari, U., Tuosto, E.: Data-driven choreographies à la Klaim. In: Boreale, M., Corradini, F., Loreti, M., Pugliese, R. (eds.) Models, Languages, and Tools for Concurrent and Distributed Programming. LNCS, vol. 11665, pp. 170–190. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21485-2_11
6. Caires, L., Vieira, H.T.: Analysis of service oriented software systems with the conversation calculus. In: Barbosa, L.S., Lumpe, M. (eds.) FACS 2010. LNCS, vol. 6921, pp. 6–33. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27269-1_2
7. Cano, M., López, H.A., Pérez, J.A., Rueda, C.: Session-based concurrency, declaratively. Acta Informatica **59**(1), 1–87 (2022)
8. Fdhila, W., Indiono, C., Rinderle-Ma, S., Reichert, M.: Dealing with change in process choreographies: design and implementation of propagation algorithms. Inf. Syst. **49**, 1–24 (2015)
9. Giallorenzo, S., Montesi, F., Gabbrielli, M.: Applied choreographies. In: Baier, C., Caires, L. (eds.) FORTE 2018. LNCS, vol. 10854, pp. 21–40. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92612-4_2

10. Giallorenzo, S., Montesi, F., Peressotti, M.: Choreographies as objects. arXiv preprint arXiv:2005.09520 (2020)
11. Guo, H.: Extension of Declarative Choreographies and Its Application in Distributed Systems. Master's thesis, University of Copenhagen (2022)
12. Hildebrandt, T.T., et al.: EcoKnow: engineering effective, co-created and compliant adaptive case management systems for knowledge workers. In: 2020 IEEE/ACM International Conference on Software and System Processes (ICSSP), pp. 155–164. ACM (2020)
13. Hildebrandt, T.T., Mukkamala, R.R.: Declarative event-based workflow as distributed dynamic condition response graphs. In: PLACES. Electronic Proceedings in Theoretical Computer Science (EPTCS). vol. 69, pp. 59–73 (2010)
14. Hildebrandt, T.T., Mukkamala, R.R., Slaats, T., Zanitti, F.: Contracts for cross-organizational workflows as timed dynamic condition response graphs. JLAMP **82**(5–7), 164–185 (2013)
15. Hildebrandt, T.T., Normann, H., Marquard, M., Debois, S., Slaats, T.: Decision modelling in timed dynamic condition response graphs with data. In: Marrella, A., Weber, B. (eds.) BPM 2021. LNBIP, vol. 436, pp. 362–374. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94343-1_28
16. Hildebrandt, T.T., Slaats, T., López, H.A., Debois, S., Carbone, M.: Declarative choreographies and liveness. In: Pérez, J.A., Yoshida, N. (eds.) FORTE 2019. LNCS, vol. 11535, pp. 129–147. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21759-4_8
17. Ladleif, J., Weske, M., Weber, I.: Modeling and enforcing blockchain-based choreographies. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 69–85. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_7
18. López, H.A., Pérez, J.A.: Time and exceptional behavior in multiparty structured interactions. In: Carbone, M., Petit, J.-M. (eds.) WS-FM 2011. LNCS, vol. 7176, pp. 48–63. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29834-9_5
19. Meyer, A., Pufahl, L., Batoulis, K., Fahland, D., Weske, M.: Automating data exchange in process choreographies. Inf. Syst. **53**, 296–329 (2015)
20. Montesi, F.: Choreographic Programming. Ph.D. thesis, ITU, Copenhagen (2013)
21. Mukkamala, R.R.: A Formal Model For Declarative Workflows: Dynamic Condition Response Graphs. Ph.D. thesis, IT University of Copenhagen (June 2012)
22. Object Management Group BPMN Technical Committee: Business Process Model and Notation, version 2.0.2 (2014). http://www.omg.org/spec/BPMN/2.0.2/PDF
23. Tiezzi, F., Re, B., Polini, A., Morichetta, A., Corradini, F.: Collaboration vs. choreography conformance in BPMN. In: Logical Methods in Computer Science. vol. 16 (2020)

Execution Semantics for Process Choreographies with Data

Tom Lichtenstein(✉) and Mathias Weske

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
{Tom.Lichtenstein,Mathias.Weske}@hpi.de

Abstract. Interaction models, such as BPMN choreography diagrams, enable the coordination of interactions between organizations within a process choreography. Since choreography participants typically do not share a central data store, data flow must be considered during design to ensure that each participant has sufficient data to continue a conversation. However, current choreography modeling languages lack a concise notation and execution semantics for data exchange. This paper refines the data flow specifications of choreography diagrams using supplemental models. Furthermore, the execution semantics for data exchange is formally defined by data-enhanced interaction Petri nets. The extended semantics enable the analysis of data awareness and data consistency at design time.

Keywords: Process choreographies · Interaction Petri nets · Data flow

1 Introduction

In a networked economy, the exchange of goods and services between organizations is key to success. To ensure effective collaboration, the interactions of the organizations' internal processes must be carefully designed, e.g., in process choreographies. Since choreography participants typically do not share a central data store, interorganizational data flow must be considered to ensure that each participant has sufficient data to advance the conversation. While interaction models such as BPMN choreography diagrams [15] allow us to define interaction behavior from a global perspective, current choreography modeling languages lack concise notation and execution semantics for data exchange. As a result, it is not possible to verify at design time that message senders are aware of the required data and that participants affected by global decisions have a consistent view of the data driving those decisions. This paper aims to improve data modeling support for process choreographies by refining the specification of exchanged data for choreography diagrams. A formal execution semantics for data exchange is provided by mapping supplemented choreography diagrams to interaction Petri nets extended with a data perspective. The mapping provides a foundation for a data-aware formal analysis and verification of choreographies at design time.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 90–106, 2023.
https://doi.org/10.1007/978-3-031-41623-1_6

The remainder of the paper is organized as follows: Sect. 2 outlines the fundamentals of process choreographies and interaction Petri nets, followed by a motivating example in Sect. 3. Section 4 introduces the refined data flow specifications and formal execution semantics for data exchange as the main contributions of this paper. Next, Sect. 5 discusses the presented approach, and Sect. 6 gives a brief overview of related work. Finally, Sect. 7 summarizes the results of this work.

2 Preliminaries

As a foundation for this work, this section outlines the basic concepts and formal principles of process choreographies in Sect. 2.1 and interaction Petri nets in Sect. 2.2.

2.1 Process Choreographies

Process choreographies define the possible interaction sequences between business actors (i.e., *choreography participants*) that collaborate to achieve a goal [5]. Each choreography participant is associated with a *role*. An execution of a choreography is referred to as a *conversation*. Definition 1 specifies the main concepts of process choreographies.

Definition 1. *(Process Choreography). A process choreography is defined by a tuple* $\mathcal{C} = (N, SF, R, M, \mathcal{G}, grd, init, resp, msg)$*, where:*

- $N \subseteq \mathcal{T} \times G \times E$ *is a finite, non-empty set of nodes including choreography tasks* $\mathcal{T}$*, gateways* G*, and events* E*,*
- E *can be partitioned into disjoint sets of start events* E_s *and end events* E_e*,*
- G *can be partitioned into disjoint sets of event-based gateway splits* G^s_e*, exclusive gateway splits* $G^s_\times$*, exclusive gateway joins* $G^j_\times$*, parallel gateway splits* G^s_+*, and parallel gateway joins* G^j_+*,*
- $SF \subseteq N \times N$ *is a finite, non-empty set of sequence flows,*
- R *is a finite, non-empty set of participant roles,*
- M *is a finite set of messages,*
- $\mathcal{G}$ *is a finite set of guards,*
- $grd : G^s_\times \times N \rightarrow \mathcal{G}$ *assigns a guard to a sequence flow,*
- $init : \mathcal{T} \rightarrow R$ *assigns the initiating role to a choreography task,*
- $resp : \mathcal{T} \rightarrow R$ *assigns the respondent role to a choreography task, and*
- $msg : \mathcal{T} \rightarrow (M \times M) \cup M \cup \{\emptyset\}$ *assigns messages to a choreography task.*

BPMN 2.0 introduces *choreography diagrams* as an interaction modeling language for process choreographies [15]. Unlike BPMN collaboration diagrams, choreography diagrams abstract from process-internal details and focus only on interorganizational behavior. Choreography diagrams represent interactions as *choreography tasks*, hereafter referred to as tasks. Each task is associated with an initiator (white badge) and a respondent (gray badge). Optionally, an initial message (white envelope) and a response message (gray envelope) can be specified. A task with a response message is considered a two-way interaction involving a request from the initiator and a response from the respondent.

Similar to BPMN collaboration diagrams, sequence flow arcs specify order dependencies between tasks. In addition, gateways allow the specification of exclusive and parallel behavior. Sequence flow arcs originating from an exclusive gateway can be associated with a guard that specifies the condition for continuing along that path. Note that all participants affected by the decision must have the same view of the data on which the decision is based [15]. In contrast to process orchestrations, choreographies do not assume a central data store, as each participant typically maintains its data locally. Data can only be exchanged via messages [12]. An example of a choreography diagram is depicted in Fig. 1.

2.2 Interaction Petri Nets

Models facilitate development and the exchange of ideas among experts, yet precise semantics are essential for their implementation and analysis. In business process modeling, *Petri nets* are widely used to provide concise execution semantics [7]. Petri nets consist of places and transitions connected by directed arcs. Places may contain tokens. If all places connected with an incoming arc contain tokens, a transition can be fired to consume tokens from the incoming places and produce tokens in the outgoing places. A distribution of tokens to places is referred to as a *marking* [1]. Decker et al. introduce *Interaction Petri Nets* (IPN) as an extension of Petri nets for describing interaction models [5]. IPNs represent each interaction by a single transition labeled with the initiator, the respondent, and a description of the message. The additional information allows reasoning about enforceability aspects of an interaction model [4]. A firing sequence of transitions represents a conversation. Based on definitions from the literature [5,10], we define interaction Petri nets as follows:

Definition 2. *(Interaction Petri net). An interaction Petri net is defined by a tuple* $\mathcal{I} = (P, T, F, R, init, resp, m_0)$*, where*

- P *is a finite set of places,*
- T *is a finite set of transitions, which can be partitioned into disjoint sets of interactions* T_I*, events* T_E*, and silent transitions* T_S*,*
- $F \subseteq (P \times T) \cup (T \times P)$ *is a finite set of arcs,*
- R *is a finite set of roles,*
- $init : T_I \rightarrow R$ *assigns an initiating role to an interaction transition,*
- $resp : T_I \rightarrow R$ *assigns a respondent to an interaction transition, and*
- $m_0 : P \rightarrow \mathbb{N}$ *assigns the initial number of tokens to each place, thus specifying an initial marking for the net.*

3 Motivating Example

To illustrate the need for concise data semantics for choreographies, in this section we present an example choreography inspired by the shipment of goods

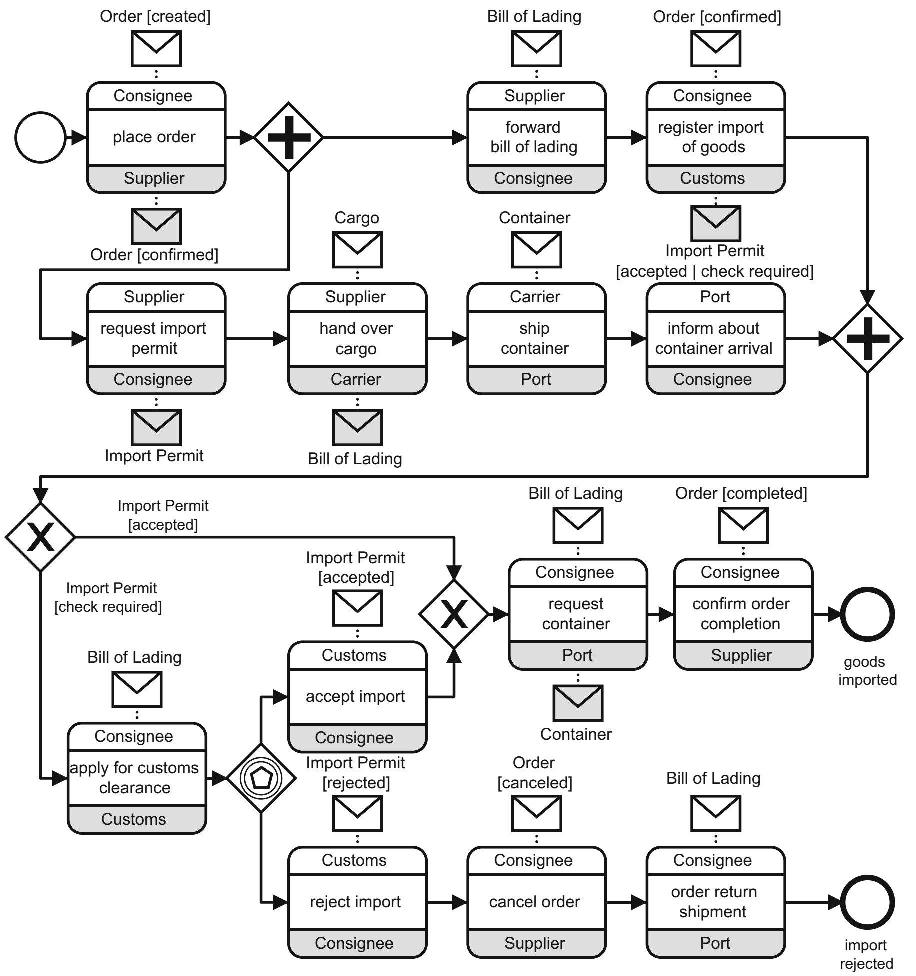


Fig. 1. Choreography diagram describing the international transport of goods by ship between a consignee and a supplier, considering customs.

by sea to a European Union member state. The choreography diagram shown in Fig. 1 illustrates the key steps, including the consignee ordering goods from a supplier, the supplier sending the container of goods to the destination port via a carrier, and customs inspecting the import of the goods. For each order, the supplier requests the import permit from the consignee, which must be issued by customs before the container is handed over to the carrier. Upon arrival, the consignee can access the container using the bill of lading issued by the carrier. If the import permit requires further checks, the consignee must first apply for customs clearance. If customs rejects the import, the container must be returned.

When data is not considered, the choreography diagram meets the local enforceability requirements [4]. However, considering data exchange raises enforceability concerns. In particular, after the order is confirmed by the supplier, the subsequent steps require the supplier to forward the bill of lading and the consignee to provide the import permit. Both documents, however, are issued by the carrier or customs respectively and are sent only in subsequent tasks, resulting in a deadlock. In addition, the BPMN 2.0 standard requires that participants affected by an exclusive gateway must share the same view of the data on which the decision is based on [15]. After the container arrived, the port has not received the import permit and therefore is unaware of whether a return shipment must be expected. Therefore, the design of the choreography raises concerns about data exchange regarding:

– Data awareness: Is the sender of a message aware of the required data?
– Data consistency: Do participants have the same view of the exchanged data?
– Data dependencies: What dependencies exist between the exchanged data?

Since erroneous data flow may not be obvious in complex choreographies, precise specifications and semantics for interorganizational data exchange are required to enable the analysis of process choreographies with data at design time.

4 Execution Semantics for Choreographies with Data

Specifying and analyzing interorganizational data flow requires extending the execution semantics and refining the notion of choreography diagrams. In the following, Sect. 4.1 introduces supplementary models for specifying data exchange in choreography diagrams. In addition, Sect. 4.2 presents data-enhanced interaction Petri nets as a formal basis for defining execution semantics for data exchange. Finally, Sect. 4.3 proposes a mapping of supplemented choreography diagrams to data-enhanced interaction Petri nets to define execution semantics for choreography diagrams with data specifications.

4.1 Data Exchange Specifications for Choreography Diagrams

Choreography diagrams allow only limited specification of data exchange. Message elements can be assigned labels to describe the content of the message, but no clear semantics are provided for the labels, which can lead to different interpretations of the behavior. In this section, we refine choreography diagrams by introducing more concise data exchange specifications for message elements. To remain compliant with the BPMN standard, the notation of the choreography diagrams is not adapted, which facilitates modeling with existing tools. Instead, the specification is composed of supplemental models that, in addition to the choreography diagram, define the behavior and relations between the exchanged data. The additional models consist of a *shared data model* and *distributed object lifecycle models*. Both types of models and their relationship to choreography diagrams are described below.

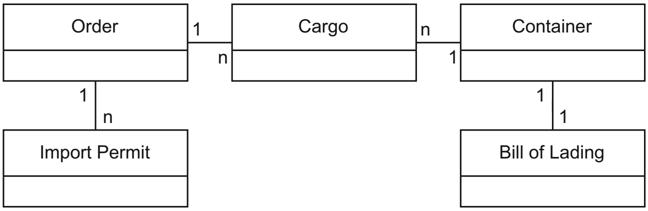


Fig. 2. A shared data model defining the message types and their relations for messages exchanged in the exemplary shipping choreography.

Shared Data Model. A shared data model is used to globally specify the types of messages exchanged in a choreography. Each class represents one message type. An instance of a class, referred to as a *message object*, is considered a unit of data, such as a document, that can be exchanged via a message associated with the corresponding class. While messages can also refer to physical items, such as a container or cargo, message objects refer only to a virtual representation of these items. Consequently, when a message is sent, the message object is not lost to the sender, but both the sender and receiver are aware of the message object.

Definition 3. *(Shared Data Model). A shared data model is defined by a tuple* $\mathcal{D} = (C, \mathcal{R}, mult)$*, where:*

- C *is a non-empty, finite set of classes,*
- $\mathcal{R} \subseteq \{(c_1, c_2) \mid c_1, c_2 \in C \wedge c_1 \neq c_2\}$ *is a symmetric relation of classes,*
- $mult : \mathcal{R} \rightarrow \{(1,1), (1,n), (n,1), (n,m)\}$ *assigns a multiplicity to a relation.*

Similar to the approach proposed by Meyer et al. [12], the data model is shared by all participants so that each participant has the same understanding of the types of messages that can be exchanged. Figure 2 illustrates a shared data model for the exemplary shipping choreography. To reference the message types in the choreography diagram, the message elements of tasks can be annotated with the label of the class, as illustrated in Fig. 1.

As stated in Definition 3, the shared data model also defines relations with multiplicities between classes. To limit complexity, we consider only *one-to-one* $(1,1)$, *many-to-one* $(n,1)$, and *many-to-many* (n,m) relations, where it is expected that m and n can be zero. Relations can constrain the creation of message objects by implying dependencies. For example, according to Fig. 2, an 'Import Permit' cannot exist without an existing 'Order' due to their many-to-one relation: $mult((ImportPermit, Order)) = (n,1)$. Therefore, an 'Import Permit' object can only be instantiated if an 'Order' object already exists, which limits the possible behavior of the choreography. Accordingly, 'Container' and 'Bill of Lading' message objects must be created simultaneously to ensure the one-to-one relation. Many-to-many relations do not affect the creation of message objects, since it is expected that the objects can exist individually, given the assumption that n and m can be zero.

Distributed Object Lifecycle. During a conversation, message objects can be created or their contents changed. Similar to data states of data objects in BPMN process diagrams [15], we use *message states*, hereinafter referred to as *states*, to reflect the content of a message object. A state provides an abstract view of the content of a message object that is relevant to the business case under consideration. For example, an 'Order' message object can be in the states 'created', 'confirmed', 'completed', or 'canceled'. The mapping between states and actual attribute values is beyond the scope of this paper. Given a class $c \in C$, S_c denotes the set of possible states that a message object of c can be in. We refer to a message object of class $c \in C$ in a particular state $s \in S_c$ by the notion $c[s]$. Each class is considered to have at least one state.

To describe the possible states and allowed state transitions for a class in a choreography, we introduce *distributed object lifecycles*. As stated in Definition 4, distributed object lifecycles extend object lifecycles by specifying which role can perform which state transitions, since in choreographies message objects may be manipulated by different participants. For example, a consignee can create an 'Order' message object in the state 'created' but only the supplier can change the state to 'confirmed' as illustrated in Fig. 3. Each state is represented by a label in a circle, and allowed state transitions are represented by directed arcs associated with roles that can perform the transition. In addition to defining state transitions, distributed object lifecycles also constrain message object creation, since objects can only be created in initial states associated with an ingoing arc without an originating state. Furthermore, only roles associated with an initial state can create new message objects in the corresponding state. Similar to the shared data model, distributed object lifecycles are available to all participants.

Definition 4. *(Distributed Object Lifecycle). Let $\mathcal{S}$ be the universe of all possible states, a distributed object lifecycle of a class $c \in C$ is a finite state machine defined by a tuple $\mathcal{L}_c = (S_c, S_c^i, \delta_c, R, role)$, where:*

- *$S_c \subseteq \mathcal{S}$ is a non-empty, finite set of states associated with c,*
- *$S_c^i \subseteq S_c$ is a non-empty, finite set of initial states,*
- *$\delta_c \subseteq S_c \times S_c$ is a finite set of state transitions,*
- *R is a non-empty, finite set of roles, and*
- *$role : S_c^i \cup \delta_c \rightarrow R$ assigns a role to an initial state or a state transition.*

A message object can only be in one state for one participant at a time. It is assumed that if multiple participants are aware of a message object in the same state, these participants have the same view of its content. However, it should be noted that creating or modifying message objects is a local operation. Only when a message object is sent via a task, the respondent will receive the message object in the corresponding state. As a result, participants may have different views of a message object during a conversation.

States can also serve as constraints on interactions, since some interactions may require a message object to be sent in a particular state. To incorporate the constraints into choreography diagrams, states can be added to the class specifications of the messages using the notion $c[s]$ mentioned above. If a task

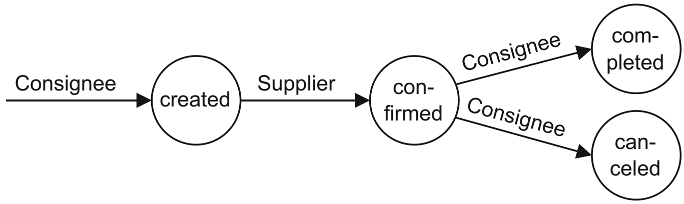


Fig. 3. A distributed object lifecycle that describes the allowed state transitions for message objects of the 'Order' class during a conversation. Annotations on arcs specify the roles that can perform the transitions.

accepts a message object in multiple states, all allowed states can be listed following the notion $c[s_1|...|s_n]$. Therefore, considering Fig. 1, the response of the task 'register import of goods' allows the sending of an 'Import Permit' message object in the state 'accepted' or 'check required'. The constraint implies further that the sender of the message must be aware of the message object in a corresponding state. If no state is specified, all possible states are accepted.

Furthermore, states can be used as guards for paths that follow exclusive gateways, as shown in Fig. 1. Thus, given a choreography $\mathcal{C}$, a guard is expected to specify a message object in an appropriate state required to continue with the associated path: $\mathcal{G} \subseteq C \times S$. For exclusive gateways, it is essential that all participants affected by the gateway have the same view of the data on which the decision is based. The identification of affected participants is discussed in more detail in Sect. 4.3.

The supplemental models introduced in this section can be used to specify exchanged data and data dependencies. However, to ensure that a choreography maintains data awareness and data consistency at design time, concise execution

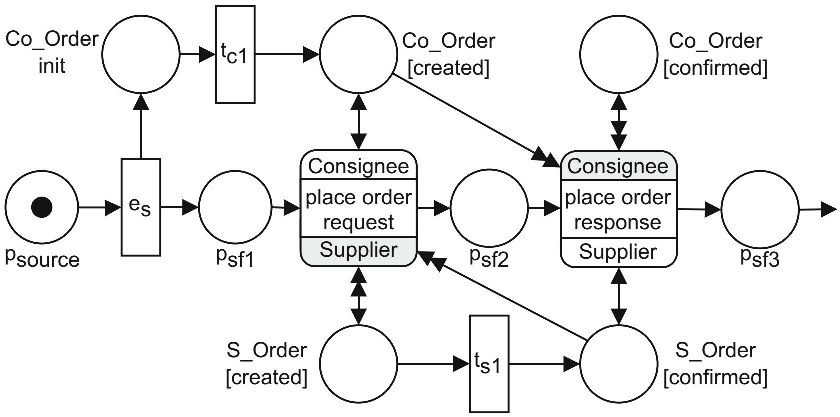


Fig. 4. Data-enhanced interaction Petri net depicting an excerpt of the shipping choreography including the start event and a two-way choreography task.

semantics are required. The next section introduces an extension to interaction Petri nets, providing a formal basis to define execution semantics for the data exchange specifications.

4.2 Data-Enhanced Interaction Petri Nets

The formal representation of data exchange in choreographies requires the extension of IPNs with additional elements. The extended net, referred to as *data-enhanced interaction Petri net*, is defined as follows:

Definition 5. *(Data-enhanced Interaction Petri Net). A data-enhanced interaction Petri net is defined by a tuple* $\mathcal{I}' = (P', T', F', RF, R, init, resp, m_0)$*, where*

- P' *is a finite set of places that can be partitioned into disjoint sets of control flow places* P*, message places* P_M*, and message initialization places* P_{MI}*,*
- T' *is a finite set of transitions that can be partitioned into disjoint sets of interactions* T_I*, events* T_E*, silent control flow transitions* T_S*, and silent message modification transitions* T_M*,*
- $F' \subseteq (P' \times T') \cup (T' \times P')$ *is a finite set of arcs that can be partitioned into sequence flow arcs* F *and message flow arcs* F_M*,*
- $RF \subseteq P_M \times T_I$ *is a finite set of reset arcs, and*
- $(P, T_I \cup T_E \cup T_S, F, R, init, resp, m_0)$ *is an interaction Petri net.*

The extension supports the representation of message objects using additional message places P_M. Each message place is dedicated to a participant and a message object in a particular state. A token in a message object place indicates that the corresponding participant is aware of the message object. To limit the creation of message objects, initialization places P_{MI} are introduced. Additional silent transitions T_M allow local creation and modification of message objects according to the message flow arcs F_M. Message flow arcs can also connect message places with interaction transitions to specify data exchange.

In addition, *reset arcs RF* are introduced, which set the number of tokens at all associated places to zero once the corresponding transition fires. Reset arcs do not constrain the firing of transitions. Thus, transitions can be fired even if they are connected to an empty place by a reset arc [1]. In the model, reset arcs are depicted with double arrowheads.

An example of a data-enhanced IPN is shown in Fig. 4. Similar to [10], the notion of interaction transitions is adapted to the contemporary notion of choreography tasks. The initiator of a task is indicated by a white badge and the respondent by a gray badge.

4.3 From Choreographies to Data-Enhanced Interaction Petri Nets

To define the execution semantics for data exchange in choreographies, we map choreography diagrams, supplemented with a shared data model and distributed data object lifecycles, to data-enhanced interaction Petri nets. The mapping requires preprocessing the choreography diagram so that each task containing a response message is split into two tasks connected by a sequence flow arc. The result is semantically equivalent, with the first task representing the initial message and the second task representing the response. Consequently, after preprocessing, each task is associated with only one or zero message elements. We define the auxiliary function $obj : \mathcal{T} \rightarrow 2^{(C \times \mathcal{S})}$ to map each task to the corresponding class and states of the allowed message objects, as specified by the message element labels.

Hence, given a process choreography $\mathcal{C} = (N, SF, R, M, \mathcal{G}, grd, init, resp, msg)$, a shared data model $\mathcal{D} = (C, \mathcal{R}, mult)$, and a distributed object lifecycle $\mathcal{L}_c = (S_c, S_c^i, \delta_c, R, role)$ for each class $c \in C$, the models can be mapped to a data-enhanced interaction Petri net $\mathcal{I}' = (P', T', F', RF, R, init, resp, m_0)$ to represent the execution semantics as follows:

The set of roles R is taken from the choreography diagram. Correspondingly, the functions $init$ and $resp$ map the same roles for an interaction transition T_I as for the corresponding interaction $\mathcal{T}$ in the choreography. The mapping of the control flow semantics essentially follows the mappings provided in [5, 10]. However, unlike their mappings, multiple transitions are created for tasks that allow sending message objects in different states, since each transition is intended to transfer only one message object in one state, as depicted in Fig. 5. Since the additional transitions represent alternative executions of the same task with the same initiator and respondent, the extension does not affect local enforceability constraints. The set of silent transitions T_S is divided into disjoint

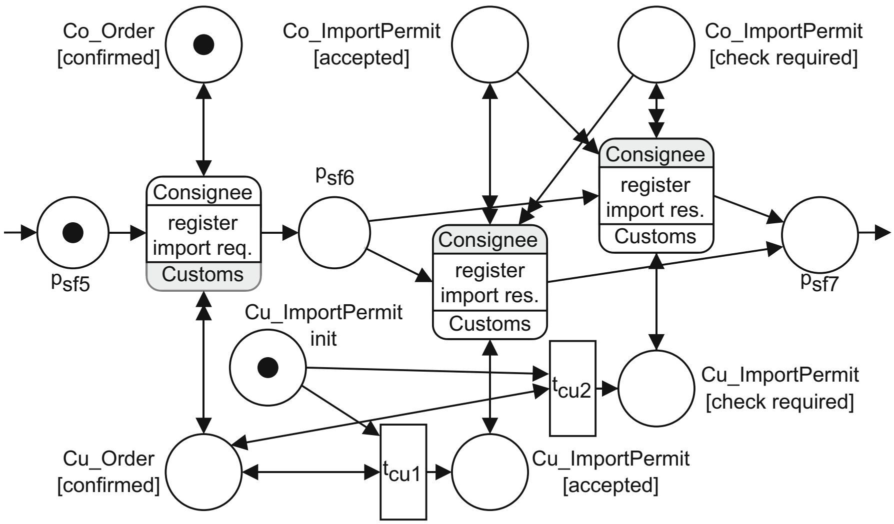


Fig. 5. Excerpt from a data-enhanced interaction Petri net representing the shipping example, illustrating the creation and sending of an 'Import Permit' message object in one of the allowed states using a predefined marking.

sets of exclusive gateways $T_\times$ and parallel gateways T_+. The mapping of the control flow to transitions and places is defined as follows:

$$
\begin{aligned}
P &= \{p_{source}, p_{sink}\} \cup \{p_{(n_1,n_2)} \mid (n_1,n_2) \in SF \wedge n_1 \notin G^s_e\}\\
T_E &= \{t_e \mid e \in E\}\\
T_I &= \{t_i \mid i \in \mathcal{T} \wedge obj(i) = \emptyset\} \cup \{t_{(i,c,s)} \mid i \in \mathcal{T} \wedge (c,s) \in obj(i)\}\\
T_\times &= \{t_{(g,n)} \mid g \in G^s_\times \wedge (g,n) \in SF\} \cup \{t_{(n,g)} \mid g \in G^j_\times \wedge (n,g) \in SF\}\\
T_+ &= \{t_g \mid g \in G^s_+ \cup G^j_+\}\\
m_0 &= \{[p_{source}]\}
\end{aligned}
$$

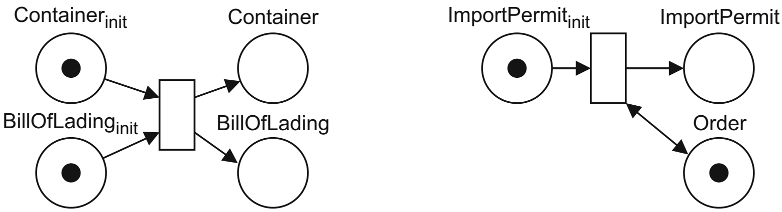
The mapping adds a source and a sink place that represent the start and end of a conversation. Only the source place contains a token in the initial marking. In addition, transitions and places are connected by arcs, enforcing the semantics of sequence flows and gateways:

$$
\begin{aligned}
F = {} & \{(p_{source}, t_e) \mid e \in E_s\} \cup \{(t_e, p_{sink}) \mid e \in E_e\} \cup\\
& \{(p_{(n_1,n_2)}, t_{n_2}) \mid (n_1,n_2) \in SF \wedge n_1 \notin G^s_e \wedge\\
& \;(n_2 \in (E_e \cup G^s_+ \cup G^j_+) \vee (n_2 \in \mathcal{T} \wedge obj(n_2) = \emptyset))\} \cup\\
& \{(p_{(n,i)}, t_{(i,c,s)}) \mid (n,i) \in SF \wedge n \notin G^s_e \wedge i \in \mathcal{T} \wedge (c,s) \in obj(i)\} \cup\\
& \{(t_{n_1}, p_{(n_1,n_2)}) \mid (n_1,n_2) \in SF \wedge (n_1 \in (E_s \cup G^s_+ \cup G^j_+) \vee\\
& \;(n_1 \in \mathcal{T} \wedge obj(n_1) = \emptyset))\} \cup\\
& \{(t_{(i,c,s)}, p_{(i,n)}) \mid (i,n) \in SF \wedge i \in \mathcal{T} \wedge (c,s) \in obj(i)\} \cup\\
& \{(p_{(n_1,g)}, t_{(g,n_2)}) \mid (n_1,g) \in SF \wedge (g,n_2) \in SF \wedge g \in G^s_\times\} \cup\\
& \{(t_{(g,n)}, p_{(g,n)}) \mid (g,n) \in SF \wedge g \in G^s_\times\} \cup\\
& \{(p_{(n,g)}, t_{(n,g)}) \mid (n,g) \in SF \wedge g \in G^j_\times\} \cup\\
& \{(t_{(n_1,g)}, p_{(g,n_2)}) \mid (n_1,g) \in SF \wedge (g,n_2) \in SF \wedge g \in G^j_\times\} \cup\\
& \{(p_{(n_1,g)}, t_{n_2}) \mid (n_1,g) \in SF \wedge (g,n_2) \in SF \wedge g \in G^s_e\}
\end{aligned}
$$

In the following, the mapping is extended with message places to incorporate data semantics. For each class in the shared data model, an initialization place P_{MI} is added to ensure that only one instance of a message object is created during conversation. In addition, message places are introduced for each class and state combination for each participant:

$$
\begin{aligned}
P_{MI} &= \{p_c \mid c \in C\}\\
P_M &= \{p_{(r,c,s)} \mid r \in R \wedge c \in C \wedge s \in S_c\}
\end{aligned}
$$

Silent transitions are introduced to create and modify message objects and their state. The creation of message objects may be subject to constraints due to



(a) One-to-One (b) Many-to-One

Fig. 6. Creation of message objects in data-enhanced interaction Petri nets considering different multiplicity constraints.

relations as specified in Sect. 4.1. Hence, the mapping enforces these constraints by following the rules depicted in Fig. 6. The rules ensure that for one-to-one relations, the message objects are created at the same time, and for many-to-one relations, the related message object must exist for creation, as illustrated in Fig. 5 for the creation of an 'Import Permit' message object. For this purpose, two auxiliary functions are introduced. $rel_{(1,1)} : C \to 2^C$ associates each class with a set of classes that have one-to-one relations to the given class, including the given class itself, while also considering transitive relations. Correspondingly, we define a function $rel_{(n,1)} : C \to 2^C$ which returns a set of classes having a many-to-one relation to the given class, so that $\forall c_1, c_2 \in C : c_1 \in rel_{(n,1)}(c_2) \iff mult((c_1, c_2)) = (n, 1)$.

Classes with a one-to-many relation to the given class may have multiple states. Hence, multiple transitions are required to create a message object. Given a class $c \in C$, we define the set of all class and state combinations having a many-to-one relation to c or to a class having a one-to-one relation with c as follows:

$$CS_c^{(n,1)} = \{(c', s) \mid \exists c'' \in rel_{(1,1)}(c) : c' \in rel_{(n,1)}(c'') \land s \in S_{c'}\}$$

Since a class can only be in a single state for a participant, we define the set of relation dependencies including sets of all possible combinations of classes and states having a corresponding many-to-one relation to c as follows:

$$RD_c = \{CS \mid CS \subseteq CS_c^{(n,1)} \land \forall (c', s') \in CS_c^{(n,1)} : \exists! s \in S_{c'} : (c', s) \in CS\}$$

Hence, the manipulation of message objects is represented by a set of silent message modification transitions T_M:

$$\begin{aligned} T_M = \{&t_{(r,C',RD,s)} \mid c \in C \land C' = rel_{(1,1)}(c) \land RD \in RD_c \land s \in S_c^i \land r = role(s)\} \cup \\ &\{t_{(r,c,s_1,s_2)} \mid c \in C \land (s_1, s_2) \in \delta_c \land r = role((s_1, s_2))\} \end{aligned}$$

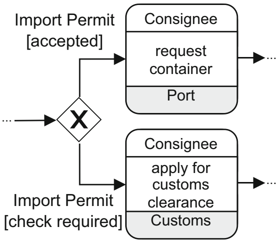
The additional nodes need to be connected with arcs to represent the respective creation and state change behaviors. The set of message flow arcs F_M can be partitioned into disjoint sets of message object manipulation arcs F_M^O, interaction dependency arcs F_M^I, and exclusive gateway dependency arcs $F_M^\times$. The set of message object manipulation arcs F_M^O connects message places P_M and initialization places P_{MI} with message modification transitions T_M according to the shared data model and distributed object lifecycles:

$$
\begin{aligned}
F_M^O = {} & \{(t_e, p_c) \mid e \in E_s \wedge c \in C\} \cup \\
& \{(p_c, t_{(r,C',RD,s)}) \mid c \in C \wedge C' = rel_{(1,1)}(c) \wedge RD \in RD_c \wedge s \in S_c^i \\
& \ \wedge r = role(s)\} \cup \\
& \{(t_{(r,C',RD,s)}, p_{(r,c,s)}) \mid c \in C \wedge C' = rel_{(1,1)}(c) \wedge RD \in RD_c \wedge s \in S_c^i \wedge \\
& \ r = role(s)\} \cup \\
& \{(p_{(r,c,s)}, t_{(r,C',RD,s')}) \mid RD \in RD_c \wedge c, c' \in C \wedge C' = rel_{(1,1)}(c') \wedge c \notin C' \wedge \\
& \ (c, s) \in RD \wedge s' \in S_{c'}^i \wedge r = role(s')\} \cup \\
& \{(t_{(r,C',RD,s')}, p_{(r,c,s)}) \mid RD \in RC_c \wedge c, c' \in C \wedge C' = rel_{(1,1)}(c') \wedge c \notin C' \wedge \\
& \ (c, s) \in RD \wedge s' \in S_{c'}^i \wedge r = role(s')\} \cup \\
& \{(p_{(r,c,s_1)}, t_{(r,c,s_1,s_2)}) \mid c \in C \wedge (s_1, s_2) \in \delta_c \wedge r = role((s_1, s_2))\} \cup \\
& \{(t_{(r,c,s_1,s_2)}, p_{(r,c,s_2)}) \mid c \in C \wedge (s_1, s_2) \in \delta_c \wedge r = role((s_1, s_2))\}
\end{aligned}
$$

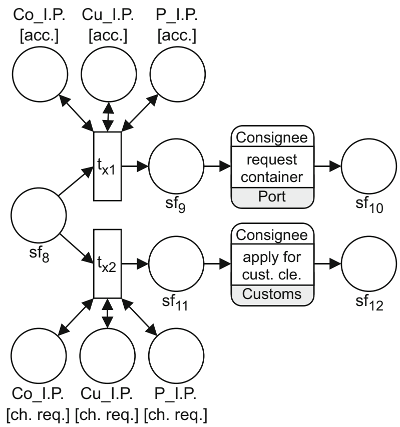
Furthermore, message places are associated with interaction transitions to represent the data transfer. For this purpose, each interaction transition must read (i.e., consume and produce) the token from a message place of the task initiator with the appropriate class and state to ensure that the initiator is aware of the data to be sent as defined in F_M^I. Thus, if a message sender is unaware of the required data during a conversation, the transition cannot fire. In the other case, when the transition is fired, a token is produced in the appropriate message place of the receiver. In addition, reset arcs RF are added to reset any message place of the same class as the exchanged message object on the receiver side to prevent participants from having message objects of the same class in multiple states, as depicted in Fig. 4 and Fig. 5.

$$
\begin{aligned}
F_M^I = {} & \{(p_{(r,c,s)}, t_{(i,c,s)}) \mid i \in \mathcal{T} \wedge r = init(i) \ \wedge (c, s) \in obj(i)\} \cup \\
& \{(t_{(i,c,s)}, p_{(r,c,s)}) \mid i \in \mathcal{T} \wedge \ r \in \{init(i), resp(i)\} \wedge (c, s) \in obj(i)\} \\
RF = {} & \{(p_{(r,c,s)}, t_{(i,c,s')}) \mid i \in \mathcal{T} \wedge r = resp(i) \wedge (c, s') \in obj(i) \wedge s \in S_c\}
\end{aligned}
$$

Finally, according to the semantics of exclusive gateways, it must be ensured that all affected participants have the same view of the data on which the decision is based [15]. To enforce semantics in a data-enhanced IPN, silent transitions representing the decision are required to read the appropriate message places of all affected participants, as illustrated in Fig. 7. Inconsistencies in the participants' data would result in transitions associated with the gateway not being able to fire, thus ensuring data consistency for a firing sequence.



(a) Choreography diagram



(b) Data-enhanced interaction Petri net

Fig. 7. Exclusive gateway deciding on subsequent path based on state of the import permit (I.P.) represented as choreography diagram (a) and data-enhanced interaction Petri net (b).

To narrow down the affected participants for a decision, we refer to the concept of *single-entry single-exit* (SESE) regions [8]. A SESE region includes all elements on a path between an exclusive gateway split and an exclusive gateway join or end event, with all paths originating from the split leading to either a join or end events. In the case of a loop, all paths but one may lead back to the initial split. Hence, given the example in Fig. 1, since not all paths originating from the exclusive gateway split lead to an exclusive gateway join, the SESE region extends until the end of the choreography. We argue that all participants involved in tasks in a SESE region started by an exclusive split can be considered affected, since execution within the region depends on the initial decision. The behavior after the SESE region is independent of the decision and can be neglected. Therefore, we define the auxiliary function $sese_\times : G^s_\times \rightarrow 2^R$, which returns the set of participants involved in tasks in a SESE region started by a given exclusive split.

$$
\begin{aligned}
F_M^\times = &\{(p_{(r,c,s)}, t_{(g,n)}) \mid g \in G^s_\times \wedge (g,n) \in SF \wedge r \in sese_\times(g) \wedge (c,s) = grd((g,n))\} \cup \\
&\{(t_{(g,n)}, p_{(r,c,s)}) \mid g \in G^s_\times \wedge (g,n) \in SF \wedge r \in sese_\times(g) \wedge (c,s) = grd((g,n))\}
\end{aligned}
$$

Consequently, the mapping inherently enforces that each sender must be aware of the message objects to be sent and that decisions can only be made if the affected participants are aware of the corresponding message object in a consistent state. In addition, dependencies within and between message objects are reflected according to the specifications of the shared data model and distributed object lifecycles. Thus, the approach provides a foundation for specifying and analyzing data exchange in process choreographies.

5 Discussion

In the following, the limitations of the presented extension are discussed. First, message objects and states provide only an abstract view of the exchanged data to focus on the aspects relevant to the business case. While this allows a conceptual design of the data exchange, the mapping between states and actual attribute values requires further research. Furthermore, while the supplementary models allow the definition of dependencies for data specified in choreography diagrams, the approach relies on the consistency between all involved models. Therefore, the development of techniques to automatically verify the consistency between these models can facilitate the design. In addition, the extension does not support multi-instance tasks, messages, or participants. The impact of multi-instance behavior on the semantics of data exchange remains to be explored. Although intermediate events are not supported, their inclusion requires only minor extensions, which are not considered due to space limitations.

The presented extension serves as a foundation for the analysis of data exchange in choreographies. Although the detection and classification of data exchange errors is beyond the scope of this work, potential errors may already be revealed by detecting deadlocks in the possible firing sequences [2]. Since the execution semantics require the initiator of a task to be aware of the message object to be sent, data awareness can be addressed in this way. Correspondingly, exclusive gateway splits can only be executed if the participants have a consistent view of the data, which addresses the data consistency concern. However, since message places may still contain tokens after a conversation terminated, a classical soundness analysis is not applicable to the presented approach [1]. Nevertheless, the extension provides a more concise specification of the data exchange in choreographies, allowing a more in-depth analysis of the message flow.

6 Related Work

While formal execution semantics for interaction models already exist in the literature, most work focuses on the ordering of interactions. Decker et al. defines the execution semantics for the interaction-centric modeling language Let's Dance using π-calculus [6]. In addition, interaction Petri nets, introduced in [5], describe the behavior of iBPMN choreographies. Najem et al. provide a mapping of choreography diagrams to colored Petri nets, which allows detecting deadlocks in the control flow of choreographies [13]. Furthermore, Corradini et al. use a Backus Normal Form syntax to check the conformance between BPMN 2.0 choreography and collaboration diagrams [3]. However, these works neglect the role of data. Based on interaction Petri nets, our work aims to extend the execution semantics of choreographies with a data perspective.

The data exchange between processes in a choreography is investigated by Meyer et al. [12]. The authors introduce a model-driven approach to enable an automated data exchange in choreographies using a global data model. While a mapping between local and global data allows reasoning about data dependencies, data awareness is not addressed. Knuplesch et al. extend the notion of BPMN choreography diagrams with virtual data objects as variables for routing conditions [10]. The authors combine interaction Petri nets and workflow nets with data. However, data dependencies are not considered. Furthermore, Nikaj et al. propose a RESTful representation for choreography diagrams [14]. Since participants must be aware of URLs to invoke them, the approach takes data awareness into account. Nevertheless, data dependencies are neglected.

The decoupling of message and data flow is discussed by Hahn et al. [9]. The authors introduce a middleware to coordinate the propagation of changes to shared data objects used in the local processes of collaborating participants. In contrast to our work, the approach relies on interconnection models and requires insight into the local data flow of the participants. Finally, Köpke et al. propose an approach to model the data flow of interorganizational process models starting from a global process model assuming a central data store [11]. In a second step, the data flow is then distributed among the participants. Due to the initial holistic view, the correctness of the data flow can be ensured with existing techniques. However, unlike our work, the approach requires detailed insight into the organization-internal process behavior of participants, which may complicate collaboration with untrusted organizations.

7 Conclusion

This paper proposes a novel way to describe data exchange in BPMN choreography diagrams by using a shared data model and distributed object lifecycles as supplemental models to define data relations and behavior. In addition, we extended interaction Petri nets with a data perspective to provide a formal basis for defining the execution semantics of data exchange in choreographies. Finally, a mapping of supplemented choreography diagrams to data-enhanced interaction Petri nets is provided, allowing the analysis of the data flow of choreography diagrams. The approach addresses the need for more concise semantics for data exchange to identify data-related flaws in interaction behavior at design time.

For future research, we plan to develop tools to facilitate the modeling and analysis of data exchange in choreographies, as well as a method for verifying the consistency of local and global data flow given a local process model. In addition, we plan to extend our mapping to support multi-instance behavior and more complex relations among message objects, and aim to uncover patterns and antipatterns in interorganizational data exchange. Despite the potential extensions, our proposal already allows for a more precise specification of data exchange for choreographies.

Acknowledgement. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 450612067

References

1. van der Aalst, W.M.P., et al.: Soundness of workflow nets: classification, decidability, and analysis. Formal Aspects Comput. **23**(3), 333–363 (2011). https://doi.org/10.1007/s00165-010-0161-4
2. Chu, F., Xie, X.: Deadlock analysis of Petri nets using siphons and mathematical programming. IEEE Trans. Robot. Autom. **13**(6), 793–804 (1997). https://doi.org/10.1109/70.650158
3. Corradini, F., Morichetta, A., Polini, A., Re, B., Tiezzi, F.: Collaboration vs. choreography conformance in BPMN. Log. Methods Comput. Sci. **16**(4) (2020)
4. Decker, G., Weske, M.: Local enforceability in interaction petri nets. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 305–319. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_22
5. Decker, G., Weske, M.: Interaction-centric modeling of process choreographies. Inf. Syst. **36**(2), 292–312 (2011). https://doi.org/10.1016/j.is.2010.06.005
6. Decker, G., Zaha, J.M., Dumas, M.: Execution semantics for service choreographies. In: Bravetti, M., Núñez, M., Zavattaro, G. (eds.) WS-FM 2006. LNCS, vol. 4184, pp. 163–177. Springer, Heidelberg (2006). https://doi.org/10.1007/11841197_11
7. Dijkman, R.M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in BPMN. Inf. Softw. Technol. **50**(12), 1281–1294 (2008). https://doi.org/10.1016/j.infsof.2008.02.006
8. Dumas, M., García-Bañuelos, L., Polyvyanyy, A.: Unraveling unstructured process models. In: Mendling, J., Weidlich, M., Weske, M. (eds.) BPMN 2010. LNBIP, vol. 67, pp. 1–7. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16298-5_1
9. Hahn, M., Breitenbücher, U., Kopp, O., Leymann, F.: Modeling and execution of data-aware choreographies: an overview. Comput. Sci. Res. Dev. **33**(3–4), 329–340 (2018). https://doi.org/10.1007/s00450-017-0387-y
10. Knuplesch, D., Pryss, R., Reichert, M.: Data-aware interaction in distributed and collaborative workflows: modeling, semantics, correctness. In: CollaborateCom 2012, Pittsburgh, PA, USA, October 14–17, 2012, pp. 223–232. ICST/IEEE (2012). https://doi.org/10.4108/icst.collaboratecom.2012.250443
11. Köpke, J., Franceschetti, M., Eder, J.: Optimizing data-flow implementations for inter-organizational processes. Distrib. Parallel Databases **37**(4), 651–695 (2019). https://doi.org/10.1007/s10619-018-7251-3
12. Meyer, A., Pufahl, L., Batoulis, K., Fahland, D., Weske, M.: Automating data exchange in process choreographies. Inf. Syst. **53**, 296–329 (2015). https://doi.org/10.1016/j.is.2015.03.008
13. Najem, T., Perucci, A.: Mapping BPMN2 service choreographies to colored petri nets. In: Camara, J., Steffen, M. (eds.) SEFM 2019. LNCS, vol. 12226, pp. 85–100. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57506-9_8
14. Nikaj, A., Weske, M.: Formal Specification of RESTful choreography properties. In: Bozzon, A., Cudre-Maroux, P., Pautasso, C. (eds.) ICWE 2016. LNCS, vol. 9671, pp. 365–372. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38791-8_21
15. Object Management Group (OMG): business process model and notation (BPMN), Version 2.0.2 (2014). https://www.omg.org/spec/BPMN/2.0.2/

Large Language Models for Business Process Management: Opportunities and Challenges

Maxim Vidgof[1(✉)], Stefan Bachhofner[1], and Jan Mendling[1,2,3]

[1] Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria
{maxim.vidgof,stefan.bachhofner}@wu.ac.at
[2] Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany
jan.mendling@hu-berlin.de
[3] Weizenbaum Institute, Hardenbergstraße 32, 10623 Berlin, Germany

Abstract. Large language models are deep learning models with a large number of parameters. The models made noticeable progress on a large number of tasks, and as a consequence allowing them to serve as valuable and versatile tools for a diverse range of applications. Their capabilities also offer opportunities for business process management, however, these opportunities have not yet been systematically investigated. In this paper, we address this research problem by foregrounding various management tasks of the BPM lifecycle. We investigate six research directions highlighting problems that need to be addressed when using large language models, including usage guidelines for practitioners.

Keywords: Natural language processing · Large language models · Generative Pre-Trained Transformer · Deep learning · Research Challenges

1 Introduction

Recent releases of applications building on Large Language Model (LLM) have been quickly adopted by large circle of users. ChatGPT stands out with reaching 100 million users in 2 months [31]. The key factor explaining this fast uptake is their general applicability making them a general-purpose technology. Also many tasks in research can be approached with LLM applications, include finding peer

This research received funding from the Teaming.AI project, which is part of the European Union's Horizon 2020 research and innovation program under grant agreement No 957402. The research by Jan Mendling was supported by the Einstein Foundation Berlin under grant EPP-2019-524 and by the German Federal Ministry of Education and Research under grant 16DII133.
M. Vidgof and S. Bachhofner—Equal contribution.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 107–123, 2023.
https://doi.org/10.1007/978-3-031-41623-1_7

reviewers, evaluating manuscripts and grants, improving prose in manuscripts, and summarizing texts [32]. For this reason, some argue that LLM – especially conversational LLM – are a *"game-changer for science"* [6].

Much of the current discussion of applications like ChatGPT is concerned with the question how good it works now and in the future. We believe that this question needs to be approached with a clearly defined task in mind. Starting with a task focus will move the discussion away from funny or disturbing errors and biases [31] towards how the collaboration between human experts and LLM applications can be organized. Furthermore, this bears the chance to learn about specific categories of failures, which eventually will help to refine the technology in a systematic way.

In this paper, we address the research challenge of how LLM applications can be integrated at different stages of business process management. To this end, we refer to the BPM lifecycle [8] and its various management tasks [13]. Our research approach is exploratory in a sense that we developed strategies of how LLM applications can be integrated in specific BPM tasks. We observe various promising usage scenarios and identify challenges for future research.

The paper is structured as follows. Section 2 discusses the essential concepts of Deep Learning (DL) and LLM in relation to Business Process Management (BPM) practises. In Sect. 3 we identify and discuss LLM applications within BPM and along the different BPM lifecycle phases. Based on these applications, Sect. 4 describes six core research directions ranging from how LLM change the dynamics and execution of BPM projects, to data sets, and benchmarks specific to BPM. Section 5 identifies challenges when using LLM. Furthermore, we provide an outlook on how LLM might evolve in the future.

2 Background

The advent of LLM applications paves the way towards a plethora of new BPM-related applications. So far, BPM has adopted natural language processing [1], artificial intelligence [7], and knowledge graphs [14] to support various application scenarios. In this section, we discuss the foundations of DL (Sect. 2.1) and LLMs (Sect. 2.2). In this way, we aim to clarify their specific capabilities.

2.1 Deep Learning

Recent LLM applications build on machine learning and deep learning models, such as recurrent neural networks (RNNs) and transformer networks. Machine Learning (ML) studies algorithms that are *"capable of learning to improve their performance of a task on the basis of their own previous experience"* [15]. In essence, ML techniques use either supervised learning, unsupervised learning, or reinforcement learning as a paradigm. Several of them are relevant for LLM.

In *supervised learning*, the ML algorithm receives as an input a collection of pairs, where one pair consists of features representing a concept, along with a label. Importantly, this label is task specific and encodes what the algorithm should learn about the concepts. Such labels can be, for instance, *spam* and *no spam* for a spam classifier, or bounding boxes with annotations for an image. There are two cases of supervised learning that are relevant for LLM: *few-shot* and *zero-shot* learning. *Few-shot* learning is when a ML algorithm adapts to a new situation with little amount of labelled data, and *zero-shot* learning is when the algorithm can do this with no labelled data at all. For example, a language model can be provided with a few input-output pairs, and the model can inverse the mapping function without any parameter changes. In *unsupervised learning*, the algorithm only receives a feature tensor of a concept as an input and the desired output is unknown. The algorithm then finds structural properties of the concepts present in the feature tensor. A typical application is dimensionality reduction, for instance using auto-encoders. In *reinforcement learning*, the algorithm receives a feature tensor of a concept as an input for which an output is produced, which is then evaluated through rewards. The algorithm then uses this feedback to improve its parameters. ChatGPT uses a form of reinforcement learning known as deep reinforcement learning to improve its language generation capabilities, in particular, *"Learning to summarize from human feedback"* [29]. ChatGPT is fine-tuned using a reward signal that assesses the quality of its generated responses, with the goal of maximizing the reward signal over time. The model's ability to learn from the reward signal allows it to generate increasingly relevant and coherent responses.

Deep learning (DL) is a ML method based on Neural Networks (NN). In general, they are NNs with many layers stacked on top each other, which enables them to learn multiple layers of representations [10]. Importantly, these representations can be learned without supervision. Networks with only one hidden layer are called shallow. Deep networks are able to handle more complex problems compared to shallow networks. Combined with the availability of large amounts of data, improvements on how to speed up the optimization, and powerful computing resources, enables them to be trained effectively. In the context of natural language processing, deep learning has been particularly effective in tasks such as machine translation, sentiment analysis, and named entity recognition. The ability of deep learning to learn multiple layers of representations from input data has proven to be particularly powerful for these tasks. This is because natural language processing involves dealing with sequences of words and characters, and the relationships between these sequences are often complex and multi-layered. The use of large amounts of labeled training data and powerful computational resources has enabled deep learning models to achieve state-of-the-art results in many Natural Language Processing (NLP) tasks. For example, the transformer architecture, introduced in the paper *"Attention is All You Need"* by Vaswani et al. [33] has become the standard architecture for many NLP tasks, including language translation and language modeling.

In recent years, BPM research has integrated the capabilities of deep learning to a large extent for process prediction. For an overview, see [16]. There are also recent applications for automatic process discovery [28], for generating process models from hand-written sketches [27], and for anomaly detection [17].

2.2 Large Language Models

LLM are DL models trained on vast amounts of text data to perform various natural language processing tasks. These models, which typically range from hundreds of millions to billions of parameters, are designed to capture the complexities and nuances of human language. The largest models, such as GPT-1 and GPT-3, are capable of generating human-like text, answering questions, translating languages, and computer code. The training process of these models involves processing massive amounts of text data, which is used to learn patterns and relationships between words and phrases. These models then use this information to predict the likelihood of a given token, or sequence of tokens, in a specific context. This allows them to generate coherent and contextually relevant text or perform other language-related tasks. The rise of large language models has resulted in significant advancements in the field of NLP, and they are widely used in various applications, including chatbots, virtual assistants, and text generation systems. One of their strengths is their ability to perform *few-shot* and *zero-shot* learning with prompt-based learning [11].

In 2018, Radford et al. introduced GPT-1 (also sometimes called simply GPT) in their paper on *"Improving language understanding by generative pre-training"* [23]. Generative Pre-trained Transformer (GPT)-1 refers to the largest model the authors have trained (110 million parameters). In the paper, the authors studied the ability of transformer networks trained in two phases for language understanding. In the first phase, they trained a transformer network to predict the next token given a set of tokens that appeared before (also called unsupervised pre-training, generative pre-training, or in statistics auto-regressive). In the second phase, the transformer networks was fine tuned on tasks with supervised learning (also called discriminative fine-tuning). In summary, their major finding is that combining task agnostic unsupervised learning in the first phase, then using this model in a second phase with supervised learning for fine tuning on tasks can lead to performance gains - from 1.5% on textual entailment to 8.9% on commonsense reasoning.

In 2019, Radford et al. introduced GPT-2 in their paper *"Language Models are Unsupervised Multitask Learners"* [24]. Again, GPT-2 refers to the largest model they have trained. GPT-2 is hence a scaled up version of GPT-1 in model size (1.5 billion parameters), and also in training data size. In particular, GPT-2 has roughly more than ten times the number of parameters than GPT-1, and is trained on roughly more than ten times the amount of training data. They report two major findings. First, the unsupervised GPT-2 can outperform language models that are trained on task specific data set, without these data sets being in the training data set of GPT-2. Second, GPT-2 seems to learn tasks (for example question answering) from unlabeled text data. In both cases, however, the performance did not reach the state-of-the-art. In summary, their major finding is that LLMs can learn tasks without the need to train them on these tasks, given that they have sufficient unlabeled training data.

In 2020, Brown et al. introduced GPT-3 with the paper *"Language Models are Few-Shot Learners"* [5]. Unlike the above two cases, GPT-3 refers to all the models the authors have trained, i.e. it refers to a family of models. The largest model the authors have trained is GPT-3 175B, a model with 175 billion parameters. In their paper, the authors showed that language models like GPT-3 can learn tasks with only a few examples, hence the title includes *"few-shot learners"*. The authors demonstrated this ability by fine-tuning GPT-3 on various tasks, including question answering and language translation, using only a small number of examples.

In 2023, OpenAI introduced GPT-4 [21]. Contrary to previous versions of GPT, this version is a multimodal model as it can process text and images as an input to produce text. This model is a major step forward as it improves on numerous benchmarks; however, it suffers from reliability issues, a limited context window, and inability to learn from experience like previous GPT models. This release, however, diverges from previous GPT models as OpenAI is secretive about *"the architecture (including model size), hardware, training compute, dataset construction, training method, or similar"*. We only know about the model that it is a transformer style-model, pre-trained on predicting the next token on publicly available and not disclosed licensed data, and then fine-tuned with Reinforcement Learning from Human Feedback (RLHF). Notwithstanding this departure, the authors include in their report findings on predicting model scalability. They in particular report on predicting the loss as a function of compute, and the mean log pass rate (a measure on how many code sample pass a unit test) as a function of compute given a training methodology. In both cases, they find that they could predict the respective measure with high accuracy based on data generated with significantly less compute (1.000 to 10.000 less). They also find the inverse scaling price for a task, meaning that the performance on a task first decreases as a function of model size and then increases after a particular model size.

OpenAI introduced a conversational LLM called ChatGPT in 2022, which builds upon their GPT LLM series [19]. As a model, the first version of ChatGPT was based on GPT-3.5 and is an InstructGPT sibling. GPT-3.5 is a GPT-3.0 model trained on a training data set that contains text and software code up to the fourth quarter of 2021 [18]. InstructGPT was introduced in *"Training language models to follow instructions with human feedback "* [22], and is a GPT-3 model fine tuned with supervised learning in the first step, and in the second step with reinforcement learning from human feedback [29]. ChatGPT is hence a GPT-3.5 model fine tuned for conversational interaction with the user. In other words, the user interacts with the model via sequence of text (the conversation) to accomplish a task. For example, we can copy and paste a text into ChatGPT's input field and ask it to summarize it. We can even be more specific, we can say that the summary should be 10 sentences long and be written in a preferred style. Importantly, if we are unhappy with the result we can ask ChatGPT to refine its own summary without copying and pasting the text it should summarize. At the moment of this writing, ChatGPT can be used with GPT-4 as the backend LLM.

There are also other large language models. In 2022, Zhang et al. introduced Open Pre-trained Transformer (OPT) with the paper *"OPT: Open Pre-trained Transformer Language Models"* [34]. The main contribution of that paper is that it makes all artifacts including the nine models available for interested researchers. These models are GPT-3 class models in parameter size and performance. Another open LLM is BigScience Large Open-science Open-access Multilingual Language Model (BLOOM) (176 billion parameters), which was developed in the BigScience Workshop [26].

2.3 Uptake of Large Language Models

Above, we briefly discuss LLMs, where we focus particularly on the GPT model family as these are the most popular LLMs, we hypothesise. It is important to recognize the transition from GPT-3 to GPT-4, as it brought a massive increase on a variety of benchmarks, in particular on academic and professional exams [21]. These performance increases in NLP tasks are a result of natural language understanding and have, as we argue, massive implications for what can be automated – the automation frontier. This frontier is arguably shifted further when natural language understanding is combined with plugin software components. In fact, at the time of writing, the company behind GPT is experimenting with ChatGPT plugins. Among the currently offered plugins are Klarna, Wolfram, the integration with vector data bases for information retrieval, and an embedded code interpreter for Python [20]. This has an impact on Robotics Process Automation (RPA), and more broadly on business process automation including Business Process Management Systems (BPMSs), and more generally on how work is carried out.

3 Large Language Models and the BPM Lifecycle

In this section we identify some of many possible applications of LLM within BPM. We systematically explore these applications along the phases of the BPM lifecycle, namely identification, discovery, analysis, redesign, implementation, and monitoring [8]. In this way, we complement recent efforts to build an overarching inventory of LLM applications, such as in other fields like data mining[1]. As mentioned in the first sentence, we identify some applications, our list of applications is by no means complete. The list serves as a starting point, and needs to be refined as more experience is gained with this technology.

3.1 Identification

The BPM lifecycle starts from *Identification*. Normally, at this stage there is not much structured process knowledge available in the company, and relevant information has to be extracted from heterogeneous internal documentation. This is

[1] See for example the OpenAI Cookbook GitHub repository, which provides code examples for the OpenAI API.

exactly where LLM shine as they can quickly scan and summarize large volumes of text, highlighting important documents or directly outputting required information.

Identifying Processes from Documentation. The idea is to give LLM all relevant documentation existing in the organization as input. This can include legal documents, job descriptions, advertisements, internal knowledge bases and handbooks. The LLM is then tasked to identify which processes are taking place in the organization. It can be further instructed to classify the input documents according to processes they describe. Multimodal LLM can improve the results even further as charts, presentations and photos can also directly be used as information sources.

Process Selection. LLM can be further asked to assess strategic importance of processes based on, e.g. number and types of documents that refer to them as well as extract this information from process descriptions. If given access to information systems supporting the process or other KPIs, LLM can also assess process health. Finally, assessing feasibility is also theoretically achievable as long as necessary information, e.g. recent technology reports, is given as input as well. Based on these criteria, LLM can prioritize the processes for further improvement.

3.2 Discovery

The second stage of BPM lifecycle is *Process Discovery*. At this stage one or a combination of process discovery methods is selected to produce process models. When one speaks of automated process discovery, one usually means process mining – a technique of extracting process models and other relevant data out of event logs left by information systems supporting the execution of a process. However, with LLM also other discovery techniques can benefit from (at least partial) automation.

Process Discovery from Documentation. Apart from process mining, documentation analysis is an established process discovery method. In this method, process analyst uses the information found in heterogeneous sources such as internal documentation, job advertisements, handbooks, etc. Searching in these documents might require a lot of time and effort. LLM are extremely suitable for this task as they can summarize high volumes of text in a concise and structured way. More precisely, they can output process descriptions in desired format (plain text, numbered lists, etc.). One can also specify the level of detail, as to whether the output should include only the activities and events or also resources and additional information. Finally, as some LLM are also capable of working with structured document formats such as XML, in fact even BPMN models can be produced automatically.

Process Discovery from Communication Logs. Another information source that can be used in evidence-based discovery is communicaiton logs, i.e. e-mails and chats between process participants: internal employees but also external partners and customers. LLM can extract patterns from these communication logs, which can be seen as various steps in a process. Then, they can similarly produce process descriptions or models.

Interview Chat Bot. Possible applications of LLM in process discovery can also go beyond evidence-based discovery. Another common discovery method are interviews with domain experts. In these interviews, process analyst asks questions about the process and produces a process model based on several interviews. Typically, several separate interviews with different domain experts are required to produce the first version of process model. Afterwards, additional rounds of interviews are conducted in order to get and incorporate feedback and to perform validation. In the worst case, domain experts might have conflicting perceptions of the process, then resolving such conflicts becomes a very difficult and time-consuming task for both process analyst and domain experts.

LLM can solve parts of this problem by providing a chat bot interface for domain experts. In this way, the domain experts answer questions in the chat. This can bring a lot of advantages. First, the domain experts do not have to allocate lengthy time slots for interviews but instead talk with the chat bot at desired pace. Second, the feedback loop gets shorter as LLM can produce process models directly after or even during the conversation with the domain expert and also do updates to the model, thus validation can happen simultaneously with model creation. Finally, the benefits will only grow if multiple domain experts interact with the chat bot simultaneously (and independently) but the chat bot can use all of this input in the conversations. The latter option is, however, more difficult to implement.

Combined Process Discovery. All process discovery methods have their advantages and drawbacks. Often, a combination of these methods is used to achieve best results. However, this combination is limited by the resources that are allocated for process discovery task. Discovery methods presented above give valuable output yet requiring much less resources. Thus, it is possible to apply more of them simultaneously for even better result. The combination of these methods can be used in addition to traditional process mining or "manual" process discovery, which will provide the richest insights. While it could happen that the results of different methods have some inconsistencies that will have to be fixed, also fixing them can be done in (semi-)automated manner.

Process Model Querying. As LLM seem to "understand" process models serialized as XML, they can be used to answer some questions about the model. This can be very useful for quality assurance. First of all, it can be used for checking syntactic quality. While there are tools out there that can do it already, and with much less overhead, it is still convenient to have this feature in LLM because LLM, in contrast to other methods, may be able to check other quality aspects as well. For instance, it can also check semantic quality. Indeed, process analyst can give LLM both interview transcript and a process model as input and LLM can check both validity and completeness based on this interview. It must be noted, of course, that this will only work under the assumption that the interview transcript has these features of validity and completeness. Another way of checking semantic quality of the model would be via process simulation, e.g. to explicitly ask LLM whether the given process model could have produced a given execution sequence or to ask LLM to give possible execution sequences that can be generated by the model. LLM are known to be able to simulate Linux shell, for instance, thus they might be also able to simutale a BPMS execution engine as long as enough input is provided. Finally, LLM can also (at least so some extent) check pragmatic quality of the models as long as some definition of guidelines, e.g. 7PMG is provided as input as well. It must be also noted that LLM can not only spot these quality issues but also suggest fixes.

3.3 Analysis

The next stage is *Process Analysis*. At this stage, the discovered processes are analyzed to find problems and bottlenecks. While this is a cognitively loaded task, LLM can be used to help human analysts in some regard.

Issue Discovery. If an issue exists in a process, chances are high somebody has already complained about it. Depending on the company, product, and process it can be the customer, partner or an employee and it can happen on different platforms, including social media, support service or internal communication tools. LLM are good at summarizing large volumes of unstructured text as well as finding patterns, and this capability can be used for this task. It is as easy as just scraping the text from these platforms and giving it as input to the LLM with a simple prompt like "find all things customers have complained about".

Issue Spotting. After an issue in the process is found, the next step is to spot the part of the process that creates this issue. In some cases, it can be a difficult task, especially in a complex process. The idea here is to give LLM all process models (or models of the relevant process in case it is known that only one process causes the issue and it is known exactly which process) and the spotted problems. The task of LLM is, by analyzing task names and descriptions to make suggestions which tasks may be responsible for the issue. In advanced cases, LLM might be even capable of suggesting some fixes. It might be something as simple as suggesting to automate some manual task that takes too long but it also might be some more complex process redesign suggestion as long as LLM is given redesign methods as additional input or is trained on redesign methods as well.

3.4 Redesign

The fourth phase of the BPM lifecycle is *Process Redesign*. In this stage, process improvement suggestions are developed based on discovered issues and general process improvement methods. These suggestions are evaluated, and a to-be process model is developed at the end of this stage.

Business Process Improvement. An obvious yet very promising use case is to just ask LLM to redesign the process. As already mentioned, simple issues arising from just one activity can be fixed by the LLM. However, it does not stop there and is theoretically only depending on the quality of the input given to the LLM. Indeed, if it is given exhaustive information about the process (detailed process model as well as description of the process or tasks) as well as detailed description of some redesign method (or it is trained on some redesign methods), redesigning the business process is as simple as just telling the LLM to apply the method on the process. This can, however, be improved even further. First, the description of the issues discovered in the previous phase can be given as additional input to guide process redesign to fix those first. Second, LLM can be instructed to apply different redesign methods and to give separate lists of suggestions given by each of them so the analyst can then select the best options. Moreover, LLM itself can be asked to choose the best suggestions and motivate its choice. It must be noted, of course, that this will only work if sufficient input is given. For instance, for inward-looking redesign methods, the methods themselves as long as detailed process information is required. For outward-looking methods, in addition to that, there should be enough outside information and/or a way for LLM to properly communicate with the outside world.

3.5 Implementation

The next phase of the BPM lifecycle is *Process Implementation*. It covers organizational and technical changes required to change the way of working of process participants as well as IT support for the to-be process.

BPMN Model Explanations with Plain Text. As mentioned, LLM can work with BPMN models serialized in XML. We have already discussed how LLM can manipulate process models in order to increase quality as well as suggest or incorporate redesign ideas. To close the circle, LLM can produce textual explanations of BPMN models. What is more interesting, one can control the level of detail as well. So, depending on the target audience, LLM can produce textual overview but also detailed descriptions of the models. It can transform it into requirements for software developers if enough details are contained in the BPMN model itself.

BPMN Model Chatbot. Building on top of the previous use case, model description can be also tailored to every specific user. This way, given a model or – better – model repository with additional documentation, LLM can prepare specific descriptions for, e.g. process owner but also for individual participants for which all specific tasks they are responsible for are also described and explained in detail. Furthermore, in this use case one can add interaction between the user and LLM. This way, user may ask clarifications for parts he did not understand or generally ask for more details as long as some guidance is required.

Process Orchestrator. LLMs can be accessed via APIs and at the same time can access APIs themselves, opening a huge variety of opportunities. While the former means it can be used for automated tasks and be called by the orchestrator, the latter means that it could theoretically be an orchestrator itself: given executable process model and additional constraints as context as well as the required instance data as input, it can theoretically execute a process by calling other APIs and assigning tasks in a more flexible way than a traditional orchestrator.

3.6 Monitoring

The last phase of the BPM lifecycle is *Process Monitoring*. At this stage, already implemented processes are executed, and their performance is monitored. The observations collected in this phase are used for operational management as well as serve as input for further iterations of the lifecycle.

Process Dashboard Chatbot. Dashboards are a powerful tool that provides overview of the most important KPIs of a process on a single screen. However, the ultimate goal of them is to tell the viewer whether the status of the process is good or not, and the numbers and colors are mostly used as an intermediary medium. LLM can take away this intermediate step and allow the user to directly know the status of the processes.

4 Research Directions

In this section we propose the research directions. We categorize the research directions into three groups. The first group studies the use of LLM, and their applications, in practice. This includes the use within BPM projects in companies or as part of an Information System (IS) (Sect. 4.1), the development of usage guidelines for practitioners and researchers (Sect. 4.2), and also the derivation of BPM tasks (Sect. 4.3) and their corresponding data sets (Sect. 4.4). The second group studies how LLM an be combined with existing BPM tools, and more generally BPM technologies, to increase user experience (Sect. 4.5). Crucially, this group draws from findings in the first group. The third and final group develops large language models specifically for business process management, so these models can understand the context and language of business processes and support various tasks, such as process discovery, monitoring, analysis, and optimization (Sect. 4.6). Again, this group builds upon the findings of the first group.

4.1 The Use of Large Language Models in BPM Practice

The first research direction studies the use of LLM in practice. One major question to answer is for which tasks LLM can be used. In Sect. 3, we present a list of tasks for which LLM can be used. However, this list might not be complete, in addition some of the tasks might turn out to be of little use. Tied to this is the question what tasks will bring, and ultimately bring the most value for an organization. The next big question is the relation between a task and the model properties needed to achieve a pre-defined value. One question here is which tasks can be achieved with already existing models. Another question to study is whether we always need the largest, and hence most accurate model, for each task. We hypothesize that this might not be the case. Finally, and most importantly, the next big question to answer is how LLM will change how work is carried out within BPM projects, and within processes that are actively managed. We for example hypothesize that conversational LLMs might take the spot of the duck in the famous *duck approach*[2]. This question is a socio-technical systems question, and we hence strongly believe that the BPM community, and the information systems community more broadly, is especially well equipped to contribute to this question.

4.2 Usage Guidelines for Researchers and Practitioners

The second research direction builds usage guidelines for BPM researchers and practitioners. One question such guidelines have to answer is given an organizational context, the lifecycle phase, and the process context of a task, suggest a LLM to achieve an expected value. In addition, such guidelines systematically collect best practices for creating prompts. For example, for the BPM lifecycle phase process implementation, and monitoring and controlling, a company might consider using a LLM within a managed process. Let us assume this company is a bank and wants to automate the task of replying to customer inquiries with LLM. Then this guideline proposes for the process implementation a specific LLM, with the number of parameters it has, gives examples on how to create a prompt template, fill the template with customer background information, and finally on how to integrate the customer inquiry within the prompt template. For process monitoring and controlling, the guidelines might propose a different model for analyzing different inquiry clusters as the lifecycle phase context is different. As an example, consider here that the LLM first categorizes each inquiry into a positive and negative sentiment, and then lists for both the top five inquiry reasons. This research direction builds upon the first research direction, as first research direction, among others, determines the tasks for which LLM can be used in principle.

4.3 Creation, Release, and Maintenance of Task Variants Specific to BPM

This research direction builds and maintains two different task lists. The first list maps general NLP tasks to tasks within BPM. As an example, consider the general NLP task of text summarizing. Within BPM, text summarizing can relate to summarizing a set of process descriptions or task descriptions. We can

[2] Rubber duck debugging.

think of this list as a one to many mapping between NLP tasks on the one hand, and BPM tasks on the other. The second list enumerates tasks that are unique to BPM. This research direction uses the findings from the directions presented in Sect. 4.1 and Sect. 4.2.

4.4 Creation, Release, and Maintenance of Data Sets and Benchmarks

Public data sets and benchmarks are crucial for the progress of LLM in research as they allow researchers to measure progress. In addition, they are also important for practitioners as they define data set properties (such as meta-information) they are likely to need themselves when they fine tune a model. As a result, data sets and benchmarks need to be properly aligned with the automation needs of BPM. Blagec et al. argue similarly as we, but for the clinical profession [3]. In their study, they analyzed 450 NLP data sets and found that *"AI benchmarks of direct clinical relevance are scarce and fail to cover most work activities that clinicians want to see addressed"*. A research direction for the BPM community is hence to do the same for BPM. One question worth studying is whether existing NLP data sets and benchmarks are of relevance to BPM, for example, if they cover the activities of BPM researchers and practitioners. This research direction builds upon the research direction in Sect. 4.3.

4.5 LLM and BPM Artifacts

This research direction studies the interplay of LLM, BPM artifacts, and BPM tasks. The goal is to understand which artifacts are necessary for LLM, and their multimodal successors, to create useful outputs. It can hence be understood as a special case of prompt engineering, which we might call multimodal prompt engineering for BPM. This is an important research direction as the output quality of a LLM depends heavily on the context quality and quantity it is given. In other words, the more context, and the higher the quality of each context, the higher the output quality of the LLM. For this reason, we believe that it should be considered its own research direction. As an example, consider again the customer inquiry process from above. In this case, we can imagine that the context of the LLM depends on the inquiry. In one case, the customer might include an image in the inquiry. Or think of the redesign phase of the inquiry process. During this phase, artifacts are created, for example drawings of processes on a board, comments to these processes in a word processor, and remarks on data availability and access in an audio file. This information might be useful when we ask – a possibly different – LLM why a customer inquiry on current special offers cannot yet be answered. The reason here might be that a central system which stores special offers does not yet exist. This research direction builds upon the directions presented in Sect. 4.1 and Sect. 4.2.

4.6 Development and Release of Large Language Modelss for Business Process Management

This research directions studies how LLM are build for BPM tasks, all previously discussed research direction are the foundation for this direction. The goal of the research direction is to build LLM that are attuned to the specific challenges and requirements of BPM, compared to general-purpose language models. This includes specialized models in the sense of exclusive for, and also general-purpose language models that are fine-tuned on the BPM domain. An important aspect of this direction is to open source the created LLM, as is done for OPT [34]. This is important for researchers can use this model in their studies, and practice as companies can use these models free of charge for their use cases.

5 Discussion

In this section we discuss the challenges of LLM, the power of combination and inflated expectations, and end with an outlook and future work.

Challenges and Risks. The use of LLM entails opportunities and challenges. For example, they can help to understand difficult research, but they also carry over deficiencies (including factual errors) in the training data set to the texts they generate [32]. In a systematic study of these errors, Borji analyzes errors of ChatGPT and categorizes them – the author further outlines and discusses the risks, limitations and societal implication of such models[3] [4]. The failure categories identified by the author include reasoning, factual, math, and coding. A similar deficiencies study was done in [2], but these authors focus on LLM in general. A news feature in Nature discusses these and the risks of using LLM [9]. One consequence for education might be that essays as an assignment should be re-considered [30].

Besides these LLM specific challenges, there are also more general challenges, some may call these general challenges risks. Two of these risks are data privacy and security concerns. If for example a company uses a LLM from a third party which also hosts it, then one can never be certain that the input to the LLM, text, images, or tabular data which includes financial company data or specifics about the IT infrastructure for example, is never shown to a third party. The input might be seen by multiple parties in such a case, if we consider a data pipeline with quality control than two parties might just be the data annotators – one who does the first labelling and the second who does quality control.

The Power of Combination and Managing Expectations. The major innovation of ChatGPT was not the introduction of a new technology, but the combination of already existing ones and an easy to use user-interface [12]. This effect of combination extends beyond LLM, NLP, or ML innovations. For example, OpenAI is currently experimenting with integrating ChatGPT with software plugins, which

[3] See the ChatGPT failure archive (GitHub) for an up-to-date list.

might even in the short run lead to a software marketplace for their platform[4]. For this reason, we suggest and advocate in our research directions above to study and build these combinations with *existing* BPM technologies, instead of solely focusing on developing new ones. In this paper, we have so far made the case for the opportunities LLM realize, shortly discussed their shortcomings, and pointed out how important it is to combine technologies within a field, and across field boundaries. However, we also stress here how important it is to manage, maybe even overshooting, expectations driven by this very recent developments. For example, the speculation about the possible capabilities on the successor of GPT-3 were driven up by the hype to a point where *"people are begging to be dissapointed"* [12].

Outlook and Future Work. LLM are used, and will be used in commercial products with huge amounts of users. We speculate that this will have an effect on research, as funding agencies might increase the amount of grants for this research field. An ever increasing user base that interacts with LLM (directly or indirectly) is therefore, in our view, inevitable. For future work, we plan to work on developing research directions that are beyond the scope of this paper. We expect that LLM will have an effect on how work is carried out (see Sect. 2.3 and Sect. 4.1). But this may have far greater impacts than what we cover here, for example on the BPM capabilities, which are strategy, governance, information technology, people, and culture [25].

6 Conclusion

LLMs have made significant progress in a wide array of tasks, and importantly also progress on ease of use, which makes them promising a universal tool. In this paper, we use exploratory research method to study possible applications of LLMs in BPM. We use the BPM lifecycle to propose applications of LLMs to showcase the impact these models might have for practical use. We also present six research directions for studying and building LLMs within BPM. Besides these opportunities, we also highlight challenges, and more specifically risks, in adopting LLMs. Importantly, we include in our discussion the importance of combination and managing expectations.

References

1. Van der Aa, H., Carmona Vargas, J., Leopold, H., Mendling, J., Padró, L.: Challenges and opportunities of applying natural language processing in business process management. In: COLING 2018: The 27th International Conference on Computational Linguistics: Proceedings of the Conference: August 20–26, 2018 Santa Fe, New Mexico, USA. pp. 2791–2801. Association for Computational Linguistics (2018)

[4] https://openai.com/blog/chatgpt-plugins.

2. Bender, E.M., Gebru, T., McMillan-Major, A., Shmitchell, S.: On the dangers of stochastic parrots: can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. FAccT 2021, New York, NY, USA, pp. 610–623. Association for Computing Machinery (2021). https://doi.org/10.1145/3442188.3445922
3. Blagec, K., Kraiger, J., Frühwirt, W., Samwald, M.: Benchmark datasets driving artificial intelligence development fail to capture the needs of medical professionals. J. Biomed. Inform. **37**, 104274 (2022)
4. Borji, A.: A categorical archive of ChatGPT failures (2023). https://doi.org/10.48550/ARXIV.2302.03494, https://arxiv.org/abs/2302.03494
5. Brown, T., et al.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020). https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
6. van Dis, E.A.M., Bollen, J., Zuidema, W., van Rooij, R., Bockting, C.L.: ChatGPT: five priorities for research. Nature **614**(7947), 224–226 (2023)
7. Dumas, M., et al.: AI-augmented business process management systems: a research manifesto. ACM Trans. Manag. Inf. Syst. **14**(1), 1–19 (2023)
8. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, vol. 2. Springer, Cham (2018). https://doi.org/10.1007/978-3-662-56509-4
9. Hutson, M.: Robo-writers: the rise and risks of language-generating AI. Nature **591**(7848), 22–25 (2021)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
11. 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. ACM Comput. Surv. **55**(9), 1–35 (2023)
12. Loizos, C.: StrictlyVC in conversation with Sam Altman, part two (OpenAI) (2023). https://www.youtube.com/watch?v=ebjkD1Om4uw, YouTube channel of Connie Loizos
13. Malinova, M., Mendling, J.: Identifying do's and don'ts using the integrated business process management framework. Bus. Process. Manag. J. **24**, 882–899 (2018)
14. 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, pp. 86–100. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-23515-3_7
15. Mjolsness, E., DeCoste, D.: Machine learning for science: state of the art and future prospects. Science **293**(5537), 2051–2055 (2001)
16. 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**, 801–827 (2022)
17. Nolle, T., Seeliger, A., Thoma, N., Mühlhäuser, M.: DeepAlign: alignment-based process anomaly correction using recurrent neural networks. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) CAiSE 2020. LNCS, vol. 12127, pp. 319–333. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49435-3_20
18. OpenAI: Model index for researchers. https://platform.openai.com/docs/model-index-for-researchers/model-index-for-researchers
19. OpenAI: Chatgpt: optimizing language models for dialogue (2022). https://openai.com/blog/chatgpt/

20. OpenAI: ChatGPT plugins (2023). https://openai.com/blog/chatgpt-plugins
21. OpenAI: GPT-4 technical report (2023). https://cdn.openai.com/papers/gpt-4.pdf
22. Ouyang, L., et al.: Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155 (2022)
23. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018). https://openai.com/blog/language-unsupervised/
24. Radford, A., et al.: Language models are unsupervised multitask learners (2019). https://openai.com/blog/better-language-models/
25. Rosemann, M., vom Brocke, J.: The six core elements of business process management. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1. IHIS, pp. 105–122. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-642-45100-3_5
26. Scao, T.L., et al.: Bloom: a 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100 (2022)
27. Schäfer, B., Van der Aa, H., Leopold, H., Stuckenschmidt, H.: Sketch2process: end-to-end BPMN sketch recognition based on neural networks. IEEE Trans. Software Eng. **49**, 2621–2641 (2022)
28. Sommers, D., Menkovski, V., Fahland, D.: Supervised learning of process discovery techniques using graph neural networks. Inf. Syst. **115**, 102209 (2023)
29. Stiennon, N., et al.: Learning to summarize with human feedback. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 3008–3021. Curran Associates, Inc. (2020). https://proceedings.neurips.cc/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf
30. Stokel-Walker, C.: AI bot ChatGPT writes smart essays-should academics worry? Nature (2022)
31. 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. Bus. Inf. Syst. Eng. **65**, 95–101 (2023)
32. Van Noorden, R.: How language-generation AIs could transform science. Nature **605**(7908), 21 (2022)
33. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
34. Zhang, S., et al.: OPT: open pre-trained transformer language models. arXiv preprint: https://arxiv.org/abs/2205.01068 (2022)

Engineering

Predicting Unseen Process Behavior Based on Context Information from Compliance Constraints

Qian Chen[1(✉)], Karolin Winter[2], and Stefanie Rinderle-Ma[1]

[1] Technical University of Munich, TUM School of Computation, Information and Technology, Garching, Germany
{qian.chen,stefanie.rinderle-ma}@tum.de

[2] Eindhoven University of Technology, Department of Industrial Engineering and Innovation Sciences, Eindhoven, The Netherlands
k.m.winter@tue.nl

Abstract. Predictive process monitoring (PPM) offers multiple benefits for enterprises, e.g., the early planning of resources. The success of PPM-based actions depends on the prediction quality and the explainability of the prediction results. Both, prediction quality and explainability, can be influenced by unseen behavior, i.e., events that have not been observed in the training data so far. Unseen behavior can be caused by, for example, concept drift. Existing approaches are concerned with strategies on how to update the prediction model if unseen behavior occurs. What has not been investigated so far, is the question how unseen behavior itself can be predicted, comparable to approaches from machine learning such as zero-shot learning. Zero-shot learning predicts new classes in case of unavailable training data by exploiting context information. This work follows this idea and proposes an approach to predict unseen process behavior, i.e., unseen event labels, based on process event streams by exploiting compliance constraints as context information. This is reasonable as compliance constraints change frequently and are often the cause for concept drift. The approach employs state transition systems as prediction models in order to explain the effects of predicting unseen behavior. The approach also provides update strategies as the event stream evolves. All algorithms are prototypically implemented and tested on an artificial as well as real-world data set.

Keywords: Predictive Process Monitoring · Unseen Behavior · Context Information · Compliance Constraints

1 Introduction

Predictive Process Monitoring (PPM) aims at predicting relevant target values based on an event log such as the next event label to occur [16] as well as the remaining time [24] or the outcome [23] of process instances. In general, PPM provides great prospects for decision support in almost any application domain.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 127–144, 2023.
https://doi.org/10.1007/978-3-031-41623-1_8

Especially in advanced application domains such as manufacturing, logistics, and healthcare, PPM is confronted with the problem of *unseen behavior*, i.e., behavior that has not been observed in the event log so far and hence is not available for the training phase of PPM. Unseen behavior might be caused by concept drift as well as by incomplete or infrequent process executions [13,14,20]. Unseen behavior in PPM results in two challenges: i) when and how to update the prediction model at the presence of unseen behavior [4,14,18,20], and ii) how to predict unseen behavior. ii) can be compared to zero-shot learning in Machine Learning (cf. [27]) where classes can be predicted also in case of unavailable training data by exploiting context information.

In this work, we follow up on the idea of zero-shot learning for unseen behavior in PPM, i.e., we investigate how context data can be exploited to enable the prediction of unseen behavior, i.e., unseen next event labels. The exploitation of context information for improving the prediction quality and the explainability of the prediction results has been investigated for different sources of context data such as text [25], expert feedback [5], and sensor data [21]. However, none of the existing approaches has exploited context data for predicting unseen behavior. The idea of exploiting context data is somehow similar to the idea of using a-priori knowledge to be *"leveraged for improving the predictive power of a predictive process monitoring technique"* [8]. The difference is that a-priori knowledge can be considered context data, but not necessarily vice versa. More precisely, context data is not necessarily available in an a-priori way, but emerges and changes during runtime.

In this work, we opt for exploiting one source of context data as a starting point, i.e., compliance constraints. The reason is that compliance constraints are prevalent in almost any application domain and are often themselves the cause for concept drift which, in turn, constitutes a source for unseen behavior. Compliance constraints usually emerge from regulatory documents and can be subject to frequent changes [10], i.e., have to be constantly monitored themselves. Consider a process for the transportation of delicate foods which needs to obey to multiple regulations. If a change in one of those regulations happens, new compliance constraints can come into effect, e.g., *If the temperature in the container exceeds 10° during transportation, these goods must be destroyed.* Since this is a newly imposed constraint, the associated event(s) corresponding to the destruction of goods have not been observed in the underlying event log and are hence not included in the training phase, causing *unseen behavior* in the test phase. The idea is to integrate information from the compliance constraint into the prediction model. This way we do not have to wait until we observe the new process behaviour caused by this constraint, e.g., the event(s) reflecting an activity related to goods destruction. Compliance constraints can therefore be seen as "promising" context to enable the prediction of unseen behavior, resulting in the overarching research question of this work:

How to exploit context information on compliance constraints to predict unseen process behavior, i.e., unseen next event labels?

The input of the approach comprises a process event log and a set of compliance constraints that apply to the underlying process. Furthermore, we consider that the process event log can be incomplete in terms of observations, i.e., not every event label that is conceivable based on the set of constraints has been observed so far. This corresponds to epistemic uncertainty as typically referenced in the machine learning community, cf., e.g., [11,26]. By incorporating knowledge on compliance constraints we do not have to wait until we observe the behavior enforced by them and update the prediction model when we have observed it. We can already anticipate unseen behavior based on, e.g., data attributes, e.g., in the food transportation example the temperature exceeding 10 degrees. After the unseen behavior has been observed our approach uses appropriate update strategies to further enhance the prediction quality.

The presented approach comprises an offline component that augments a state transition system with contextual information from compliance constraints. The augmented transition system is then used for prediction with an event stream in the online component. The approach is evaluated based on synthetic and real-world event logs without and with existing update strategies. Moreover, the augmented transition system is compared to deep learning methods with either a single $LSTM$ layer [14] or a Process Transformer [3].

The remainder of this paper is structured as follows. Section 2 outlines the problem statement, Sect. 3 presents the next event label prediction approach using constraint context information. The approach is evaluated in Sect. 4, related work is discussed in Sect. 5 before the paper concludes in Sect. 6.

2 Problem Statement and Preliminaries

Next event label prediction takes an event log (training data) and an event stream (test data) as input and predicts based on the event log the next event label for the ongoing process instance observed in the event stream. The event log contains events which origin from different cases, each case represented by a trace. An event can have multiple attributes, e.g., an event label, a timestamp, a life cycle transition, data values, or resources that were involved during the execution. In the following running example, we abstract from this representation by just depicting a trace as sequence of event labels and their number of occurrences.

Example: *Assume a scenario with event log* $L = [< A, B, E >^{100}, < A, C >^{100}]$ *as training data where the number in the superscript denotes the frequency of the trace occurrence, and event stream* S *as test data, i.e.,* $S =< A_1, A_2, B_1, C_2 >$ *where the subscript reflects the case id. For S, existing prediction models would result in predicting* E *for case 1 and no prediction for case 2.*

In this paper we aim to predict the next event label for an evolving event stream based on an event log and a set of compliance constraints as additional context information. A compliance constraint c is defined as a triplet (p, s, r) consisting of a non-empty predecessor event p, a possibly empty successor event s and r specifying the relation between p and s. This definition is deliberately kept independent from any formalism such as linear temporal logic, in order to show the general applicability of the approach.

Example (ctd.): *Assume that in the scenario described above, the following two constraints are imposed on the process execution due to, e.g., newly arising or updated regulations:*

c_1 : "D directly follows C"
c_2 : "Y eventually follows B"

Constraints c_1 *and* c_2 *can be formalized as* $c_1 = (\{C\}, \{D\}, \{directly\ follows\})$ *and* $c_2 = (\{B\}, \{Y\}, \{eventually\ follows\})$. *The directly follows relation means that whenever C occurs, D must occur next without other events in between. The eventually follows relation implies that whenever B occurs, Y must occur afterwards* [12].

For the running example, existing next event label prediction without considering the constraint information, cannot predict events D and Y since they have not been observed in L. Only approaches considering updates can incorporate this information if at some time it is observed in the stream. By integrating compliance constraints c_1 and c_2 into the prediction model, we gain knowledge about those additional events in advance allowing for their prediction.

Example (ctd.): *Including the additional knowledge contributed by constraints* c_1 *and* c_2 *into next event label prediction, we envision the prediction of* E *or* Y *for case 1 and the prediction of* D *for case 2.*

The presented approach will be capable of predicting unseen events that stem from constraints without requiring updating the prediction model. However, as soon as unseen behavior emerging from sources other than constraints is observed in the stream, this information is included via updates, as well.

Example (ctd.): *Assume that event stream S evolves in the following steps. Then existing approaches and the envisioned approach including constraints yield the prediction results summarized in Table 1 without updating the prediction models. We can see that existing approaches do not yield any predictions if the trace length exceeds the longest observed trace which is the case for any of the streams in Table 1. This might give the prediction with constraint information an edge, at least until prediction models are updated. We will investigate the "sweet spot" between the effort and gain of including constraint information into the prediction, in connection with update strategies, in Sect. 4.*

Table 1. Prediction results with evolving event stream; –: no prediction

Event stream	Existing approaches	Including constraints
$S =< A_1, A_2, B_1, C_2, E_1 >$	case 1: – ; case 2: –	case 1: Y; case 2: D
$S =< A_1, A_2, B_1, C_2, E_1, D_2 >$	case 1: – ; case 2: –	case 1: Y; case 2: –
$S =< A_1, A_2, B_1, C_2, E_1, D_2, Y_1 >$	case 1: – ; case 2: –	case 1: – ; case 2: –

In order to consider that unseen behavior is predicted, we aim at predicting not only the next event labels, but also how certain we are for the upcoming event label.

In the paper, we assume that we are in a violation free setting, meaning that the event log L never violated the set of compliance constraints C we consider with unseen behavior, but violations of other compliance constraints are in principle possible. In the case of the event stream, we only include cases that do not contain violations of compliance constraints when updating the prediction model. Moreover, we consider that we have observed the predecessor of a constraint. That in turn means that a chaining of constraints, i.e., "B follows A" and "C follows B", may not occur as we would lack the predecessor event B. Furthermore, the assumption is made that there are no overlaps in control-flow constraints, i.e., "B directly/eventually follows A" and at the same time "X directly/eventually follows A" cannot occur without further knowledge. Further knowledge means in this case that certain conditions on data attributes need to hold, e.g., "If the credit amount is greater than 10.000€, "perform detailed check" must happen after "request received", otherwise "perform normal check" must happen after "request received". In this work, we only consider control-flow constraints without any further information (e.g., data attributes), but it will be taken into account in future work.

3 Next Event Label Prediction Approach

In order to address the problem as outlined in Sect. 2, we follow the basic idea of predictive process monitoring approaches by initially training a prediction model in an offline component and carrying out the prediction and update of the prediction model in an online component. For the offline component the input consists of an event log L and a set of compliance constraints C. The output is a prediction model trained based on information from L combined with the external knowledge based on C. The online component takes an event stream S and the set of compliance constraints C as input and delivers the prediction of the next event label with corresponding probability. In order to foster explainability the user is informed whether the prediction was made solely based on L or based on which compliance constraint in C. Moreover, several update strategies allowing to cope with concept drifts induced by changes in the underlying process models are integrated into the online component.

As prediction model we opt for transition systems as introduced in [1,19]. Both papers focus on remaining time prediction and the latter incorporates regression models based on data attributes and allows for activity sequence prediction as well. Transition systems are selected because we are facing the challenges of i) frequent changes causing unseen behavior and ii) explainability. Considering i) compared to, e.g., deep-learning models, transition systems with appropriate abstractions can be constructed at low computational costs allowing for incorporating changes as soon as they occur without waiting hours or days for retraining the prediction model. For ii) transition systems are a white box model and allow for different explainability options, i.e., we can easily convey prediction results to users and, e.g., distinguish between predictions that are made solely based on the given event log or predictions that were made based on a particular constraint in combination with the probabilities attached to the prediction result.

3.1 Creating the Prediction Model – Offline Component

For the offline component, at first, an annotated transition system based on event log L is constructed analogously to [1,19] and later on augmented with constraint information. A *transition system* TS constructed based on event log L consists of a set of states S, a set of event labels E and a set of transitions T. A transition t is a triplet (s_1, e, s_2) determining how one state s_1 is conveyed into another state s_2 via an event e. Events and states can be represented through different representation functions. An example for an event representation function is the function that maps an event onto its event label. An example for a state representation function is the function that maps a partial trace onto its sequence of event labels.

Figure 1 depicts the transition system TS for the running example as introduced in Sect. 2 where the parts of TS generated based on the event log L are depicted in black. The partial traces created from the traces in L are $< A >, < A, B >, < A, B, E >, < A, C >$. As illustrated in Fig. 1 in the case of i) representing states as sequences of partial traces, i.e., mapping each partial trace onto the event labels while taking the order into account, the set of states S consists of exactly those traces. In the case of ii) representation as last event, i.e., mapping each partial trace onto the label of the last event, the states are given as A, B, C and E. Note that artificial empty start states $<>$ and $()$.

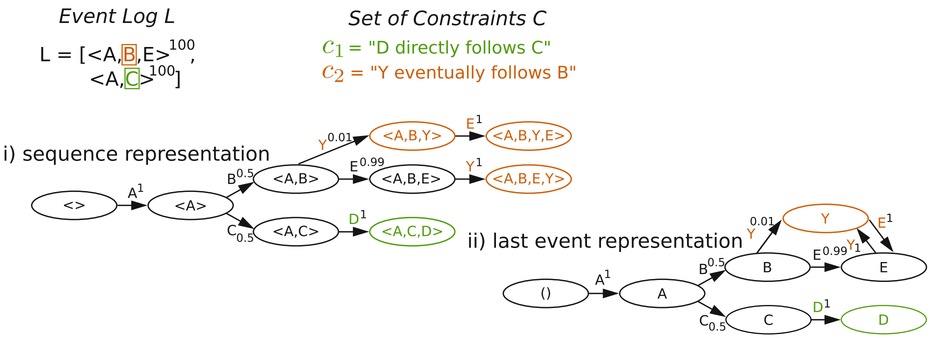


Fig. 1. Augmented Annotated Transition Systems for the Running Example

States can be further annotated with all possible next states, and each of them is associated with a probability from the prior state to them. The probability of each outgoing state is calculated using a measurement function. An *annotated* *transition system* ATS is then a transition system TS together with an event and state representation as well as a measurement function [1]. As measurement function for the TS constructed based on event log L we consider the function calculating the probability based on relative occurrences of the transitions within the log file. For instance, the probability from state $< A >$ to state $< A, B >$ in the running example in Fig. 1 is the quotient of the number of visits from $< A >$ to $< A, B >$ and the number of total visits from state $< A >$ to its all possible next states. So in this case, the probability from state $< A >$ to state $< A, B >$ is 0.5, as $< A, B >$ has been observed 100 times in the event log and $< A, C >$ occurs in 100 cases.

For further details on transition systems, state and event representations, as well as measurement functions, we refer the reader to [1, 19].

In order to incorporate constraint information, the basic annotated transition system ATS calculated based on event log L is augmented based on information from the set of compliance constraints C. The result is denoted as *Augmented Annotated Transition System AATS* and is constructed based on Algorithm 1. The algorithm works independently from the chosen state and event representations.

Algorithm 1. Annotated Transition System Augmentation Algorithm

Input: ATS = annotated transition system based on an event log, C = set of constraints, one constraint c corresponds to a triple (predecessor, successor, relation)

Output: $AATS$ = augmented annotated transition system

1: **for** each eventually follows constraint c in C **do**
2:     **if** c.predecessor and c.successor have been observed in ATS **then** c was already observed, do no augmentation for constraint
3:     **end if**
4:     **if** only c.predecessor has been observed in ATS **then** add c.successor to every state which contains c.predecessor; extend beyond constraints if the state with further events
5:     **end if**
6:     **if** c.predecessor has not been observed but c.successor has been observed in ATS **then** there must be an error or violation in the log for constraint
7:     **end if**
8: **end for**
9: **for** each directly follows constraint c in C **do**
10:     **if** c.predecessor and c.successor have been observed in ATS **then** c was already observed, do no augmentation for constraint
11:     **end if**
12:     **if** only c.predecessor has been observed in ATS **then** add c.successor to every state which ends with c.predecessor; extend beyond constraints if the state with further events
13:     **end if**
14:     **if** c.predecessor has not been observed but c.successor has been observed in ATS **then** there must be an error or violation in the log for constraint
15:     **end if**
16: **end for**

In Algorithm 1, we take as input the basic annotated transition system ATS and a set of compliance constraints C. In order to avoid violation after augmentation, the basic ATS is augmented with eventually follows constraints first (see line 1). For example, if an additional constraint c_3 : "F directly follows E" is introduced for the running example in Sect. 2, then $< A, B, E >$ is extended with F. However, the eventually follows constraint c_2 : "Y eventually follows B" will generate a violated trace $< A, B, E, Y, F >$ on top of $< A, B, E, F >$. Next, the basic ATS is augmented with the eventually follows constraint based on three scenarios. For scenario 1 (line 2 and line 3), if both events are observed in ATS then no augmentation is conducted. The main scenario this paper focuses on is described from line 4 to line 5, in which the predecessor has been observed in ATS while the successor is unseen at the moment. For eventually follows augmentation, the successor should be added to the end of each state in which the predecessor is included. We represent this augmented state as a constraint state as it is directly constructed from a constraint. When an additional state is added to the basic transition system, the count of the corresponding transition from the initial state to the additional state is increased by 1. If the initial state which contains the predecessor of the constraint has further events, then those events need to be added after the constraint state. Line 6 and line 7 indicate a violation scenario if the successor is observed before the predecessor in the log. The augmentation process for directly follows constraints is similar to the eventually follows one except for the second scenario from line 12 to line 13. In this case, only states which end up with the predecessor of the constraint are augmented with the successor of the constraint. We again extend the constraint state with following events if the initial state has further events.

After constraint augmentation as described in Algorithm 1, additional states and transitions are appended on top of the basic ATS. This leads to an update of annotations constructed in ATS before. Here we use the same measurement function as for ATS to calculate the probability of relative occurrences for states from the basic transition system and additional states from the augmented transition system. For example, the probability from state $< A, C >$ to state $< A, C, D >$ in the running example as depicted in Fig. 1 is 1 since there is only one transition from $< A, C >$ to $< A, C, D >$ which is augmented from the constraint.

In Fig. 1, the transitions and states that are inserted based on constraint information are depicted in green and orange depending on which constraint was used to create the state.

3.2 Next Event Label Prediction – Online Component

The online component serves two purposes, i.e., predicting the next event label and constantly updating and improving the existing prediction model based on event stream S. In the phase of online prediction, the $AATS$ constructed from the offline phase is applied to incoming traces from the event stream to predict the next event labels with their associated probabilities. The next event label with the highest probability is selected from the set of possible events. Note here, if the next event label is predicted based on information from compliance constraints, then the corresponding constraint information is provided, as well. This enables our approach to explain whether the prediction is based on the event log or the set of constraints. There could also be multiple next event labels possible for a partial trace with the same probability. In this case, the $AATS$ will predict all possible next event labels with again the probabilities and constraints if available.

As we use occurrence frequencies as measurement function for calculating probabilities from one state to another, consequently, updating the prediction model constantly as event stream evolves is needed. Here, the updating mechanism can be either triggered based on i) the event stream S or by ii) changes in the constraint set C. For i), the following scenarios are possible: a) only labels that were already observed in the log and constraint set are in the stream, i.e., no new/additional states and transitions need to be created and only the probabilities need to be recalculated; b) new labels are observed resulting in the need for a full retraining of the model, i.e., both ATS creation and constraint augmentation need to be conducted again. That means the initial ATS is updated based on new states and transitions, then the constraint augmentation according to Algorithm 1 is applied to the updated ATS. For ii) we need to recalculate only the augmentation part based on the initial ATS that is constructed from the event log L, i.e., perform Algorithm 1 with the newly updated set of compliance constraints. Combinations of i) and ii) are also conceivable.

4 Evaluation

The approach is prototypically implemented and available at https://www.cs.cit.tum.de/bpm/software/. To evaluate the approach a synthetic event log[1] generated using the Cloud Process Execution Engine[2] (CPEE) [15] and a well-established real-life event log, the Helpdesk event log[3], are used. As to the best of our knowledge, this approach is the first to aim at predicting unseen behavior resulting from constraints, we consider the following four comparisons, i) ATS vs. AATS without updates, ii) ATS vs. AATS with updates, iii) AATS vs. a deep-learning model without updates, and iv) AATS vs. a deep-learning model with updates. The first comparison aims at illustrating the advantage of taking constraint information into account. The second comparison shall demonstrate that the updating strategy for the AATS was chosen correctly, and the third comparison shall provide insights on how well a basic technique like the AATS performs against sophisticated prediction models. The last comparison seeks to highlight the benefits of our proposed approach against existing approaches in dealing with unseen behavior, i.e., updating prediction models when considering unseen event labels or new sequences. Given that the prediction of the next event label is a multi-class classification problem, we choose *accuracy*, *precision*, *recall*, and the *f1-score* as metrics. The *accuracy* measures the proportion of correct predictions to the total number of predictions made. The *f1-score* is the harmonic mean of *precision* (or positive predictive value) and *recall* (or sensitivity), where *precision* determines the exactness of the model, and *recall* measures the model's completeness [3].

1 https://www.cs.cit.tum.de/bpm/data/.
2 https://cpee.org, last access: 2023-03-21.
3 https://data.4tu.nl/articles/_/12675977/1.

4.1 Data Sets

All data sets consist of an event log L containing only the predecessor events of the compliance constraints, i.e., the successor events are still unseen, the set of compliance constraints C and a test set representing the event stream S. The latter contains full information, i.e., also the unseen event labels for the constraints in set C.

Synthetic Data. In order to meet our violation-free assumption (cf. Sect. 2), we generate a process model with decision and parallel gateways as shown in Fig. 2. The training set contains 200 cases among which 4 different activities A, B, C, D are executed. Then one directly follows constraint: $c_1 = (\{D\}, \{E\}, \{directly\ follows\})$ and two eventually follows constraints: $c_2 = (\{B\}, \{X\}, \{eventually\ follows\})$, $c_3 = (\{C\}, \{Y\}, \{eventually\ follows\})$ are imposed on the process model. This is reasonable since constraints might change over time and these changes should be integrated into the process model as soon as possible. As our paper aims to address partially seen cases, thus, all predecessors of constraints (i.e., B, C, D) are already observed in the training set, while the successors of them (denoted in green and orange boxes in Fig. 2) (i.e., E, X, Y) remain unseen in the historic event log. However, those unseen activities from the set of constraints could occur in stream data as time goes by. Therefore, the test set with 200 cases is generated based on the whole process model with unseen activities E, X, Y additionally.

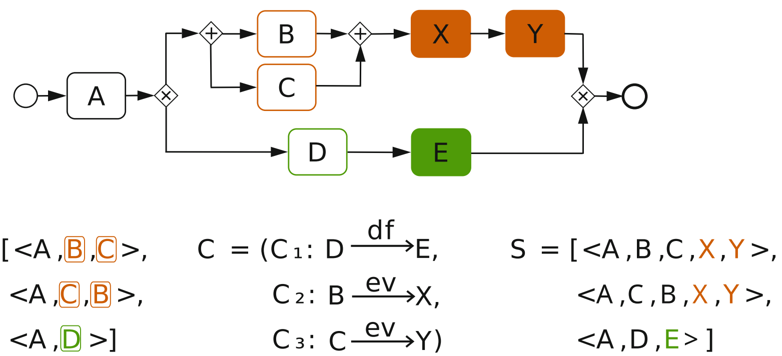


Fig. 2. Process model for synthetic data set

Helpdesk Dataset. The Helpdesk event log reflects a process to resolve tickets that are raised by users. The data set contains 21348 events with 14 distinct activities distributed among 4580 cases. The average case length of all process instances is 4.66, while the maximal value is up to 15. In addition to the event log, a set of constraints is needed. As this data set does not include any information about compliance constraints, we artificially create a directly follows constraint

$c_1 = (\{Resolve\ ticket\}, \{Closed\}, \{directly\ follows\})$ based on the analysis of the log file. To align with our assumption, all events with event label "Closed" are removed from the log file to keep the successor of the constraint unseen. The training set consists of all events without "Closed", while the test set is the same as the original log file with "Closed".

4.2 ATS vs. AATS Without Updates

To demonstrate the benefits of leveraging additional information from compliance constraints, we compare experimental results between traditional ATS and our envisioned approach without updates. The test results based on the Synthetic and Helpdesk data set are provided in Table 2.

In general, our approach performs better than ATS in terms of all metrics in both data sets. In particular, a significant improvement can be observed for the synthetic data set. This is because the prediction model for ATS is solely trained based on the historic event log in which unseen activities like E, X, Y have never been observed. Thus, the prediction model is unable to deal with these unknown situations in online prediction without updating accordingly. However, our approach has considered unseen process behavior originated from the constraint set in the training phase already. This enables $AATS$ to cope with unseen cases even without updating strategies. For the Helpdesk data set, the overall predictive quality for both approaches is not as expected. Considering that the compliance constraint $c_1 = (\{Resolve\ ticket\}, \{Closed\}, \{directly\ follows\})$ is artificially created to test the presented approach. There are many violated cases against constraint c_1 in the test set. In this case, without updating the model during predictions hinders the possibility for both two models to capture those "violated" cases caused by the introduction of the artificially generated constraint. The slight improvement for $AATS$ compared to ATS in this data set can be attributed to the limited information (i.e., only one directly follows constraint with "Closed" as an unseen event label) we can obtain from constraints.

Table 2. Results ATS vs. AATS (no updates)

Data set	Approach	Accuracy	F1-score	Precision	Recall
Synthetic	ATS	0.250	0.249	0.313	0.250
	AATS	**0.815**	**0.810**	**0.897**	**0.815**
Helpdesk	ATS	0.432	**0.345**	0.363	0.432
	AATS	**0.450**	0.344	**0.567**	**0.450**

4.3 ATS vs. AATS with Updates

We assume that both approaches are capable of updating the prediction models if unseen behavior is observed in the event stream (cf. results in Table 3).

Compared to Sect. 4.2, if ATS is embedded with updating strategies during prediction, it outperforms our approach especially for the Synthetic data set. This is explainable since our approach strives to cover all possible variants of enriching the event log with additional information from constraints. Given that we use the probability of relative occurrences as the measurement function, the increase in the total amount of possible transitions for one state to another results in a decrease in the probability of the appropriate transition. Moreover, $AATS$ does not undergo updates as ATS did because the test set lacks unseen event labels for $AATS$ to update after the augmentation of constraints. Instead, it only updates the probability for each transition of the evolving stream. For the Helpdesk data set, the performances of both approaches with updating strategies are significantly improved compared to the results in Table 2. Since the test set contains lots of violated cases and unseen behavior (i.e., "Closed") for ATS, it is hard to tell the source of it's improvement. By contrast, the better performance of $AATS$ is owing to updating on those violations as we have incorporated the unseen event label into model training.

Table 3. Results ATS vs. AATS (with updates)

Data set	Approach	Accuracy	F1-score	Precision	Recall
Synthetic	ATS	**0.865**	**0.867**	**0.934**	**0.865**
	AATS	0.816	0.812	0.898	0.816
Helpdesk	ATS	0.705	0.670	**0.748**	0.705
	AATS	0.705	0.670	0.745	0.705

4.4 AATS vs. Deep-Learning Model Without Updates

Approaches using transition systems and deep-learning-based models without updating mechanisms are compared to provide insights into model selections in predictive process monitoring. Here we adopt Process Transformer proposed by [3] as the prediction model. The reason is that the Process Transformer uses the self-attention mechanism to reason over long-range dependencies and is able to process inputs in parallel [3]. We use the default hyperparameter configurations and enrich the training set with unseen event labels from the set of constraints. Then the prediction model (i.e., Process Transformer) is trained based on the enriched data set with all possible variations regarding constraints we considered. The test results are provided in Table 4.

For Synthetic data set, $AATS$ performs better than the deep-learning-based approach. However, the latter outperforms $AATS$ considerably on Helpdesk data set. This could be attributed to the type of constraints we considered. Two eventually follows constraints imposed on the Synthetic data set result in an enriched training set with plenty of variations, e.g., $< A, C, Y, B, X >$, while most of the augmented samples are incorrect cases that will not occur in the event stream (i.e., the test set only contains traces $< A, B, C, X, Y >$ and $< A, C, B, X, Y >$). Note that $AATS$ covers all these variations as well, but with the count of 1 for each variation regardless of the data size, whereas the deep-learning-based approach does augmentations for each individual trace if the partially seen criteria is satisfied. Thus, there are lots of incorrectly enriched cases in the training set. Conversely, the Helpdesk data set only considers one directly follows constraint, thus, no misleading samples are generated based on this constraint.

Table 4. Results DL vs. AATS (no updates)

Data set	Approach	Accuracy	F1-score	Precision	Recall
Synthetic	DL	0.732	0.714	0.729	0.732
	AATS	**0.815**	**0.810**	**0.897**	**0.815**
Helpdesk	DL	**0.844**	**0.812**	**0.796**	**0.844**
	AATS	0.450	0.344	0.567	0.450

4.5 AATS vs. Deep-Learning Model with Updates

To shed light on the effectiveness of the proposed approach, we conduct experiments comparing $AATS$ and existing approaches in dealing with unseen process behavior, i.e., updating deep-learning models on demand. We choose the most comparable deep-learning-based approach proposed in [14] with a single $LSTM$ layer. Following the experimental settings as stated in [14], we expand the Synthetic data set to include 1000 cases, of which 10% are allocated as a training set, and the remaining 90% are assigned to the test set. This means 100 cases are selected from the training set as we described in Sect. 4.1, and 900 cases are generated based on the process model with compliance constraints. The same training-test splitting is applied to Helpdesk data set as well. Moreover, the timestamp in the test set of the Synthetic data needs to be adapted to simulate an online setting as mentioned in [14], i.e., daily prediction. Experimental results are summarized in Table 5. Different update strategies provided in [14] are denoted as S_0 (do not update), S_1 (update on new activities), S_2 (update on new sequences) and S_3 (update every day). Time spent for each update strategy is represented as hours:minutes:seconds.

For both two data sets, $AATS$ underperforms the deep-learning approach with updating strategies like S_3 (update every day). By contrast, in terms of computational time, deep-learning approaches with updating mechanisms can take hours and even days to update prediction models when coping with relatively large data sets (e.g., Helpdesk data set).

Findings: To sum up, $AATS$ performs better than ATS if no updates are conducted during online prediction. When considering updates, ATS can learn from the proper incoming traces under the violation-free assumption directly. This leads to a better predictive performance than $AATS$ as it cannot handle

Table 5. Results DL vs. AATS (with updates)

Data set	Approach	Accuracy	F1-score	Precision	Recall	Time
Synthetic	S_0	0.403	0.401	0.498	0.403	00:00:01
	S_1	0.688	0.638	0.697	0.688	00:00:02
	S_2	0.403	0.401	0.498	0.403	00:00:01
	S_3	**0.829**	**0.828**	**0.899**	**0.829**	00:00:47
	AATS	0.813	0.809	0.896	0.813	00:00:02
Helpdesk	S_0	0.275	0.274	0.460	0.275	00:00:20
	S_1	0.657	0.652	0.659	0.657	00:04:46
	S_2	**0.778**	**0.756**	0.741	**0.778**	09:36:10
	S_3	0.776	0.755	**0.742**	0.776	28:42:32
	AATS	0.703	0.671	0.735	0.703	00:06:53

the reduction of the augmented transition system appropriately. In light of prediction model selections, deep-learning models demonstrate superior prediction quality compared to transition systems if eventually follows constraints are augmented properly. Nevertheless, it is computationally expensive for deep-learning-based approaches when considering update mechanisms. We discuss options for improvement in Sect. 6.

5 Related Work

This paper addresses research questions at the intersection of the i) exploitation of (external) context information and ii) handling of unseen behavior in predictive process monitoring. For i) [2,6,17] present a taxonomy for process context information, differentiating its origin into internal/intrinsic (within the log) and external (outside the log) context data. [25] distinguish between structured and unstructured context information, stating that *"the majority of the works on context-aware predictive business process monitoring relies on structured context data created by the information system itself like process performance metrics."*. [7] discover context-aware prediction models based on internal context data. [22] include internal, textual data as unstructured context data to predict the outcome of a running case. By contrast, [28] provide an approach for remaining time prediction by considering sentiments triggered by news (external context data). [5] propose an approach to identify context information for performance indicator prediction based on expert and domain knowledge. [21] exploit (external) sensor streams for predicting and explaining concept drift. The only approach exploiting information on compliance constraints in predictive process monitoring is [8] by ranking predictions based on their compliance with imposed constraints. However, the predictions are still based on observed behavior, i.e., they do not consider unseen behavior yet. Only few approaches envision strategies to cope with unseen behavior (ii). [4,20] elaborate update strategies for prediction models. Incremental learning for predicting the outcome of a process instance has been applied by [9,13]. [18] use incremental learning for predicting the next activity. In [14], we conclude that an ““*update on demand” strategy yields the best results in terms of balancing prediction quality and performance.”*. These approaches do not predict unseen behavior themselves as advocated in this work, but can be combined in order to deal with the evolving event stream.

6 Conclusion

This paper provides an approach to exploit contextual information from a set of compliance constraints C for predicting unseen behavior in the training data, i.e., in the event logs. For this, we augment state transition systems with the constraint information and use them as prediction model. A first insight is that for violation-free logs, i.e., logs that respect the compliance constraints we considered, the prediction quality is higher for the prediction with context information. When updating the models without context information with unseen behavior, the prediction quality converges to similar values. An interesting insight is that we drop the assumption of violation-free event logs, the prediction quality of the augmented prediction model might not exceed the quality of the prediction models without augmentation. Conversely, an unexpectedly low prediction quality of the approach with compliance constraint information can be interpreted as an indicator that the underlying logs contain violations, contributed by, for example, interleaving. Interleaving occurs if the directly follows semantic of a constraint is "broken" by executing an activity from another parallel branch. In these situations, the approach can also be used as a mechanism to detect possible compliance violations.

Discussion: We can understand the assumptions made in Sect. 2 as current limitations of the approach. At first, we only consider directly follows and eventually follows semantics of the constraints. However, compliance constraints can be more expressive, e.g., restricting the resource perspective via a separation of duty constraint. Second, we assume violation-free logs which might not be the case for real-world logs. In this case, prediction quality lower than expected can indicate compliance violations. Another limitation refers to the measurement function that is used for calculating the probabilities in the augmented state transition system. In this work, it uses occurrence frequencies. In future work, additional information can be exploited such as remaining time or data values from the event log to weigh probabilities differently. Third, we do not explicitly consider changes in constraint set C, though the presented approach is able to deal with such changes if they are under the partially seen scenario by re-running the augmentation algorithm.

Future work aims at exploiting compliance constraints referring to process perspectives, e.g., data attributes, beyond control flow when predicting next event labels. We will also test different abstraction functions, cf. [1]. Moreover, prediction goals such as remaining time will be added to the approach. Future work will also feature experiments with different update strategies in case of constraint set updates.

Acknowledgements. This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 277991500.

References

1. van der Aalst, W.M.P., Schonenberg, M.H., Song, M.: Time prediction based on process mining. Inf. Syst. **36**(2), 450–475 (2011). https://doi.org/10.1016/j.is.2010.09.001
2. Brunk, J., Stierle, M., Papke, L., Revoredo, K., Matzner, M., Becker, J.: Cause vs. effect in context-sensitive prediction of business process instances. Inf. Syst. **95**, 101635 (2021). https://doi.org/10.1016/j.is.2020.101635
3. Bukhsh, Z.A., Saeed, A., Dijkman, R.M.: Processtransformer: predictive business process monitoring with transformer network. CoRR abs/2104.00721 (2021). https://arxiv.org/abs/2104.00721
4. Chamorro, A.E.M., Nepomuceno-Chamorro, I.A., Resinas, M., Ruiz-Cortés, A.: Updating prediction models for predictive process monitoring. In: Advanced Information Systems Engineering, pp. 304–318 (2022). https://doi.org/10.1007/978-3-031-07472-1_18
5. Chamorro, A.E.M., Revoredo, K., Resinas, M., del-Río-Ortega, A., Santoro, F.M., Ruiz-Cortés, A.: Context-aware process performance indicator prediction. IEEE Access **8**, 222050–222063 (2020). https://doi.org/10.1109/ACCESS.2020.3044670
6. Ehrendorfer, M., Mangler, J., Rinderle-Ma, S.: Assessing the impact of context data on process outcomes during runtime. In: Hacid, H., Kao, O., Mecella, M., Moha, N., Paik, H. (eds.) ICSOC 2021. LNCS, vol. 13121, pp. 3–18. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91431-8_1
7. Folino, F., Guarascio, M., Pontieri, L.: Discovering context-aware models for predicting business process performances. In: Meersman, R., et al. (eds.) OTM 2012. LNCS, vol. 7565, pp. 287–304. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33606-5_18
8. 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
9. Francescomarino, C.D., Ghidini, C., Maggi, F.M., Rizzi, W., Persia, C.D.: Incremental predictive process monitoring: How to deal with the variability of real environments. CoRR abs/1804.03967 (2018). http://arxiv.org/abs/1804.03967
10. Hashmi, M., Governatori, G., Lam, H.-P., Wynn, M.T.: Are we done with business process compliance: state of the art and challenges ahead. Knowl. Inf. Syst. **57**(1), 79–133 (2018). https://doi.org/10.1007/s10115-017-1142-1
11. Hüllermeier, E., Waegeman, W.: Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Mach. Learn. **110**(3), 457–506 (2021). https://doi.org/10.1007/s10994-021-05946-3
12. Ly, L.T., Maggi, F.M., Montali, M., Rinderle-Ma, S., van der Aalst, W.M.P.: Compliance monitoring in business processes: functionalities, application, and tool-support. Inf. Syst. **54**, 209–234 (2015). https://doi.org/10.1016/j.is.2015.02.007
13. Maisenbacher, M., Weidlich, M.: Handling concept drift in predictive process monitoring. In: IEEE International Conference on Services Computing, pp. 1–8 (2017). https://doi.org/10.1109/SCC.2017.10

14. Mangat, A.S., Rinderle-Ma, S.: Next-activity prediction for non-stationary processes with unseen data variability. In: Almeida, J.P.A., Karastoyanova, D., Guizzardi, G., Montali, M., Maggi, F.M., Fonseca, C.M. (eds.) EDOC 2022. LNCS, vol. 13585, pp. 145–161. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17604-3_9
15. Mangler, J., Rinderle-Ma, S.: Cloud process execution engine: architecture and interfaces (2022). https://doi.org/10.48550/ARXIV.2208.12214
16. 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). https://doi.org/10.1109/TSC.2017.2772256
17. Park, G., Benzin, J., van der Aalst, W.M.P.: Detecting context-aware deviations in process executions. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds.) BPM 2022. LNBIP, pp. 190–206. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16171-1_12
18. Pauwels, S., Calders, T.: Incremental predictive process monitoring: the next activity case. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 123–140. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_10
19. Polato, M., Sperduti, A., Burattin, A., Leoni, M.: Time and activity sequence prediction of business process instances. Computing **100**(9), 1005–1031 (2018). https://doi.org/10.1007/s00607-018-0593-x
20. Rizzi, W., Di Francescomarino, C., Ghidini, C., Maggi, F.M.: How do I update my model? On the resilience of Predictive Process Monitoring models to change. Knowl. Inf. Syst. (9), 1–32 (2022). https://doi.org/10.1007/s10115-022-01666-9
21. Stertz, F., Rinderle-Ma, S., Mangler, J.: Analyzing process concept drifts based on sensor event streams during runtime. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 202–219. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_12
22. 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
23. 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). https://doi.org/10.1145/3301300
24. 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. Intell. Syst. Technol. **10**(4), 34:1-34:34 (2019). https://doi.org/10.1145/3331449
25. Weinzierl, S., Revoredo, K.C., Matzner, M.: Predictive business process monitoringwith context information from documents. In: 27th European Conference on Information Systems - Information Systems for a Sharing Society, ECIS 2019, Stockholm and Uppsala, Sweden, June 8–14, 2019 (2019). https://aisel.aisnet.org/ecis2019_rip/59
26. Weytjens, H., Weerdt, J.D.: Learning uncertainty with artificial neural networks for predictive process monitoring. Appl. Soft Comput. 109134 (2022). https://doi.org/10.1016/j.asoc.2022.109134

27. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. IEEE Trans. Pattern Anal. Mach. Intell. **41**(9), 2251–2265 (2019). https://doi.org/10.1109/TPAMI.2018.2857768
28. 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

The Interplay Between High-Level Problems and the Process Instances that Give Rise to Them

Bianka Bakullari[1(✉)], Jules van Thoor[2], Dirk Fahland[2], and Wil M. P. van der Aalst[1]

[1] RWTH Aachen University, Aachen, Germany
{bianka.bakullari,wvdaalst}@pads.rwth-aachen.de
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
d.fahland@tue.nl

Abstract. Business processes may face a variety of problems due to the number of tasks that need to be handled within short time periods, resources' workload and working patterns, as well as bottlenecks. These problems may arise locally and be short-lived, but as the process is forced to operate outside its standard capacity, the effect on the underlying process instances can be costly. We use the term *high-level behavior* to cover all process behavior which can not be captured in terms of the individual process instances. The natural question arises as to how the characteristics of cases relate to the high-level behavior they give rise to. In this work, we first show how to detect and correlate observations of high-level problems, as well as determine the corresponding (non-)participating cases. Then we show how to assess the connection between any case-level characteristic and any given detected sequence of high-level problems. Applying our method on the event data of a real loan application process revealed which specific combinations of delays, batching and busy resources at which particular parts of the process correlate with an application's duration and chance of a positive outcome.

Keywords: batch · workload · throughput time · outcome

1 Introduction

1.1 Motivation

Process mining techniques analyze event data stored in information systems in order to get insights of real business processes [1]. Organizations strive to improve their running processes by reducing cost and waste, improving resource utilization and customer satisfaction, and so on. Many Key Performance Indicators (KPIs) describe the process in terms of the individual process instances

We thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 145–162, 2023.
https://doi.org/10.1007/978-3-031-41623-1_9

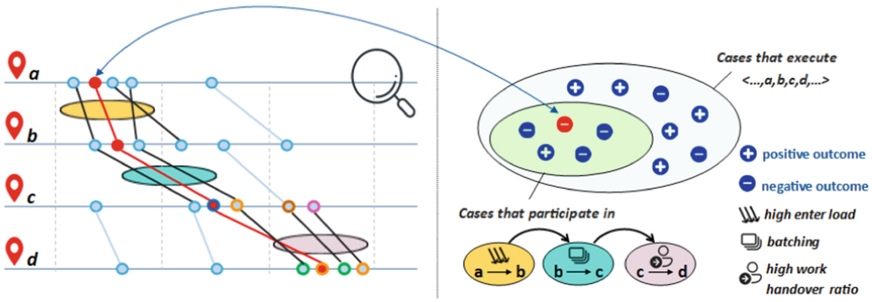


Fig. 1. An illustration of the approach: On the left, a sketch of the performance spectrum [5] showing process instances running through segments (a, b), (b, c) and (c, d). At each segment, several event pairs give rise to various patterns: high entering load at (a, b) (top cloud), batching at (b, c) (middle cloud), and high work handover ratio at (c, d) (bottom cloud). The cases which are involved in this pattern sequence are considered as "participating" w.r.t. that sequence. Given a case-level property, we analyze how its value changes when comparing these cases (e.g., the process outcome of the participating red case is negative) to the cases which visited the same locations where the pattern emerged (here (a, b), (b, c), and (c, d)), but did not give rise to such pattern.

(also called *cases*), e.g., by referring to the average time it takes for a case to complete the process (the *throughput time*), the average accumulated cost or positive outcome rate. Process instances, however, do not run in isolation. From this viewpoint, cases that are simultaneously active in a process resemble cars moving along traffic. Cars can cause traffic jams which, in turn, cause delays and accidents. Similarly, cases may overload the process and the workers, leading to congestion and delays. Moreover, when attending to multiple active cases, resources may execute work in *batches* which, in turn, also influences the manner in which process traffic moves forward. We refer to this kind of emergent process behavior, which is not detectable at the level of the individual instances, as *high-level behavior*. This behavior is *dynamic*; that is, it may arise locally and be short-lived, but it can have an influence on the process runs of the cases active at that time. Nevertheless, similar to traffic jams, there is always a specific set of cases involved whenever such behavior emerges. On one hand, the characteristics of a case may aggravate the emergence of high-level behavior, e.g., a demanding case can block resources for longer time periods. On the other hand, the outcome of a case can also be affected by high-level behavior occurring throughout its run, e.g., a case may not receive the necessary attention if by chance it enters the process in a busy period. There is obviously an interplay between the high-level behavior that arises in the process and the cases which give rise to it. Our method explores this interplay by detecting which patterns of high-level behavior emerge surprisingly often from specific case types. The advantage of possessing this knowledge can be manifold. Depending on the case property at hand, one can adjust the process for specific types of cases in order to avoid undesired but expected high-level problems, or one can make a better online prediction of the progress of a case given its involvement in specific patterns of high-level behavior.

1.2 Example

Suppose that the red lines in Fig. 1 describe the process run of a case which executes activities $\langle a, b, c, d \rangle$. Assume that, in this process, it is unusual to observe four cases entering segment (a, b) simultaneously in a short time frame (the lines within the top yellow cloud). Moreover, multiple cases including the red case execute activities b and c very close to each other with a very similar waiting time in-between (the lines within the middle blue cloud). Later in the process, four cases traverse segment (c, d) in a short time frame (the lines within the bottom violet cloud) and work is handed over from four resources (four different colors encircling the c events) to only two resources (only green and orange encircling the four d events). Assuming that this work handover ratio is unusually high, we can claim that in this example, the red case is involved in three patterns at three different process segments it traverses: *high enter load* at (a, b), *batching* at (b, c) and *high work handover ratio* at (c, d). Now suppose that the red case turns out to have a negative outcome in the process (see the red case's minus sign and positioning in the right side of Fig. 1). The question arises whether participating in this specific sequence of high-level patterns increases or decreases the likelihood of a negative outcome in the process. In this study, we evaluate whether a particular type of cases is disproportionately represented within the case set that generates a specific episode of high-level behavior. When such a situation occurs, we consider the connection between that specific episode and case property as a valuable insight into the behavior of the process.

1.3 Approach

As shown in the previous example, diverse types of high-level patterns can emerge at different locations and times within the process. To be able to compute how strongly cases' participation in high-level behavior correlates with any particular case characteristic, we first need to define what high-level behavior may look like (the clouds in Fig. 1). In this work, we introduce different types of high-level behavior at the segment level that relate to load (enter and exit rates), resource busyness (handover ratio and workload), and working patterns (batches and delays). We use the idea introduced in previous work [2] and conceptualize each outlier observation of such behavior using *high-level events*. It is worth emphasizing that multiple concurrently active cases can give rise to various forms of high-level behavior. This behavior can refer not only to process traffic and workload but also e.g., compliance with regulations which guide how the process should be executed given any particular set of active cases. Within this work, we outline high-level behavior that is specifically related to congestion as it is commonly observed across different process domains. The method could, however, be easily extended to any other type of high-level problem that arises from a set of cases that pass through a process segment in close time proximity. Given a high-level event, we determine what qualifies a case as "(non-)participating" (where to assign each case w.r.t. the sets depicted as circles in the right side of Fig. 1).

Any two high-level events with sufficient overlap in time, location and underlying case sets are assumed to be correlated, leading this way to sequences of subsequently connected high-level events (such as the sequence consisting of the three clouds in Fig. 1). A case participates in such a *high-level path* whenever its events are involved in each high-level event comprising the path. In Fig. 1, these are the cases whose black lines are caught up in all three clouds, such as the red case. In order to investigate the correlation between a case-level characteristic and a case's participation in a given high-level path, we compare the participating cases only to those cases which visit the same locations in the process where the high-level behavior was observed. In the example from Fig. 1, these would be segments (a, b), (b, c), and (c, d) in this order. Hence, in this work, we define the "non-participating" cases to be those case which are not participating, but could have participated from a control-flow perspective.

2 Related Work

Many of the recently developed process mining techniques acknowledge that the progress of cases in a process is influenced by the coexistence of other cases with which they must share process capacities. In [3], the authors aim to improve the rate of positive outcomes in the process. They propose appropriate interventions on changeable case aspects after having identified which treatments have a high causal effect on which case types. This idea is taken further in [4] where the appropriate time for applying a given time reducing intervention on a running case is determined. This decision is based on the causal effect of the intervention which, in turn, includes the number of active cases as an additional feature in the learning process. Results in [15] show that the prediction of case delay at a certain activity is improved when the model is either a transition system whose state space is extended with system load information, or the model is based on queueing theory. The method proposed in [13] shows how information regarding workload and resource availability can be extracted from raw event data and encoded into congestion graphs. From these congestion graphs, congestion-related features can be extracted which are then used for predicting the time until next activity. Inter-case dependencies are also acknowledged in [14] where the current case prediction also factors in the predictions of the cases coexisting with the current one. All these approaches acknowledge that process instances do not run in isolation and as such, integrating dedicated features which capture congestion to train time prediction models improves their accuracy. While many of the high-level patterns we analyze in this work relate to process congestion, we do not encode our observations as additional features describing our running cases for prediction purposes. Instead, each sequence of recurring outlier observations represents an explicit variant of high-level behavior which is a trait of the process itself. We then look back into the "low-level" cases which were part of these observations to reason on whether a specific case type is over- or underrepresented in this group.

The performance spectrum [6] clearly showed that processes—even within the same segment—exhibit non-stationary behavior which is not observable under aggregation. The emerging patterns can reveal batching behavior [8] which can have an influence on performance. In [9], the authors use visual analytics techniques based on the performance spectrum to demonstrate how errors in remaining time prediction are reduced when information on batching behavior is encoded in the learning process. Our method conceptualizes many of the patterns that can be seen in the performance spectrum—including batching—through dedicated events, which can then be mined for further automated analysis.

Several recent methods have been developed which analyze how resources handle tasks from concurrently active cases within a process. In [16], the authors provide insights into how resources prioritize their work by employing specific queueing disciplines when processing individual tasks. In [10], the authors detail the detection of batching behavior not only at the level of individual tasks but also across several linked activities. Moreover, the approach outlined in [11] considers various batch behaviors concerned with multiple perspectives such as activities, resources, and data perspectives as well as allowing for batch detection even when batching is temporarily interrupted. In [18], the authors introduce an enhanced resource profiling technique that considers not only the executed activities, but also the context (duration, case attributes) as well as multitasking. In our work, we incorporate the resource dimension within the high-level events that describe outlier observations concerning batching and workload in specific process segments. However, these observations are local and temporary and their emergence becomes relevant only in relation to the types of cases involved.

In [7], the authors introduce the concept of contextual association, wherein a group of cases exclusively exhibits concept drift whenever a shared object is subject to a change. In our work, cases which give rise to high-level behavior are contextually associated due to their shared location and time in which the behavior is observed. In this scenario, the term "context" solely represents the *coordinate* of a temporary observation in the process.

The idea of conceptualizing outlier behavior related to load and delays as events themselves was first introduced in [17]. In that work, the emerging *system-level events* from a Baggage Handling System (BHS) were connected based on time and place proximity. The resulting cascades revealed how undesired system-level behavior arose and propagated throughout the BHS. Similar work extending this idea was done in [19] where DBSCAN is used to find frequent sequences of anomalies arising at the system-level.

The method we propose in this paper fits in the *high-level event mining* framework we introduced in [2]. Each high-level event consists of the type of behavior detected, the entity involved and the time of detection. In this work, we extend the types of high-level events that can be observed at the segment-level and propose a more refined way of correlating them. Ultimately, we take a look at the underlying process instances and explore whether associations exist.

3 Preliminaries

Definition 1 (Power set, Sequence, Suffix). *Given a set A, $\mathcal{P}(A)$ is the* power set *of A and A^* are the finite sequences over A. For any $s, s' \in A^*$, we say $s' = \langle a'_1, ..., a'_m \rangle$ is a* suffix *of $s = \langle a_1, ..., a_n \rangle$ (denoted $s' \preceq s$) if and only if there is some $i \in \{0, ..., n-m\}$ such that for $j = 1, ..., m$ it holds that $a'_j = a_{i+j}$.*

Definition 2 (Events, Event log). *$\mathcal{U}_{ev}$ is the* universe of events *and Act, $Case$, Res are the sets of* activity names, case identifiers *and* resource names, *respectively. T is the totally ordered set of* timestamps. *$L = (E, Attr, \pi)$ is an* event log *where $E \subseteq \mathcal{U}_{ev}$ is a finite set of events, $\{act, case, res, time\} \subseteq Attr$ is a set of attribute names and $\pi \in E \times Attr \not\rightarrow Val$ a (partial) function that assigns each event e a value $\pi(e, att)$ or is undefined (written $\pi(e, att) = \perp$). For any $e \in E$, $\pi(e, act) \in Act$, $\pi(e, case) \in Case$, $\pi(e, res) \in Res$, and $\pi(e, time) \in T$.*

For any attribute $att \in Attr$, we write $att(e)$ instead of $\pi(e, att)$ when the event log is clear from the context. Moreover, we assume that any two events of the same case never have identical timestamps.

Definition 3 (Traces, Steps, Segments). *The cases of an event log $L = (E, Attr, \pi)$ are $cases(L) = \{case(e) \mid e \in E\}$. For any case $c \in cases(L)$ with corresponding event set $E_c = \{e \in E \mid case(e) = c\}$, the* trace *of c is the sequence $\sigma(c) = \langle e_1, ..., e_{|E_c|} \rangle \in E_c^*$ containing all events from E_c ordered by time, i.e., $\forall_{1 \leq i < j \leq |E_c|}\ time(e_i) < time(e_j)$. A* step *is a pair of directly following events in a case in L. More precisely, the steps of L are $steps(L) = \{(e, e') \in E \times E \mid \exists_{c \in cases(L)}\ \sigma(c) = \langle ..., e, e', ... \rangle\}$. Moreover, we define $S(L) = \{(act(e), act(e')) \mid (e, e') \in steps(L)\}$ as the* segments *of L.*

A step is a pair of events which happened directly after each other in the same case. A segment is a pair of activities that directly follow each other in the log.

Definition 4 (Framing, Time Windows). *A* framing *is a function $\phi \in T \rightarrow \mathbb{N}$ mapping timestamps to numbers such that $\forall_{t_1, t_2 \in T}\ t_1 < t_2 \Rightarrow \phi(t_1) \leq \phi(t_2)$. Each $w \in rng(\phi)$ represents time window $\overrightarrow{w} = [w_{start}, w_{end}]$, where $w_{start} = min\{t \in T \mid \phi(t) = w\}$ and $w_{end} = max\{t \in T \mid \phi(t) = w\}$.*

Given an event log $L = (E, Attr, \pi)$ and a framing ϕ, set $W_{L,\phi} = \{w \in \mathbb{N} \mid min\{\phi(time(e)) \mid e \in E\} \leq w \leq max\{\phi(time(e)) \mid e \in E\}\}$ contains all time windows of L w.r.t. framing ϕ. Note that for any $e \in E$, $\phi(time(e)) = w$ whenever e occurred within $\overrightarrow{w}$.

4 Method

4.1 Detecting High-Level Behavior Using High-Level Events

The example in Sec. 1.2 illustrated three important components which comprise high-level behavior: the *type* of behavior observed, the *location* in the process where it emerges and the *time* aspect related to it. We call each pair of location and time information a *coordinate*.

Definition 5 (Coordinates). *Given log $L = (E, attr, \pi)$ and window set $W_{L,\phi}$ w.r.t framing ϕ, let $W^2_{L,\phi} = \{(w_1, w_2) \in W_{L,\phi} \mid w_1 \leq w_2\}$. The set $CO(L, \phi) = S(L) \times (W_{L,\phi} \cup W^2_{L,\phi})$ contains the* coordinates *of log L w.r.t. ϕ. Each coordinate $co = (s, \theta) \in CO(L, \phi)$ refers to a position in space (segment s) and time (window if $\theta \in W_{L,\phi}$ and window pair if $\theta \in W^2_{L,\phi}$).*

Each outlier observation we considered in Sec. 1.2 emerged from a specific set of steps. Which steps were involved in the observation depended on the type of behavior we were looking for. For instance, the steps in the first cloud in Fig. 1 represent the incoming load at segment (a, b) within a particular time window. Next, we conceptualize the colored clouds and the steps that are involved in them using *high-level features*. Each high-level feature consists of its type and a pattern. One can think of the type being the color of the cloud and the pattern being the function which determines which subset of steps occurring in a given coordinate may give rise to that type of feature.

Definition 6 (Pattern, Feature type). *Given a log $L = (E, attr, \pi)$ and framing ϕ, a* pattern *is a (partial) function $\rho_{L,\phi} \in CO(L, \phi) \nrightarrow \mathcal{P}(E \times E) \times \mathbb{R}$ which assigns a set of event pairs and a number to each coordinate of L and ϕ. A* high-level feature *w.r.t. L and ϕ is a pair $hlf = (type, \rho_{L,\phi})$ where $type \in \mathcal{U}_{type}$ is a feature type from the universe $\mathcal{U}_{type}$ of feature types and $\rho_{L,\phi}$ is its pattern.*

In the remainder, given log L and framing ϕ, we write $\rho^{type}_{L,\phi}$ to refer to the pattern of the high-level feature of type *type*.

In this work, we consider feature types *enter*, *exit*, *workload*, *handover*, *batch*, and *delay*. For each of these feature types, we now show how their patterns are determined. Let $co = (s, w) \in S(L) \times W_{L,\phi}$ be a coordinate from log L with time windows from framing ϕ. Let $I_s = \{(e, e') \in steps(L) \mid (act(e), act(e')) = s\}$ be the event pairs (steps) that traverse segment s in the process and let $I_w = \{e \in E \mid time(e) \in \overrightarrow{w}\}$ be the events that occur within time window w. Feature type *enter* is concerned with the steps that enter segment s during $\overrightarrow{w}$ and thus $\rho^{enter}_{L,\phi}(co) = (I^{enter}, val)$ where $I^{enter} = \{(e, e') \in I_s \mid e \in I_w\}$ and $val = |I^{enter}|$. Feature type *exit* is concerned with the steps that exit segment s during $\overrightarrow{w}$ and thus $\rho^{exit}_{L,\phi}(co) = (I^{exit}, val)$ where $I^{exit} = \{(e, e') \in I_s \mid e' \in I_w\}$ and $val = |I^{exit}|$. Feature type *workload* is concerned with the steps that exit segment s during $\overrightarrow{w}$ for which it is the same resource executing both activities of s. Thus, $\rho^{workload}_{L,\phi}(co) = (I^{wld}, val)$ where $I^{wld} = \{(e, e') \in I_s \mid e' \in I_w \wedge res(e) = res(e')\}$ and $val = |I^{wld}|$. Conversely, feature type *handover* is concerned with the steps that exit segment s during $\overrightarrow{w}$ for which there were two different resources executing the activities of s (and so real work handover took place). Hence, $\rho^{handover}_{L,\phi}(co) = (I^{hdo}, val)$ where $I^{hdo} = \{(e, e') \in I_s \mid e' \in I_w \wedge res(e) \neq res(e')\}$ and $val = |I^{hdo}|$.

The time aspect of steps which are handled in batches refers to two time windows. Let $(w, w') \in W^2_{L,\phi}$ be a pair of time windows and let $co = (s, (w, w')) \in S(L) \times W^2_{L,\phi}$ be a coordinate. Let $I_{w,w'} = \{(e, e') \in steps(L) \mid time(e) \in \overrightarrow{w} \wedge time(e') \in \overrightarrow{w'}\}$ be the set of event pairs (steps) where the first event occurs during w and the second event occurs during w'. Feature type *batch* is concerned with the steps that enter segment s during $\overrightarrow{w}$ and exit s during $\overrightarrow{w'}$.

Thus, $\rho_{L,\phi}^{batch}(co) = (I_s \cap I_{w,w'}, |I_s \cap I_{w,w'}|)$. Given a window distance $\delta \in \mathbb{N}$, the *delay* w.r.t. to δ is concerned with the steps that together experience a similar delay which is at least δ. I.e., $\rho_{L,\phi}^{delay}(co) = (I^{delay}, val)$ where $val = |I^{delay}|$ and $I^{delay} = I_s \cap I_{w,w'}$ if $w' - w \geq \delta$ and $\emptyset$ otherwise.

Definition 7 (High-level event). *Let L be an event log, ϕ a framing and $Type \subseteq \mathcal{U}_{type}$ a set of feature types. Let $thr \in Type \times S(L) \rightarrow \mathbb{R}$ be a function assigning a threshold to any type-segment pair. We observe* high-level event *$h = (type, co) \in Type \times CO(L, \phi)$ with $co = (s, \theta)$ if and only if $\rho_{L,\phi}^{type}(co) = (I^{type}, val)$ and $val \geq thr(type, s)$. Moreover, we call $C(h) \subseteq cases(L)$ the* cases of *h and for any $c \in cases(L)$, it holds that $c \in C(h)$ if and only if there exists a step $(e, e') \in I^{type}$ with $case(e) = c$. The set $\mathcal{H}_{L,\phi,Type,thr}$ contains all high-level events observed w.r.t. L, ϕ, $Type$, and thresholds from thr.*

For any of the six feature types described above, a high-level event of type *type* is observed at coordinate *co* if and only if the number of event pairs that comprise the pattern related to *type* at that coordinate is higher than a given threshold. Note that we propose the threshold to be determined based on both the feature type and the segment. For instance, for any segment s, one observes a high-level event of type *exit* at coordinate (s, w) whenever the number of steps leaving s during w is above the pth percentile of all numbers of steps which leave s in any given window throughout the process.

In the remainder of this work, we fix L, ϕ, $Type$ and thr and set $\mathcal{H} = \mathcal{H}_{L,\phi,Type,thr}$.

4.2 Connecting High-Level Events

The various high-level events observed throughout the process are not independent of each other. As each high-level event relates to a time and place of occurrence, one can reason about time and space proximity. Moreover, many of the steps in the patterns that give rise to them may belong to the same cases. It seems natural to connect the three high-level events of types *enter*, *batch* and *handover* observed in Fig. 1 into a single *episode* of high-level behavior. That is because going from one high-level event to the next, one can notice that the cases involved have high overlap (1), and the first high-level event "stops" at the same place (2) and at the same time period (3) where the second one "begins". We use the terms *case overlap*, *location overlap*, and *time overlap* to refer to these connection criteria and we introduce them formally in this section.

Definition 8 (Start Spread, End Spread). *Let $h = (type, co) \in \mathcal{H}$ be a high-level event and let $\rho_{L,\phi}^{type}(co) = (I^{type}, val)$. We refer to time period*

$$start(h) = [min\{time(e) \mid (e, e') \in I^{type}\}, max\{time(e) \mid \{(e, e') \in I^{type}\}]$$

as the start spread *of h. Similarly, we refer to*

$$end(h) = [min\{time(e') \mid (e, e') \in I^{type}\}, max\{time(e') \mid \{(e, e') \in I^{type}\}]$$

as the end spread *of h.*

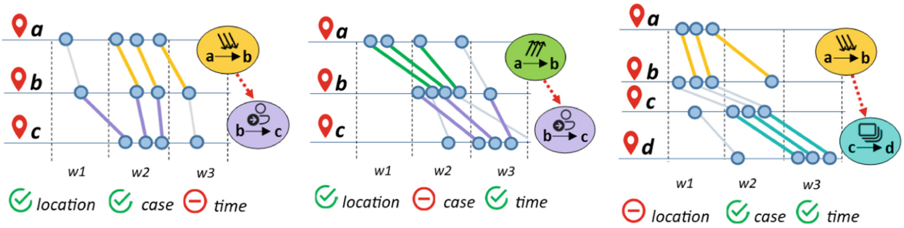


Fig. 2. For $\lambda = 0.5$, each figure shows an example where one overlap criterion from Def. 9 is not satisfied. In the left figure, high-level events $(enter, ((a, b), w_2))$ and $(handover, ((b, c), w_2)$ have no time overlap. In the middle figure, $(exit, ((a, b), w_2))$ and $(handover, ((b, c), w_3)$ have no sufficient case overlap. In the right figure, $(enter, ((a, b), w_1))$ and $(batch, ((c, d), (w_2, w_3)))$ have no location overlap.

In other words, given a high-level event related to segment (a, b), the start spread covers the time period between the first and the last executions of a from the steps in the corresponding pattern. Similarly, the end spread covers the time period between the first and the last executions of b from those same steps.

Definition 9 (Overlap, Propagation). *Let $h = (type, co), h' = (type', co') \in \mathcal{H}$ be two high-level events with $co = (s, \theta)$ and $co' = (s', \theta')$. Given some $\lambda \in [0, 1]$, we say pair (h, h') has* case overlap *w.r.t. λ (denoted $h \overset{case}{\rightsquigarrow}_\lambda h'$) if and only if $\frac{|C(h_1) \cap C(h_2)|}{|C(h_1) \cup C(h_2)|} \geq \lambda$. We say pair (h, h') has* location overlap *(denoted $h \overset{loc}{\rightsquigarrow} h'$) if and only if $s = (a, b)$, $s' = (a', b')$ and $b = a'$. Moreover, we say pair (h, h') has* time overlap *(denoted $h \overset{time}{\rightsquigarrow} h'$) if and only if either $end(h_1) \subseteq start(h_2)$ or $start(h_2) \subseteq end(h_1)$. Ultimately, we say there is* propagation *from h to h' w.r.t. λ (denoted $h \rightsquigarrow_\lambda h'$) if and only if the pair (h, h') has case overlap w.r.t. λ, location overlap and time overlap. More precisely:*

$$\forall_{h,h' \in \mathcal{H}} \; h \rightsquigarrow_\lambda h' \;\Leftrightarrow\; h \overset{case}{\rightsquigarrow}_\lambda h' \;\wedge\; h \overset{loc}{\rightsquigarrow} h' \;\wedge\; h \overset{time}{\rightsquigarrow} h'.$$

Two high-level events have time overlap whenever the end spread of the first one is contained in the start spread of the second one or the other way around. Figure 2 depicts examples of high-level event pairs that satisfy two of the overlap criteria, but not the third one.

Whenever a pair of high-level events are close in time and space and their cases overlap sufficiently, we assume that their observations are correlated and we say that the first high-level event propagates to the second one. Subsequent pairs of high-level events for which propagation occurs can lead to sequences which we call *high-level episodes*.

Definition 10 (High-level episode). *Given high-level event set $\mathcal{H}$ and some case overlap threshold λ, any sequence of high-level events $\varepsilon = \langle h_1, ..., h_n \rangle \in \mathcal{H}^*$ creates a* high-level episode *if and only if $\forall_{1 \leq i < n} \; h_i \rightsquigarrow_\lambda h_{i+1}$ and $\bigcap_{h \in \varepsilon} C(h) \geq \lambda$. The set $\mathcal{E}(\mathcal{H}, \lambda)$ contains all such high-level episodes.*

In order to reason about recurring behavior, we abstract from the time component describing the high-level event and focus instead only on the type and location of the corresponding observation (the *high-level activity*). Moreover, we lift this concept to episodes and call the projection of an episode onto its high-level activities a *high-level path*.

Definition 11 (High-level path). *Let $\mathcal{H}$ be a set of high-level events and $\lambda \in [0,1]$. For any $h = (type, co) \in \mathcal{H}$ with $co = (s, \theta)$, the* high-level activity *of h is $h\!\uparrow = (type, s)$. For any episode $\varepsilon = \langle h_1, ..., h_n \rangle \in \mathcal{E}(\mathcal{H}, \lambda)$, the sequence $\varepsilon\!\uparrow = \langle h_1\!\uparrow, ..., h_n\!\uparrow \rangle$ is its corresponding* high-level path. *Multiset $P(\mathcal{H}, \lambda) = [\varepsilon\!\uparrow \mid \varepsilon \in \mathcal{E}(\mathcal{H}, \lambda)]$ contains all such high-level paths.*

Note that while each episode is unique because the high-level events are unique, the high-level paths may be recurring for different episodes. It is for these recurring paths that we want to investigate the correlation with the properties of the cases involved.

4.3 Case Participation in High-Level Behavior

The participating cases of a given high-level path are those which are involved in all high-level events of an episode that executes the corresponding path. The non-participating cases are those which are not participating, but which traverse the process segments underlying the path throughout their process run.

Definition 12 ((Non-)participating cases). *Given high-level event set $\mathcal{H}$ and case overlap threshold λ, let $p = \langle h_1, ..., h_n \rangle \in P(\mathcal{H}, \lambda)$ be a high-level path and for each $i \in \{1, .., n-1\}$, let $s_i = (a_i, a_{i+1})$ be the segment where the high-level event h_i was observed. The* participating cases *of p are $C_p = \{c \in cases(L) \mid \exists_{\varepsilon \in \mathcal{E}(\mathcal{H},\lambda)}\ \varepsilon\!\uparrow = p \ \wedge\ c \in \bigcap_{h \in \varepsilon} C(h)\}$. The* non-participating cases *of p are $\overline{C_p} = \{c \in cases(L) \setminus C_p \mid \sigma(c) = \langle e_1, ..., e_k \rangle \ \wedge\ \langle a_1, a_2, ..., a_n \rangle \preceq \langle act(e_1), ..., act(e_k) \rangle\}$.*

To measure the correlation of a case-level attribute and a high-level path, the set of cases $C_p \cup \overline{C_p}$ is additionally partitioned according to the chosen case attribute value (see Table 1). The correlation is then computed using the χ^2 test of independence on these two partitions. The χ^2 test measures the difference between the observed and expected frequencies for each combination of the values of two categorical variables. The null hypothesis states that there is no relationship between case participation in a given high-level path and the chosen case attribute. We consider the correlation as being statistically significant, and thus reject the null hypothesis, if the corresponding p-value of the result is smaller than 0.05.

5 Evaluation

To evaluate our method, we used the BPI Challenge 2017 log[1], which corresponds to a loan application process performed in a financial institution. Each case in

[1] https://data.4tu.nl/articles/dataset/BPI_Challenge_2017/12696884.

Table 1. Given some high-level path p, the participating and non-participating case sets C_p and $\overline{C_p}$ are further split based on the chosen categorical attribute values (here: category 1 and category 2). The correlation between the attribute and the high-level path is computed using the χ^2 test of independence on the row partition (the chosen case-level attribute) and on the column partition ((non-)participation in the high-level path).

Case-level attribute	Participating C_p	Non-participating $\overline{C_p}$
category 1	n_1	n_2
category 2	n_3	n_4
	$n_1 + n_3 = \|C_p\|$	$n_2 + n_4 = \|\overline{C_p}\|$

this event log is an application. The applications can result in being successful or unsuccessful. Moreover, the duration of applications varies from less than 10 d to over 30 d. In this section, we analyze which sequences of high-level activities are strongly associated with the case attributes *outcome* and *throughput time*. In the following, we briefly describe the event log, the setup of our experiments and some general statistics over the detected high-level events. Afterwards, we comment on some of the high-level paths which showed a statistically significant correlation with the outcome and the throughput time of cases. The method together with the evaluation script is available as a Python implementation[2].

5.1 The BPI Challenge 2017 Event Log

The Application log of the BPIC 2017 contains a total of 31509 applications from January 2016 to February 2017. The general control-flow of an application can be described as follows: first, a request for a loan is made. Then, the submitted application is assessed. If a credit offer can be made, the bank composes the offer and sends it to the customer. Some time later, a bank employee calls the customer to remind them about the offer and answer any possible questions. The customer sends all necessary documents to the bank which in turn has to validate them. If the documents are incomplete, the customer is informed by a bank employee that that they have to resend the documents. If the documents are complete, the bank either composes another offer, or it makes the ultimate decision to either accept or deny the application. The activities of this process are divided into three categories: *Application State Changes* (preceded by A_), *Offer State Changes* (preceded by O_), and *Workflow Events* (preceded by W_). The workflow events additionally contain *lifecycle information* (e.g. schedule, start, suspend, resume, complete, ate_abort/withdraw). As the steps where cases move from one state of a workflow event to another make up for a significant amount of waiting time and rework in the process, we classify workflow related activities using both the event type and its lifecycle information. This results in activities like W_Call after offers|SUSPEND or W_Call after offers|RESUME.

[2] https://github.com/biankabakullari/hlem-framework.

Table 2. The total number of observed high-level events in the loan application log. For each feature type, one can see the absolute and relative number of high-level events of that type, the number of distinct segments where those high-level events were observed and the segment where they were observed most often.

feature type	# hle (%)	# distinct segments	most frequent segment
workload	1103 (20,82 %)	36	(W_Call incomplete files|schedule, W_Call incomplete files|start)
handover	214 (4,04%)	10	(W_Call incomplete files|suspend, W_Call incomplete files|resume)
enter	1394 (26,31%)	43	(A_Create Application, A_Submitted)
exit	1377 (25,99%)	43	(A_Create Application, A_Submitted)
batch	1048 (19,78%)	11	(W_Call after offers|suspend, W_Call after offers|ate_abort)
delay	162 (3,06%)	5	(W_Validate application|suspend, W_Validate application|resume)

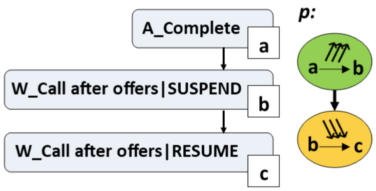
5.2 Experimental Setting and General Statistics

Before applying our method on the event log, we projected the traces onto the 43 most frequent segments. We split the time scope of the event data onto 398 windows, each corresponding to exactly one day. We evaluated the patterns related to feature types *enter*, *exit*, *workload*, *handover*, *batch* and *delay* throughout the entire coordinate space. Moreover, for *workload* and *handover*, we only considered human resources. Here, it means that we removed User_1 which was a system resource, and considered only real employees of the bank (User_2 to User_149). Each observation counted as a high-level event whenever the measured number in the corresponding pattern was at least as high as the 90th percentile of all the values obtained for that same type-segment pair. For delays, we chose a δ based on the 70th percentile of the days it takes to traverse a particular segment. This resulted in a total of 5298 high-level events. Table 2 shows the absolute and relative frequencies of high-level events for each feature type, together with the number of distinct segments where that type of high-level events was observed. One can notice how the activities related to application files being incomplete (W_Call incomplete files), the communication with the customers (W_Call after offers) and the application validation (W_Validate application) are most often subject to high-level behavior.

We connected the high-level events into episodes using a case overlap threshold of $\lambda = 0.5$. This generated 102060 episodes, which corresponded to 68538 distinct high-level paths. For these paths, we investigated the correlation with the *outcome* and *throughput time* of cases.

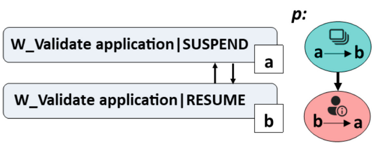
5.3 Outcome: Success Rate

The *success rate* refers to the number of times an application results in a positive outcome (customer accepts an offer and the loan is granted) divided by the total number of applications. In our event log, a *successful* case translates into its trace containing activity A_Pending. In total, 17228 (54.85%) cases are successful and the other 12183 (45.15%) cases are unsuccessful. The latter are the cases where the loan is either denied by the bank or cancelled by the customer.



outcome	C_p	$\overline{C_p}$
successful	1087 (51,06%)	11106 (53,48%)
unsuccessful	1042 (48,94%)	9661 (46,52%)
Total: 22896	$\lvert C_p \rvert = 2129$	$\lvert \overline{C_p} \rvert = 20767$

Fig. 3. The participating (C_p) and non-participating ($\overline{C_p}$) cases of the high-level path $p = \langle (exit, (a, b)), (enter, (b, c)) \rangle$ where a = A_Complete, b = W_Call after offers|SUSPEND, and c = W_Call after offers|RESUME. This path was observed 14 times in the event log. Here, $\chi^2 \approx 4,55$ and $p = 0,0329$.

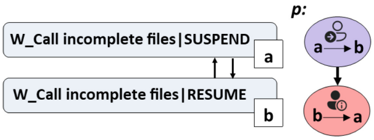


outcome	C_p	$\overline{C_p}$
successful	368 (72,73%)	7160 (77,93%)
unsuccessful	138 (27,27%)	2028 (22,07%)
Total: 9694	$\lvert C_p \rvert = 506$	$\lvert \overline{C_p} \rvert = 9188$

Fig. 4. The participating (C_p) and non-participating ($\overline{C_p}$) cases of the high-level path $p = \langle (batch, (a, b)), (workload, (b, a)) \rangle$ where a = W_Validate application|SUSPEND and b = W_Validate application|RESUME. This path was observed 10 times in the event log. Here, $\chi^2 \approx 7,48$ and $p = 0,0063$.

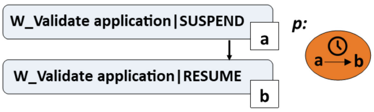
Next, we show four frequent high-level paths which showed a statistically significant correlation with the case success rate. For the two possible outcomes of *success*, a significant correlation is observed whenever $\chi^2 \geq 3.841$.

The path in Fig. 3 shows that the success rate is lower for the cases which in large groups simultaneously go from having completed the application into the part where the bank initiates communication with them. Moreover, it seems that for many cases in the process, the activities W_Validate application and W_Call incomplete files are first suspended, then resumed and afterwards suspended again. Suspending after resuming seems to be associated with high resource workload (both paths in Fig. 4 and 5). This high workload is preceded by batching behavior (Fig. 4) and high work handover ratio (Fig. 5). Participation in both these high-level paths seems to also be negatively associated with the case success rate. Additionally, cases whose validation is resumed in batches with a long waiting time after the suspension (Fig. 6) seem to also show lower success rates than the cases whose validation is suspended and then later resumed with a shorter period in-between.



outcome	C_p	$\overline{C_p}$
successful	852 (79,04%)	10668 (85,05%)
unsuccessful	226 (20,96%)	1875 (14,95%)
Total: 13621	$\lvert C_p \rvert = 1078$	$\lvert \overline{C_p} \rvert = 12543$

Fig. 5. The participating (C_p) and non-participating ($\overline{C_p}$) cases of the high-level path $p = \langle (handover, (a, b)), (workload, (b, a)) \rangle$ where a = W_Call incomplete files|SUSPEND and b = W_Call incomplete files|RESUME. This path was observed 16 times in the event log. Here, $\chi^2 \approx 27,54$ and $p \approx 1,53 \cdot 10^{-7}$.



outcome	C_p	$\overline{C_p}$
successful	928 (72,22%)	7822 (76,81%)
unsuccessful	357 (27,78%)	2362 (23,19%)
Total: 11469	$\lvert C_p \rvert = 1285$	$\lvert \overline{C_p} \rvert = 10184$

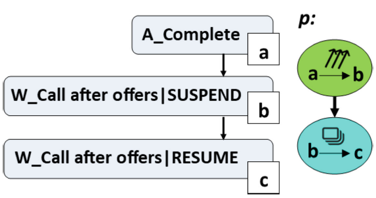
Fig. 6. The participating (C_p) and non-participating ($\overline{C_p}$) cases of the high-level path $p = \langle (delay, (a, b)) \rangle$ where a = W_Validate application|SUSPEND and b = W_Validate application|RESUME. This path was observed 66 times in the event log. Here, $\chi^2 \approx 13,28$ and $p = 0,0002676$.

5.4 Throughput Time

The throughput time of a case is the time elapsed between the case's first and last event. According to [12], one can notice clear trends in the progress among the applications which take between 10–30 days to complete, and the applications which finish faster or pass the 30 d mark. We use these throughput time categories to analyze the influence of our high-level paths on case duration. In total, 7454 (23,73%) cases finish in less than 10 d, 12963 (41,27%) cases take between 10 and 30 d, and 10994 (35,00%) cases spend longer than 30 d in the process.

Next, we show three high-level paths which showed a strong association with the case throughput time. While it is unsurprising that high-level problems delay the progress of cases, our analysis reveals more detailed insights into what type of subsequent problems emerging at which process segments show particularly high association with the case duration. For the three throughput time categories, a significant correlation is observed whenever $\chi^2 \geq 5.991$.

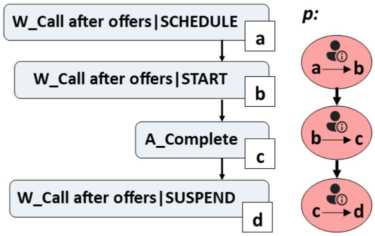
Similarly to Fig. 3, the path in Fig. 7 shows that the cases which simultaneously transition from having completed the application into a part where the bank attempts to communicate with them in batches, also suffer longer throughput times. One can notice that there are more cases that pass the 30 d' mark and less cases that finish in under 10 d from the group of participating cases than from the group of non-participating cases. Moreover, having overloaded employees taking care of the communication process with the customers (the path in Fig. 8) also correlates with longer case throughput times (especially as the ratio of cases finishing in under 10 d decreases). The path in Fig. 9 covers the scenario when a case is validated, an offer is returned, but then the validating process has to be suspended. It seems that for the cases which experience batching and high workload in this process part, the throughput time worsens.



high exit load *batching*

thr. time	C_p	$\overline{C_p}$
$\leq$ *10 days*	273 (13,75%)	3860 (18,46%)
10-30 days	848 (42,72%)	8984 (42,96%)
$\geq$ *30 days*	864 (45,53%)	8069 (38,58%)
Total: 22898	$\lvert C_p \rvert = 1985$	$\lvert \overline{C_p} \rvert = 20913$

Fig. 7. The participating (C_p) and non-participating ($\overline{C_p}$) cases of the high-level path $p = \langle (exit, (a, b)), (batch, (b, c)) \rangle$ where a = A_Complete, b = W_Call after offers|SUSPEND, and c = W_Call after offers|RESUME. This path was observed 15 times in the event log. Here, $\chi^2 \approx 33,61$ and $p \approx 5,04 \cdot 10^{-8}$.

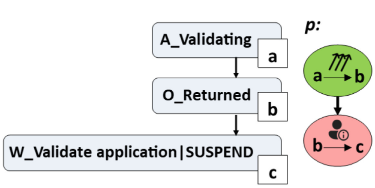


high workload

thr. time	C_p	$\overline{C_p}$
$\leq$ **10 days**	224 (18,73%)	6709 (23,20%)
10-30 days	550 (45,99%)	12006 (41,53%)
$\geq$ **30 days**	422 (35,28%)	10197 (35,27%)
Total: 30108	$\lvert C_p \rvert = 1196$	$\lvert \overline{C_p} \rvert = 28912$

Fig. 8. The participating (C_p) and non-participating ($\overline{C_p}$) cases of the high-level path $p = \langle (workload, (a, b)), (workload, (b, c)), (workload, (c, d)) \rangle$ where a = W_Call after offers|SCHEDULE, b = W_Call after offers|START, c = A_Complete, and d = W_Call after offers|SUSPEND. This path was observed 15 times in the event log. Here, $\chi^2 \approx 15,47$ and $p = 0,000437$.

To conclude, we noticed that participation in high-level behavior was negatively associated with both *outcome* and *throughput time* as the participating cases showed lower success rates and higher throughput times compared to the non-participating cases.



thr. time	C_p	$\overline{C_p}$
$\leq$ 10 days	448 (26,18%)	5067 (30,28%)
10-30 days	1019 (59,56%)	9542 (57,02%)
$\geq$ 30 days	244 (14,26%)	2125 (12,70%)
Total: 18445	$\lvert C_p \rvert = 1711$	$\lvert \overline{C_p} \rvert = 16734$

Fig. 9. The participating (C_p) and non-participating ($\overline{C_p}$) cases of the high-level path $p = \langle (exit, (a, b)), (workload, (b, c)) \rangle$ where a = A_Validating, b = O_Returned, and c = W_Validate application|SUSPEND. This path was observed 14 times in the event log. Here, $\chi^2 \approx 13,40$ and $p \approx 0,000123$.

6 Conclusion

In this work, we aimed to explore the interplay between high-level problems in the process and the process instances which underlie them. These problems were related to observations of high loads, busy resources, batching behavior and delays in particular locations in the process throughout different points in time. We conceptualized each single outlier observation as a high-level event and we connected these high-level events into episodes whenever they were close enough in time, space, and when the cases giving rise to them were similar. For a given sequence of (outlier) observations, we wanted to investigate whether the cases that participate in that high-level behavior differ significantly from the cases which do not. For the comparison to be as meaningful as possible, the control group contained only the cases which were similar to the participating cases from a control-flow perspective. Our experiments showed that for the loan application process, there were several examples of high-level behavior at particular segments which were negatively associated with the case outcome (loan application success rate) and throughput time.

While the same method can be applied w.r.t. any process property at the case-level, the discovered significance in the connection between the emerging process behavior and the process instances underlying it is bidirectional. In future work, one could explore the cause-effect relationship behind these correlations. For *intrinsic* case properties (e.g., credit score of applicant), one could argue that it is the property itself which triggers certain kinds of high-level behavior. For *extrinsic* case properties (such as throughput time, or assigned resource for a specific task), a cause-effect discussion needs to consider the time when that property's value was set and when the high-level behavior emerged. Moreover, we cannot exclude the presence of confounding variables, that is, process aspects which influence both case participation and the case characteristic considered.

A further improvement to the method could be in the automatic segment selection. The experiments showed that some particular activities are run subsequently in an automatic way, so that reasoning about e.g., delays or work handover for these activity pairs makes less sense. Moreover, one could extend the method with a way of evaluating and ordering the detected high-level behavior by how surprising or interesting it is for the process at hand. Lastly, this method could be integrated in an interactive tool where the user selects the case property as well as high-level feature types and as a result, a list of most significant and interesting high-level behaviors w.r.t. that property is shown.

References

1. van der Aalst, W.M.P.: Process Mining: Data science in Action. Tech. rep. (2014)
2. Bakullari, B., van der Aalst, W.M.P.: High-level event mining: A framework. In: 2022 4th International Conference on Process Mining (ICPM) (2022)
3. Bozorgi, Z.D., Teinemaa, I., Dumas, M., La Rosa, M., Polyvyanyy, A.: Process mining meets causal machine learning: discovering causal rules from event logs. In: 2020 2nd International Conference on Process Mining (ICPM) (2020)
4. 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) (2021)
5. Denisov, V., Belkina, E., Fahland, D., van der Aalst, W.M.P.: The performance spectrum miner: visual analytics for fine-grained performance analysis of processes. In: International Conference on Business Process Management (BPM) (2018)
6. Denisov, V., Fahland, D., van der Aalst, W.M.P.: Unbiased, fine-grained description of processes performance from event data. In: International Conference on Business Process Management (BPM) (2018)
7. Dubinsky, Y., Soffer, P., Hadar, I.: Detecting cross-case associations in an event log: toward a pattern-based detection. Softw. Syst. Model (2023). https://doi.org/10.1007/s10270-023-01100-w
8. Klijn, E.L., Fahland, D.: Performance mining for batch processing using the performance spectrum. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) BPM 2019. LNBIP, vol. 362, pp. 172–185. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37453-2_15
9. Klijn, E.L., Fahland, D.: Identifying and reducing errors in remaining time prediction due to inter-case dynamics. In: 2020 2nd International Conference on Process Mining (ICPM) (2020)
10. Martin, N., Pufahl, L., Mannhardt, F.: Detection of batch activities from event logs. Inf. Syst. **95**, 77–92 (2021)
11. Pika, A., Ouyang, C., ter Hofstede, A.: Configurable batch-processing discovery from event logs. ACM Trans. Manag. Inf. Syst. **13**, 28 (2022)
12. Rodrigues, A.M.B., et al.: Stairway to value : mining a loan application process (2017)
13. Senderovich, A., Beck, J., Gal, A., Weidlich, M.: Congestion graphs for automated time predictions. Proc. AAAI Conf. Artif. Intell. **33**, 4854–4861 (2019)
14. Senderovich, A., Francescomarino, C.D., Maggi, F.M.: From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring. Inf. Syst. **84**, 255–264 (2019)
15. Senderovich, A., Weidlich, M., Gal, A., Mandelbaum, A.: Queue mining for delay prediction in multi-class service processes. Inf. Syst. **53**, 278–295 (2015)
16. Suriadi, S., Wynn, M., Xu, J., van der Aalst, W., ter Hofstede, A.: Discovering work Prioritisation patterns from event logs. Decis. Support Syst. **100**, 77–92 (2017)

17. Toosinezhad, Z., Fahland, D., Köroglu, Ö., van der Aalst, W.M.P.: Detecting system-level behavior leading to dynamic bottlenecks. In: 2020 2nd International Conference on Process Mining (ICPM) (2020)
18. van Hulzen, G.A., Li, C.Y., Martin, N., van Zelst, S.J., Depaire, B.: Mining context-aware resource profiles in the presence of multitasking. Artif. Intell. Med. **134**, 102434 (2022)
19. Wimbauer, A., Richter, F., Seidl, T.: PErrCas: process error cascade mining in trace streams. In: Munoz-Gama, J., Lu, X. (eds.) ICPM 2021. LNBIP, vol. 433, pp. 224–236. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98581-3_17

Adding the Sustainability Dimension in Process Mining Discovery Algorithms Evaluation

Andrea Delgado[1(✉)], Félix García[2], Ma Ángeles Moraga[2], Daniel Calegari[1], Alberto Gordillo[2], and Leonel Peña[1]

[1] Instituto de Computación, Facultad de Ingeniería, Universidad de la República, 11300 Montevideo, Uruguay
{adelgado, dcalegar, lpena}@fing.edu.uy

[2] Alarcos Research Group, Escuela Superior de Informática, University of Castilla - La Mancha, 13071 Ciudad Real, Spain
{Felix.Garcia,MariaAngeles.Moraga,Alberto.Gordillo}@uclm.es

Abstract. Sustainability has captured the attention of the classical management of business processes. Organizations have become increasingly aware of the need to achieve information technology (IT)-enabled business processes that are successful in their economy and ecological and social impact. In this context, Green BPM concerns business processes' modeling, deployment, optimization, and management with dedicated consideration for environmental consequences. Automated process discovery is a crucial process mining task to help organizations to get knowledge of the process they carry out in their daily operation, providing the basis for insights and evidence-based improvement decisions. Several process discovery algorithms have been developed and evaluated by the classical measures on resulting models, such as fitness, precision, f-score, soundness, complexity (size, structuredness, and control-flow complexity), generalization, and the execution time of the algorithm. Within the context of automated process discovery, sustainability adds a new indicator: energy efficiency. This paper extends a well-known benchmark for evaluating automated process discovery methods, measuring the energy efficiency of selected discovery methods with the same publicly available dataset. The expected contribution is to raise more awareness among the developers of process discovery methods about the energy impact of their solutions beyond the more traditional well-known measures.

Keywords: Sustainability · Green BPM · process mining · discovery algorithms · energy efficiency

Partially supported by project "Minería de procesos y datos para la mejora de procesos colaborativos aplicada a e-Government" funded by Agencia Nacional de Investigación e Innovación (ANII), Fondo María Viñas (FMV) "Proyecto ANII N° FMV_1_2021_1_167483", Uruguay, and by projects OASSIS (PID2021-122554OB-C31/ AEI/10.13039/ 501100011033/FEDER, UE) and EMMA (Project SBPLY/21/180501/000115, funded by CECD (JCCM) and FEDER funds), Spain.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 163–177, 2023.
https://doi.org/10.1007/978-3-031-41623-1_10

1 Introduction

Sustainability, in its broadest sense, is the ability to meet the needs of the present without compromising the ability of future generations to satisfy their own needs [31]. Organizations have become aware of the need to achieve information technology (IT)-enabled business processes that are successful concerning their economy and ecological and social impact. The challenge is, therefore, about how sustainability performance indicators can be considered in managing an organization's processes to warrant the establishment of "the Sustainable Enterprise" [9]. Green BPM concerns business process management with dedicated consideration for environmental consequences [13], adding a sustainability dimension to the classical dimensions: cost, quality, time, and flexibility.

A powerful tool in BPM for the analysis and optimization of existing business processes is automated process discovery, an essential process mining task [2], from where the understanding of the processes of an organization begins. Process discovery takes as input an event log, usually built from data extracted from the actual execution of support systems. It outputs a business process model capturing the behavior seen in the log. This behavior is mainly focused on the control-flow perspective of the business process. Still, it can be extended with other perspectives: the organizational or resource perspective, the data perspective, and the time perspective.

Due to the heterogeneous nature of process discovery methods, its comparative evaluation provides evidence of its usefulness in different contexts, according to its performance and the quality of the resulting business process model, using classical measures such as fitness, precision, f-score, soundness, complexity (size, structuredness, and control-flow complexity) and generalization. In [7], the authors define an open-source benchmark framework for process discovery methods and use it for comparing seven implementations of representative methods: α\$, Inductive Miner (IM), Evolutionary Tree Miner (ETM), Fodina (FO), Structured Heuristic Miner 6.0 (S-HM6), Split Miner (SM), and Hybrid ILP Miner (HILP). The authors used two heterogeneous datasets: a collection of real-life event logs publicly available at the 4TU Centre for Research data and proprietary logs from several companies worldwide. The comparison covers nine quality metrics focused on performance and quality, as mentioned before.

Within the context of automated process discovery, sustainability adds a new indicator: energy efficiency, which should also be considered a first-class citizen to support the decision-making process in process discovery problems. To analyze the potential impact that process discovery methods can have on energy consumption, this paper extends the work in [7], evaluating the energy consumption of selected discovery methods using publicly available datasets. We included the automated process discovery methods IM, ETM, FO, S-HM6, SM, and HILP with default parameters based on the original benchmark. These methods were applied to all the public logs from the Business Process Intelligence Challenge (BPIC) (2012–2015, 2017) and the RTFMP and SEPSIS logs. The empirical study was conducted using the FEETINGS framework [22] and a hardware measuring device to obtain realistic consumption values. The expected contribution is to raise more awareness among the developers of process discovery methods about the energy impact of their solutions beyond the more traditional well-known measures.

The rest of the paper is structured as follows. In Sect. 2, we introduce the background and discuss related work on sustainability. In Sect. 3, we present the evaluation we carried out, including the process efficiency measures and results obtained. In Sect. 4, we present the discussion of results and findings. Finally, in Sect. 5, we provide conclusions and an outline of future work.

2 Background and Related Work

2.1 Green BPM

For organizations dealing with BPM, the orientation toward more sustainable processes that use resources more efficiently is considered Green BPM. The systematic literature review in [13] concluded that Green BPM research has mainly focused on the capability areas for process modeling, deployment, optimization, and management, following a similar evolution as the BPM discipline. Early BPM research focused on the traditional business process lifecycle, with the modeling capability area as the first citizen (i.e., extending modeling notations, adapting modeling notations, and adding patterns), with much attention paid to frameworks consisting of formulas and standards to measure and control emissions or energy in business processes. To facilitate successful Green BPM, organizational capability areas (i.e., culture, structure) should also be considered as important [13]. In [15], the authors highlight the contribution of ICTs to business processes to reduce CO2 emissions and provide them with applications and services to manage their energy resources efficiently and control the environmental impact of their process activities. In our previous work, a BPMS-Game is proposed [24], gamification is applied to business process management systems (BPMS) to improve their sustainability, engaging users to be more environmentally friendly in their daily work.

Green BPM has therefore been focused on the environmental dimension, which is also the focus of this work. More specifically, we are interested in how supporting BPM technologies must be evaluated from an environmental perspective to support a more Green-IN-friendly business process, which means the business process is green. It contrasts with the other perspective (Green-By), which tackles how the business process can generate value from a green perspective. More specifically, we tackle the undeniable influence that supporting software can have on energy consumption, a growing research field in the last few years [12]. BPM-supporting software can have a significant energy consumption impact, especially considering those which could require a large amount of energy to operate, e.g., those based on artificial intelligence algorithms [30]. Process mining evaluation from an energy perspective is a representative example of this, as tackled in the following subsection.

2.2 Green Process Mining

Most related work about process mining and sustainability can be categorized under the "BY" perspective. In [4], the authors propose a framework based on process mining techniques and a genetic algorithm to promote the pursuit of circular manufacturing strategies by identifying the disassembly sequence, which is more energy-effective. A framework based on machine learning and process mining techniques is proposed in [14] to automatically generate simulation models for smart factories. An agile operations management (AOM) system is the target in [18] to respond to problems for green factories. In [19], a data-driven methodology that exploits data integration, process mining, and analytics to identify bottlenecks is proposed.

From the "IN" perspective, some design principles that data scientists can apply to reduce the footprint of data mining are identified in [29]. Five principles are proposed based on reducing and reusing operations, data, models, and sharing data, models, and skills. Empirical evaluation of some of these principles was also conducted using a software tool that estimates energy consumption based on the usage of computational resources. We aim to raise awareness about this crucial field and advocate using hardware measuring instruments to support decisions with accurate consumption data. In [21], nine process mining critical success factors are also identified, which could serve as a reference to conduct empirical studies to evaluate their energy impact.

The notion of "Green Data Science" (GDS) is introduced in [1] due to the possible "pollution" which can be caused by data science. In this context, data can be analyzed from four angles: fairness, confidentiality, accuracy, and transparency. As an example, "green challenges" in process mining are discussed by considering processes and event data, such as: taking into account fairness to avoid undesirable forms of discrimination; anonymization and de-identification of the event data (confidentiality); to address quality problems of the data to improve their accuracy; to support the provenance of event data (transparency).

As stated in [26], software programmers have little experience with software energy consumption. Similarly, developers of state-of-the-art process mining methods do not consider energy efficiency, as the focus is on the traditional quality metrics (accuracy, generalization, etc.). Green metrics should also be considered in the evaluation of these methods. Therefore, it is necessary to continue working in the "IN" perspective identifying aspects or factors that can impact the "greenability" of process mining techniques with supporting evidence.

2.3 Automated Process Discovery Benchmark

In [7], the authors defined an open-source benchmark for the comparative evaluation of automated process discovery methods. The benchmark defines unified settings using public datasets and evaluation metrics, allowing a reproducible and consistent empirical comparison of methods.

The benchmark defined nine accuracy and complexity metrics for the evaluation. Within the accuracy metrics, the authors measure recall (a.k.a. **fitness**), i.e., the degree to which every trace in the log can be aligned (with a small number of errors) with a trace produced by the model; **precision** which measures the ability of a model to generate only the behavior found in the log; and they combine them into a single measure known as **F-score**, which is the harmonic mean of the two measurements. The authors also measure **generalization**, which refers to the ability of models to generate traces that are not present in the log, but the business process can produce that under observation. They used k-fold cross-validation for measuring generalization with a value of k = 3 for performance reasons. For measuring complexity, i.e., how difficult it is to understand a model, the authors used three measures: **size**, i.e., number of nodes; Control-Flow Complexity (**CFC**), i.e., the amount of branching caused by gateways in the model; and **structuredness**, i.e., the percentage of nodes located directly inside a block-structured single-entry single-exit fragment. The authors also assess **soundness** reporting whether the model violates one of the three soundness criteria: option to complete, proper completion, and no dead transitions. Finally, the evaluation also considers the **execution time**.

The evaluation is performed using two datasets: a collection of real-life event logs publicly available at the 4TU Centre for Research Data, conformed by logs from different domains and complexities, including BPI Challenge (BPIC) logs, the Road Traffic Fines Management Process (RTFMP) log, and the SEPSIS Cases log; and a second dataset composed of twelve proprietary and heterogeneous logs from several organizations. This last dataset is not available for future uses of the benchmark.

The authors selected seven implementations of representative methods to evaluate, which are summarized in Table 1:

1. $\alpha\$$ [16]. An extension of the well-known α algorithm [3], which can discover invisible tasks involved in non-free-choice constructs.
2. Inductive Miner (IM) [20]. It is focused on providing a behaviorally correct (sound) process tree by repeatedly splitting the event log into sub-logs.
3. Evolutionary Tree Miner (ETM) [11]. It is focused on providing a process tree applying genetic algorithms based on preferences concerning fitness, precision, generalization, and complexity.
4. Fodina (FO) [10]. It is based on the well-known Heuristics Miner [32] (an improvement of the α miner) but more robust to noisy data.
5. Structured Heuristic Miner 6.0 (S-HM$_6$) [6]. It is also an improvement of the Heuristics Miner algorithm to separate the objective of producing accurate models and ensuring their structuredness and soundness.
6. Split Miner (SM) [5]. It is focused on producing simple process models with low branching complexity and consistently high and balanced fitness, precision, and generalization.
7. Hybrid ILP Miner (HILP) [34]. It is an improvement of the Integer Linear Programming (ILP) miner [33] based on varying the number of variables used for solving the ILP problem.

As a general result, the authors concluded that no method outperforms all others across all metrics. IM, ETM, and SM showed to be the most effective

Table 1. Summary of selected discovery methods (from [7])

Method	Model Language	Semantic Constructs
α\$ [16]	Petri Nets	AND/XOR/Loop
Inductive Miner [20]	Process Trees	AND/XOR/OR/Loop
Evolutionary Tree Miner [11]	Process Trees	AND/XOR/OR/Loop
Fodina [10]	BPMN	AND/XOR/Loop
Structured Heuristic Miner 6.0 [6]	BPMN	AND/XOR/Loop
Split Miner [5]	BPMN	AND/XOR/Loop
Hybrid ILP Miner [34]	Petri Nets	AND/XOR/Loop

methods in terms of the simplicity of the resulting models and values of fitness, precision, and F-score. Nevertheless, there is a common weakness, which is their inability to handle large-scale real-life logs. For the execution time, SM outperformed all the other methods: it was the fastest discovery method 23 times out of 24. In contrast, ETM was the slowest method, reaching a timeout of four hours for 22 logs. The authors also performed a second evaluation to analyze how each discovery method could improve its output using hyperparameter optimization. In this case, results showed that FO and S-HM_6 could also perform very well, though at the expense of long execution times (up to 24 h for some logs) and powerful computational resources. Due to the extremely-long execution times, it was prohibitive to hyper-parameter optimize the α\$ and ETM methods.

3 Evaluation

In this section, we present the evaluation we have carried out regarding the energy efficiency of the PM algorithms. We describe the setup, methodology, and benchmark extension with corresponding evaluation metrics and results.

3.1 SetUp and Methodology

The PM algorithms' energy efficiency was evaluated using FEETINGS (Framework for Energy Efficiency Testing to Improve eNvironmental Goals of the Software) [22]. FEETINGS provides conceptual, methodological, and technological components to rigorously measure the energy consumption of running software.

From a methodological perspective, we have followed GSMP (Green Software Measurement Process) [23], which is available as an electronic process guide[1]. GSMP is a specialized approach for measuring software energy consumption to ensure accurate capture, analysis, and interpretation of software energy consumption data. GSMP involves seven phases, starting from the definition of

[1] https://alarcos.esi.uclm.es/feetings/.

the requirements and software entity to be evaluated, followed by the measurement environment configuration and preparation, the collection of the energy consumption data, and the data analysis and reporting.

From the technical side (see Fig. 1), a hardware measuring instrument was used to obtain more accurate and realistic energy consumption data, as compared to software estimators, which is named EET (Energy Efficiency Tester). EET captures consumption data through its connections to the PC's power supply (DUT: Device Under Test) with a sampling frequency of 100 Hz. Besides obtaining the PC's total energy consumption, EET can also capture the energy consumption of different components (processor, graphic card, hard drive disk, and monitor). The environment is completed with the ELLIOT software tool to visualize and analyze the energy consumption results captured by EET.

Fig. 1. FEETINGS technological environment

The characteristics of the DUT which was used to conduct this study are:

- Monitor: Falkon Q2702S 27" 2K
- Motherboard: Asus Prime B460-Plus.
- Processor: i7 10700 2,9 GHz
- RAM: 4 modules of 32 GB DDR4 Kingston 2666MHz CL16
- Graphics card: Zotac Gaming GeForce RTX 3060 12 GB GDDR6
- Hard Disk: Kingston SSD A400 - 480GB SATA
- Power supply Energy: PS901SX 900W
- OS: Windows 11 Pro

3.2 Benchmark Extension

Concerning the test cases specification, we used the benchmark code provided by the authors of [7]. Since we want to measure the energy consumption of the method precisely, the code was slightly modified to include the calls to automatically start and stop the energy measuring device (EET). It allowed us to circumscribe the measurement to the algorithm execution and leave outside the rest: the benchmark measures preparation and calculation, the preparation and saving of the result, etc.

As presented in Sect. 2, the evaluation metrics used in the benchmark to evaluate the process discovery methods were the classical measures of resulting models, such as fitness, precision, f-score, soundness, complexity, generalization, and execution time. Since the authors in [7] already provide definitions, results, and discussion regarding these measures, we will not dive into them here. Following the main objective of this work, we extended the benchmark evaluation metrics with the *HDD*, *Processor*, and *DUT* total energy consumption to evaluate the energy efficiency perspective. As the benchmark can be invoked to measure all or selected measures over several or selected event logs, we invoked the benchmark using only the execution time measure for the algorithm, ten times for each algorithm, providing the average execution time. We also include the execution time presented in the paper for each algorithm for reference.

We evaluated six of the seven representative methods of the original benchmark. The algorithm a$ was discarded in our study, as it showed scalability issues in the original benchmark with several logs. It was also confirmed in our initial evaluation to select the algorithms to be evaluated. We evaluated the methods with default parameters. We did not include the hyper-parameter evaluation since that analysis focuses on evaluating possible improvements for the classical measures on resulting models, so we left it for future work. Regarding the logs used as input, we included all the public logs from the Business Process Intelligence Challenge (BPIC) from 2012 to 2015 and 2017 and the RTFMP and SEPSIS logs. Table 2 presents the descriptive statistics of such logs from [7].

Table 2. Descriptive statistics of public logs from [7]

Log Name	Total traces	Dist traces (%)	Total events	Dist events	Tr.length Min	Tr.length Avg	Tr.length Max
BPIC12	13,087	33.4	262,200	36	3	20	175
BPIC13cp	1,487	12.3	6,600	7	1	4	35
BPIC13inc	7,554	20.0	65,533	13	1	9	123
BPIC14f	41,353	36.1	369,485	9	3	9	167
BPIC151f	902	32.7	21,656	70	5	24	50
BPIC152f	681	61.7	24,678	82	4	36	63
BPIC153f	1,369	60.3	43,786	62	4	32	54
BPIC154f	860	52.4	29,403	65	5	34	54
BPIC155f	975	45.7	30,030	74	4	31	61
BPIC17f	21,861	40.1	714,198	41	11	33	113
RTFMP	150,370	0.2	561,470	11	2	4	20
SEPSIS	1,050	80.6	15,214	16	3	14	185

In summary, 72 test cases were defined for the evaluation, corresponding to executing the algorithms (S-HM, SM, IM, FO, ETM, and HILP) on the 12 logs. Each test case was executed ten times to get more reliable energy consumption data from the measuring device (EET), and the average value was used. This number of repetitions is chosen because it is considered enough according to the characteristics of this study and of the algorithms evaluated, whose execution times are much longer than other typical energy efficiency evaluation studies. In addition, it is essential to highlight that a hardware device has been used that obtains 100 samples (instant power values) per second, producing; as a result, a great number of values per test case.

3.3 Results

In Table 3, we present the results of the execution, i.e., the energy consumption results for each log and method, including the HDD, Processor, and DUT consumption values and the execution time. As mentioned, we include the benchmark execution time as a reference. However, we do not include the rest of the measures since they focus on the resulting model generated by the method. Our execution results also confirm the general benchmark results: the SM is the fastest algorithm for all logs. Conversely, ETM presents longer execution times in half of the logs, and HILP presents the longest time of all executions and time out in two logs. In Fig. 2 we present the energy consumption by method for each log. The DUT consumption by log values for all logs is shown in Fig. 3.

Regarding the consumption values, in most cases, the methods with higher execution times also had higher consumption, usually due to the high correlation between energy consumption and execution time. However, there are specific cases in which the consumption was higher than others but with fewer execution times, namely FO for BPIC155f, BPIC152f, BPIC151f, IM for BPIC152f, BPIC153f, S-HM6 for BPIC151f, SEPSIS, SM for BPIC151f, BPIC12, ETM for BPIC151f, BPIC125f, BPIC17 and HILP for SEPSIS. The log BPIC151f requires, in all cases except in FO and HILP methods, less time but higher consumption, which denotes that the characteristics of the log make it perform worse from an energy efficiency point of view.

In this regard, it is essential to note that time is a very influential variable in the total energy consumption equation (Energy = Power * Time). However, as can also be seen in our results, it cannot be used as the only consumption indicator. In fact, a common misconception is that simply reducing runtime will reduce energy consumption [28] [25]. An example of this, as indicated in [27], is that such a reduction in time may lead to an increase in CPU cycles which could increase the equation's power (P) value.

If we specifically analyze the energy performance of the methods by logs, the results show that BPIC17 is the log with the worst results for the methods ETM, FO, IM, and SM, being BPIC12 the worst log case for S-HM and BPIC153f for HILP. On the contrary, BPIC13CP is the log with the best consumption results for all the methods. Furthermore, ETM has the higher consumption results for

Table 3. Default parameters evaluation results for the BPIC logs extending [7]

Log	Discovery Method	Consumption (J)			Exec. Time	Exec. Time p.
		HDD	Proc.	DUT		
BPIC12	S-HM6	2032,50	6664,83	43853,27	365,2315	227,80
	SM	4,56	14,45	42,61	0,5793	0,58
	IM	37,18	110,56	730,12	4,5613	6,60
	FO	45,75	141,85	820,23	7,1124	9,66
	ETM	5854,18	103454,20	194147,95	3601,94	14400,00
	HILP	9148,19	23124,06	112696,88	958,72	772,20
BPIC13cp	S-HM6	1,13	13,57	43,41	0,37	130,00
	SM	0,22	1,25	2,36	0,07	0,03
	IM	1,24	8,13	43,10	0,41	0,10
	FO	1,08	6,65	33,13	0,35	0,06
	ETM	548,43	1634,75	6183,93	45,14	14400,00
	HILP	2,32	6,70	18,02	0,25	0,10
BPIC13inc	S-HM6	3,73	27,64	114,11	1,11	0,80
	SM	1,00	3,84	12,00	0,23	0,23
	IM	5,77	24,37	138,06	1,20	1,00
	FO	6,43	28,67	149,28	1,35	1,41
	ETM	4503,39	11155,79	55222,26	519,98	14400,00
	HILP	7,92	34,04	94,71	0,83	2,50
BPIC14f	S-HM6	82,29	356,80	1749,58	17,16	147,40
	SM	3,06	13,75	36,74	0,64	0,59
	IM	17,22	65,59	465,58	3,57	3,40
	FO	90,22	376,52	1484,50	18,79	27,73
	ETM	16121,78	63220,56	220415,74	3658,23	14400,00
	HILP	374,92	1226,27	5329,13	37,94	7,30
BPIC151f	S-HM6	699,75	4985,36	20983,52	148,05	128,10
	SM	1,66	15,76	30,73	0,34	0,48
	IM	5,06	45,84	163,55	1,04	0,60
	FO	3,74	31,25	100,84	0,78	1,02
	ETM	406,77	4275,43	12287,66	81,24	14400,00
	HILP	2179,33	4421,29	25171,22	168,43	4,40
BPIC152f	S-HM6	1179,93	6398,74	30199,53	234,71	163,20
	SM	1,78	13,37	29,49	0,35	0,25
	IM	7,63	77,18	264,47	1,48	0,70
	FO	5,48	46,39	162,01	1,06	0,61
	ETM	543,22	4908,19	14587,60	106,29	14400,00
	HILP	–	–	–	–	–
BPIC153f	S-HM6	778,63	4422,73	20465,30	147,54	139,90
	SM	2,01	11,08	32,56	0,38	0,36
	IM	17,07	130,63	555,92	3,19	1,30
	FO	6,35	44,90	156,16	1,19	0,89
	ETM	957,25	9485,02	28776,58	185,95	14400,00
	HILP	28160,63	636963,75	1616191,01	10230,77	1062,90
BPIC154f	S-HM6	865,11	4849,50	21969,83	157,78	136,90
	SM	1,88	12,01	25,68	0,34	0,25
	IM	7,85	59,34	210,56	1,39	0,70
	FO	5,19	33,48	122,66	0,92	0,50
	ETM	527,27	5497,04	16048,24	105,85	14400,00
	HILP	2523,08	18390,39	66830,13	396,91	14,70
BPIC155f	S-HM6	871,58	4373,47	20644,36	150,00	141,90
	SM	2,23	13,97	32,07	0,37	0,27
	IM	9,36	65,85	256,92	1,51	1,50
	FO	6,32	41,44	149,47	1,02	0,56
	ETM	490,64	5012,13	14971,74	96,65	14400,00
	HILP	–	–	–	–	–
BPIC17f	S-HM6	57,72	305,48	1165,02	9,73	143,20
	SM	10,32	40,01	159,58	2,04	2,53
	IM	74,51	306,55	1401,85	10,53	13,30
	FO	309,22	898,89	2712,05	28,92	64,33
	ETM	17772,54	65937,14	242578,92	3619,62	14400,00
	HILP	2030,01	23124,90	74161,70	458,45	384,50
RTFMP	S-HM6	1140,77	4832,90	30923,02	248,80	262,70
	SM	4,15	10,70	72,27	0,95	1,25
	IM	36,96	192,85	1170,91	8,18	10,90
	FO	16,33	71,93	443,50	3,58	2,57
	ETM	4867,04	46715,78	142435,03	958,86	14400,00
	HILP	19,46	120,04	461,99	4,38	3,50
SEPSIS	S-HM6	1152,79	7545,75	37204,38	246,58	242,70
	SM	0,54	1,42	6,25	0,13	0,05
	IM	2,58	20,14	83,44	0,57	0,40
	FO	2,10	15,23	65,92	0,48	0,17
	ETM	548,63	3594,88	11785,93	109,32	14400,00
	HILP	13,69	322,82	750,39	3,33	1,60

half the logs and HILP and S-HM6 for three logs each. SM is the best method for all logs concerning energy consumption.

It is also interesting how the consumption results can increase depending on the applied method on a specific log. Namely, a relative percentage variation of the energy consumption from the best to the worst method is observed, as shown in Table 4. These results also illustrate the high variation between the performance of the methods depending on the log. The correlation between log size (events, traces), distribution of traces and events (DT(%), DE), time, and DUT consumption variables were also analyzed. Given the non-normal distri-

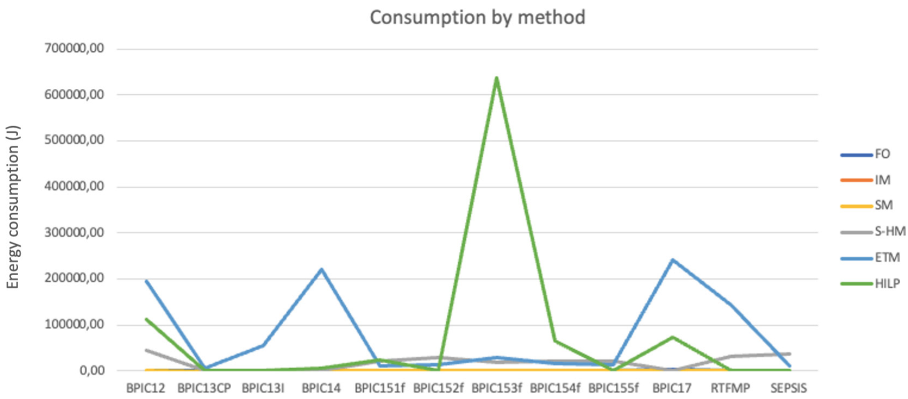


Fig. 2. Consumption by method

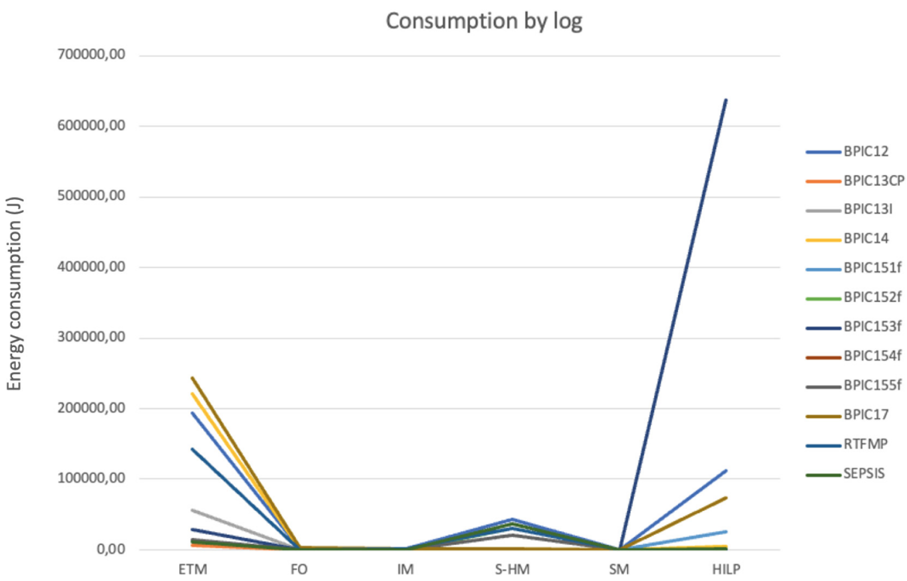


Fig. 3. Consumption by log

bution of the values of these variables, the Spearman correlation coefficient was used, concluding that there is a significant correlation between the execution time of this study and the execution time of the benchmark and also between consumption and the number of events, but not in the number of traces.

4 Discussion

Considering the results from the energy efficiency perspective, the ETM and HILP methods present the worst results, followed by the S-HM6. Conversely, the SM method has the best results for all logs. Therefore, if the target is focused only on the consumption perspective, practitioners should avoid the first methods and consider the most environmentally-friendly option.

Table 4. Best and worst consumption by log

Log	Rel. Inc. (%)	Worst cons. (J)	Best cons. (J)
SEPSIS	595270,08	37204,38	6,25
BPIC12	55639,40	194147,95	42,61
BPIC152f	102406,00	30199,53	29,49
BPIC154f	260241,94	66830,13	25,68
BPIC151f	81910,90	25171,22	30,73
BPIC155f	64372,81	20644,36	32,07
BPIC153f	1956276,87	636963,75	32,56
RTFMP	197087,35	142435,03	72,27
BPIC14	599933.97	220415,74	36,74
BPIC13CP	262030,93	6183,93	2,36
BPIC17	152010,85	242578,92	159,58
BPIC13I	460185,50	55222,26	12,00

However, energy efficiency and other process mining evaluation measures should be considered, as illustrated in this study. The number of events in the log is one of the characteristics which can affect the energy consumption of the method. Other factors, such as the number and percentage of distinct traces, also deserve further investigation, as discussed in [8]. The synthetic or real nature of the logs can be another determining factor, given that, as shown in the original benchmark, IM, ETM, and SM, which performed very well in accuracy evaluation, can fail when challenged with large-scale unfiltered real-life events logs. As pointed out in [21], among the success factors of process mining, data and event log quality could most affect energy consumption. Consequently, specific empirical studies to challenge existing methods with real logs and evaluate the log quality could enrich decision-making when selecting methods.

On the other hand, the accuracy results of the discovered models are vital indicators for selecting the most appropriate method. Considering energy consumption as an additional key indicator beyond execution time, a suitable trade-off must be made. Therefore, it should be analyzed whether the differences between the methods used concerning their accuracy are worthwhile from an energy point of view. The original benchmark results [7] evidenced significant differences in the discovered models, being the best-performing methods in accuracy IM, ETM, and SM, along with consistently performing very well in fitness (IM), precision (ETM, SM), F-score (SM), complexity (IM, ETM, SM) and execution time (SM). However, the issue that can arise in this context is, is a 0.01 improvement in fitness worth, for example, a 10 % increase in energy consumption in return? As shown in [17], in which the energy impact of changing the optimization method of an ML model based on logistic regression is analyzed, an improvement in accuracy of 0.016 percent required double the energy to run.

Hyperparameter optimization is another factor that should not be ignored. As evidenced by the benchmark results, the performance of the methods could be improved, and energy consumption could also be evaluated in such a context.

Finally, to transfer knowledge, we must provide guidelines and good practices. For example, in [29], some principles of Green Data Mining that data scientists can apply to reduce this footprint. In this way, developers can make the best decisions, considering energy efficiency as an essential factor.

5 Conclusions

This paper presents an extension of the automated process discovery benchmark [7] that includes measures for the energy efficiency perspective adding the sustainability dimension to the evaluation of process discovery algorithms. We used the framework FEETINGS from the methodological side and the hardware instrument tester EET, which allowed us to measure the consumption energy of the selected process discovery algorithms, presented in Sect. 3.

Neither energy efficiency nor other process mining evaluation measures should be considered in isolation, e.g., it could be necessary to evaluate whether an improvement in the accuracy of the discovery method is worthwhile from an energy point of view. Further studies are also needed to delve into the aspects that affect energy efficiency, such as the number of events in the log, the number and percentage of distinct traces, and the synthetic or real nature of the logs.

As more organizations are concerned about having sustainable development that helps take care of the environment and ecological effects, being aware of the energy consumption of algorithms, programs, and systems in general, but of automated process discovery in particular, is of utmost importance. We believe energy efficiency should also be considered a first-citizen measure to support decision-making in process discovery problems.

References

1. van der Aalst, W.M.P.: Green data science - using big data in an "environmentally friendly" manner. In: ICEIS - 18th International Conference on Enterprise Information Systems. vol. 1, pp. 9–21. SciTePress (2016)
2. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016). https://doi.org/10.1007/978-3-662-49851-4
3. van der Aalst, W., Weijters, A., Maruster, L.: Workflow mining: discovering process models from event logs. IEEE Trans. Knowl. Data Eng. **16**, 1128–1142 (2004)
4. Acerbi, F., Polenghi, A., Quadrini, W., Macchi, M., Taisch, M.: Fostering circular manufacturing through the integration of genetic algorithm and process mining. In: Kim, D.Y., von Cieminski, G., Romero, D. (eds) Advances in Production Management Systems. Smart Manufacturing and Logistics Systems: Turning Ideas into Action. APMS 2022. IFIP Advances in Information and Communication Technology. vol 664. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16411-8_47
5. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L.: Split miner: discovering accurate and simple business process models from event logs. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 1–10. IEEE Computer Society (2017)
6. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L., Bruno, G.: Automated discovery of structured process models from event logs: the discover-and-structure approach. Data Knowl. Eng. **117**, 373–392 (2018)
7. Augusto, A., et al.: Automated discovery of process models from event logs: review and benchmark. IEEE Trans. Knowl. Data Eng. **31**(4), 686–705 (2019)
8. 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)
9. vom Brocke, J., Seidel, S., Recker, J. (eds.): Green Business Process Management - Towards the Sustainable Enterprise. Springer (2012). https://doi.org/10.1007/978-3-642-27488-6
10. vanden Broucke, S.K.L.M., Weerdt, J.D.: Fodina: a robust and flexible heuristic process discovery technique. Decis. Support Syst. **100**, 109–118 (2017)
11. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Quality dimensions in process discovery: the importance of fitness, precision, generalization and simplicity. Int. J. Cooperative Inf. Syst. **23**(1), 1440001 (2014)
12. Calero, C., Moraga, M.Á., Piattini, M. (eds.): Software Sustainability. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-69970-3
13. Couckuyt, D., Looy, A.V.: A systematic review of green business process management. Bus. Process. Manag. J. **26**(2), 421–446 (2020)
14. Friederich, J., Francis, D.P., Lazarova-Molnar, S., Mohamed, N.: A framework for data-driven digital twins of smart manufacturing systems. Comput. Ind. **136**, 103586 (2022)
15. González, A.H., Calero, C., Parra, D.P., Mancebo, J.: Approaching green BPM characterisation. J. Softw. Evol. Process. **31**(2), e2145 (2019)
16. Guo, Q., Wen, L., Wang, J., Yan, Z., Yu, P.S.: Mining invisible tasks in non-free-choice constructs. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) BPM 2015. LNCS, vol. 9253, pp. 109–125. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_7

17. Gutiérrez, M., Moraga, M.Á., García, F.: Analysing the energy impact of different optimisations for machine learning models. In: 2022 International Conference on ICT for Sustainability (ICT4S), pp. 46–52 (2022)
18. Jo, H., Noh, S.D., Cho, Y.: An agile operations management system for green factory. Int. J. Precis. Eng. Manufact. -Green Technol. **1**(2), 131–143 (2014). https://doi.org/10.1007/s40684-014-0018-z
19. Kumbhar, M., Ng, A., Bandaru, S.: Bottleneck Detection Through Data Integration, Process Mining and Factory Physics-Based Analytics (04 2022)
20. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs - a constructive approach. In: Colom, J.-M., Desel, J. (eds.) PETRI NETS 2013. LNCS, vol. 7927, pp. 311–329. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38697-8_17
21. Mamudu, A., Bandara, W., Wynn, M.T., Leemans, S.J.J.: A process mining success factors model. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds) Business Process Management. BPM 2022. Lecture Notes in Computer Science. vol 13420. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16103-2_12
22. Mancebo, J., Calero, C., García, F., Moraga, M.Á., de Guzmán, I.G.R.: FEETINGS: framework for energy efficiency testing to improve environmental goal of the software. Sustain. Comput. Informatics Syst. **30**, 100558 (2021)
23. Mancebo, J., García, F., Calero, C.: A process for analysing the energy efficiency of software. Inf. Softw. Technol. **134**, 106560 (2021)
24. Mancebo, J., Garcia, F., Pedreira, O., Moraga, M.A.: BPMS-Game: tool for business process gamification. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNBIP, vol. 297, pp. 127–140. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65015-9_8
25. Manotas, I., et al.: An empirical study of practitioners' perspectives on green software engineering. In: 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), pp. 237–248 (2016)
26. Pang, C., Hindle, A., Adams, B., Hassan, A.E.: What do programmers know about software energy consumption? IEEE Softw. **33**(3), 83–89 (2016)
27. Pinto, G., Castor, F.: Energy efficiency: a new concern for application software developers. Commun. ACM **60**(12), 68–75 (2017)
28. Pinto, G., Castor, F., Liu, Y.D.: Mining questions about software energy consumption. In: Proceedings of the 11th Working Conference on Mining Software Repositories, pp. 22–31. MSR 2014, ACM (2014)
29. Schneider, J., Seidel, S., Basalla, M., vom Brocke, J.: Reuse, reduce, support: design principles for green data mining. Bus. Inf. Syst. Eng. **65**(1), 65–83 (2023)
30. Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O.: Green AI. Commun. ACM **63**(12), 54–63 (2020)
31. United Nations, U.: Report of the world commission on environment and development: our common future (brundtland report). Tech. rep. (1987)
32. Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering workflow models from event-based data using little thumb. Integr. Comput. Aided Eng. **10**(2), 151–162 (2003)
33. van der Werf, J.M.E.M., van Dongen, B.F., Hurkens, C.A.J., Serebrenik, A.: Process discovery using integer linear programming. Fundam. Informaticae **94**(3–4), 387–412 (2009)
34. van Zelst, S.J., van Dongen, B.F., van der Aalst, W.M.P.: ILP-based process discovery using hybrid regions. In: International Workshop on Algorithms & Theories for the Analysis of Event Data, ATAED. vol. 1371, pp. 47–61. CEUR-WS.org (2015)

Steady State Estimation for Business Process Simulations

Mahsa Pourbafrani[1(✉)], Niels Lücking[2], Matthieu Lucke[2], and Wil M.P.van der Aalst[1,2]

[1] Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany
{mahsa.bafrani,wvdaalst}@pads.rwth-aachen.de
[2] Celonis, Munich, Germany
{niels.lucking,m.lucke}@celonis.com

Abstract. Simulation is a powerful tool to explore and analyze business processes and their potential improvements. Recorded event data allow for the generation of data-driven simulation models using process mining. The accuracy of existing approaches, however, remains a challenge. Various efforts are being made to improve the quality of the used data and techniques, such as extracting detailed resource performance. One of the least addressed challenges is the initial state of the simulation run. Starting from a steady state has been considered in simulation in other fields. In current process simulation approaches, the executions mostly start from an empty state. This assumption leads to initialization bias, or the startup problem, which has an impact on the early results and limits the types of analysis that can be performed. In this paper, we propose an approach to estimate a steady state of simulation models, which enables the generation of more realistic simulation results. The evaluation using real-world and synthetic event data shows the requirements for and advantages of starting from representative steady states in process simulations.

Keywords: process mining · data-driven simulation · event logs · steady-state simulation · discrete event simulation

1 Introduction

Process mining analyzes event data to provide insights into the processes, such as running process models and their conformance and performance behavior. Simulation models are used to generate future results and outcomes for processes. The insights provided by process mining can be used to simulate how the processes will continue and what the impact of the changes will be [1]. The captured events including the process instances, performed actions, and their

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-EXC-2023 Internet of Production - 390621612. We also thank the Alexander von Humboldt (AvH) Stiftung for supporting our research. We thank Celonis for supporting the project.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 178–195, 2023.
https://doi.org/10.1007/978-3-031-41623-1_11

timestamps, in event data, are the starting point for forming data-driven simulation models. Simulation models should be able to reproduce the same outputs as the input event data. However, due to various challenges such as data quality, models cannot fully capture the complexity of the real world, e.g., external factors and resource behaviors.

One of the open challenges is the starting point of process simulation models. Current data-driven process simulation approaches mostly start from an empty state. As a result, if they manage to reach a steady state, it takes time, i.e., a warm-up period is required while simulating. The assumption of an empty state at the start of the simulation can prevent accurate results from being obtained. In particular, simulation may never reach a steady state in the case of an unstable process with the incorrect start point. When analyzing event data of processes, one typically considers only complete cases. If the period in which event data are collected is of the same order of magnitude as the average flow time (e.g., months), then this creates a misleadingly low load. However, it is also difficult to use incomplete cases. Therefore, it is important to estimate a representative *steady state* and start simulating from this point. Short-term simulation and the accuracy of simulation results share common aspects, such as loading a representative initial state and starting simulation from a non-trivial initial state [14].

In this paper, we propose an approach to estimate steady states of processes using their event logs, which can be loaded into the simulation as the starting point. The simulation engine is considered a black box, since the focus is on the illustration of the effect and necessity of having steady states in process simulation models. We follow three main objectives while designing the approach: (1) practical components of the approach should be general, not specific to any simulation engine but applicable to common data-driven simulation approaches, (2) the time patterns in event data should be captured while discovering steady states, as there might not be a real steady state, e.g., concept drift, and (3) practical component of the approach must be efficient since simulation results might ultimately converge toward the same state as one that starts from a cold start. We evaluated our approach in estimating steady states of processes for simulation purposes using both real and synthetic logs.

Throughout the paper, a *state* of a process is an event log at a specific point in time, including the events that started before but were not finished by that time. A *start state* of a simulation is the state from which the execution will start. A *steady state* is a state of a process that shows the common behavior of the process w.r.t. different aspects, e.g., the number of cases in the process. *warm-up* period refers to the period of time that a simulation started from a cold start needs to reach a steady state of the process. The start state of a simulation can be an empty state (also referred to as a *cold start*) or a given (discovered using event data) state.

The remainder of the paper is as follows. The related work is introduced in Sect. 2. We elaborate on the motivation using the running example in Sect. 3. Preliminaries are introduced in Sect. 4. We explain the approach in Sect. 5 and evaluate it in Sect. 6. Section 7 concludes this work.

2 Related Work

Simulation of operational processes used to be a complex task for an expert due to the design of the process model, estimating the parameters, and providing simulation configurations [15]. The increasing availability of data and advances in process mining have led to semi-automatic approaches, where parts of the simulation model and parameters are mined and others are based on the user's input [1], e.g., [13]. This is the current setting in most academic and commercial tools, e.g., Celonis[1].

Most research efforts are currently focused on improving simulation quality, which still struggles to represent reality enough to be widely applicable. There are interactive approaches, such as [10], that propose using multiple metrics, such as stochastic conformance checking, to measure the quality of simulation results more accurately. Recent work, such as [6], focuses on the resource aspect to generate a more accurate simulation by considering multitasking and resource profiles. Another example is considering factors such as activities and external delays in generating simulation models [3]. Techniques such as those described in [2] employ hyper-parameterization to iteratively search for the set of parameters w.r.t. more accurate simulation results. In [11], a survey of data-driven simulation approaches in process mining w.r.t. user information and insights from event data has been performed. It includes a review of current data-driven simulation approaches in process mining, their challenges, and their limitations.

Aside from the current focuses and challenges in data-driven simulation approaches in process mining, steady states and starting points of simulations are still open questions. For simulations, it is known that the ideal starting state must be close to the steady state [7]. In [9], a survey of steady-state approaches for the queuing system is proposed, where the focus is on statistical analysis rather than data-driven approaches. Tail management is the default solution for the warm-up issue, as it is general and simple. Most research efforts have been focused on determining the length of the initial transient [5]. Determining the warm-up period is also one of the most common techniques in simulations for reaching a steady state of a system [8]. The general categorization of these methods is *graphical, statistical, and heuristic*. For instance, one of the most commonly used techniques to graphically specify the warm-up period is Welch's Method [7]. The presented approach in [12], which is based on time series analysis, is also a sample of statistical analysis methods for determining the warm-up period. An example of a heuristic technique is [4], where the rule is "truncate a series of measurements until the first of the series is neither the maximum nor the minimum of the remaining sets". However, these approaches do not rely on a system's historical data.

There are approaches to use steady states in specific fields (e.g., [16]), but they are also dependent on domain knowledge. The proposed techniques for estimating the steady state in general-purpose simulations are generally not based on system data and neither consider the specific attributes, e.g., case attributes

[1] https://www.celonis.com.

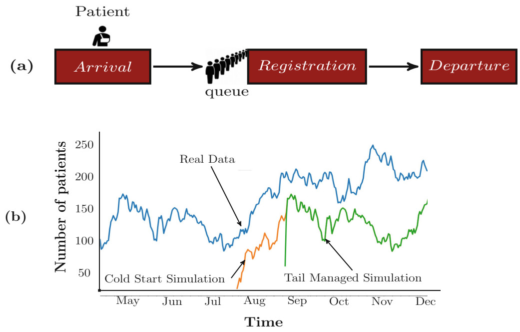


Fig. 1. Running example (a) and the number of cases in the process (b).

in the process context. The concept of starting business process simulation in its current state for short-term simulation has been proposed in [14]. In this setting, the current state is provided by the information system used. In [14], YAWL provides this state, and simulation is used to do a "fast forward" in time. In conclusion, process simulation approaches have mostly either not explicitly considered the steady-state situation or have not used event logs. Our practical components for a steady-state start can also be used for current state start and are less restrictive.

3 Running Example

Consider the simple example of the registration process in an emergency room department (Fig. 1 (a)). Patients arrive and proceed to the registration desk, where they queue until a staff member is free to process their case. The nurse evaluates the severity and registers them, and indicates their next steps. In this example, a patient (case) can be in three locations, either being processed in an activity, traveling between activities or queuing if no nurse (resource) is free.

Before we start modifying the simulation to explore changes, we first simulate the as-is situation to verify that our model correctly reflects reality. We compare the simulation results by looking at the number of cases in the process, hence the number of patients in the emergency arrival room. Figure 1(b) shows the real numbers over time, compared to a cold start simulation, and a tail-managed simulation, i.e., the initial simulation results are not considered (warm-up period). A simulation starts in an empty state with no patients. As patients are generated, the simulation fills up over time and eventually might reach equilibrium. The issue is that a hospital is never empty. This initial warm-up phase biases the simulation toward wrong conclusions. The ideal solution would be to start with the correct number of patients. The state-of-the-art solution is tail management; a longer simulation where the beginning is discarded. However, the discarded part represents misspent time and simulation resources. Furthermore, tail management raises the issue of determining the length of the warm-up period and how to estimate the cut position [8].

4 Preliminaries

In this section, we introduce process mining concepts, define an event log, and continue by elaborating on the simulation concepts used in this paper.

Process Mining. The executed activities for specific process instances at a specific point in time are the events that are captured in the form of event logs in the context of process mining. These events are recorded with associated information in an event log.

Table 1 is the hospital event log for the running example with attributes Case ID, Patient Name, Activity, Resource, and Activity Completion Time. Each row represents a unique event, and the columns contain the associated information. We define the used form of event log in Sect. 5.

Table 1. A part of the running example event log.

Event	Case	Patient	Activity	Resource	Time
871	34	John	Arrival	N/A	11:42
872	78	Alex	Registration	Nessa	12:50
873	34	John	Registration	Max	12:55
874	90	Tim	Arrival	N/A	12:50
875	78	Alex	Departure	N/A	12:58

State. The state of a process is determined by variables that describe it at a precise moment in time. We use event logs with a timestamp to describe the state of a process. If there are no events in the state, it is an empty state. The current state of a process is defined as the events that have begun but have not yet been completed in the current moment of time. In the running example, the state of the process at time t=12:00 includes two events with Event IDs, 872 and 873, where these two events and their corresponding attributes started before time t and have not finished by that time. In Sect. 5, we define states formally in Definition 2.

Simulation. The simulation engine is considered to receive a state and produce another state, i.e., given an event log, generating another event log. A simulation execution (called simulation run) can be seen as a sequence of states beginning at a starting state, where the rules of the simulation engine define the succession of states. The starting state usually is an empty state (the simulation is then called a cold start simulation).

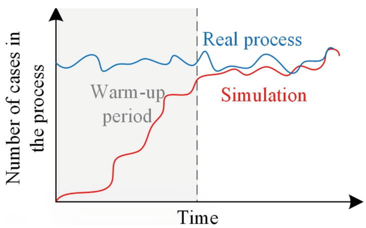


Fig. 2. Different simulation phases for an example process.

Simulations can reach such a steady state since the model remains fixed. In some cases, simulations of real-world processes do not reach a steady state due to internal and external factors, e.g., concept drift or human involvement. However, the steady state of a simulation is a useful indicator of the process's behavior. Figure 2 depicts simulation phases over time in relation to the process's real state. In an ideal situation, the simulation will eventually reach a steady state, however, in some cases, this may never happen.

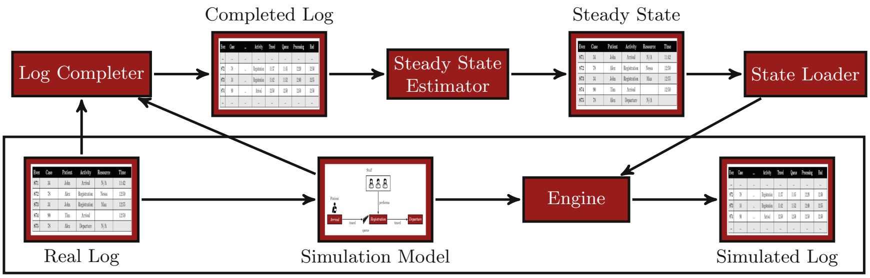


Fig. 3. The overview of the presented approach, including three main steps (top). The highlighted part (bottom) indicates the common components in data-driven simulation approaches.

5 Approach

A high-level overview of our approach is presented in Fig. 3. The selected part (bottom) illustrates the common path in data-driven simulations of processes. The process model and process aspects, e.g., resource schedules, are extracted from the event log. The simulation model is then generated (designed) on the basis of the provided insights and gets executed using a simulation engine, and the generated event logs are captured. The simulated event logs are compared with the original event log to indicate the accuracy of the simulation. This can be done iteratively to find the parameters as proposed in [2]. To make our approach for estimating a steady state of simulation models generic, we consider the simulation engine and highlighted part in Fig. 3 to be regarded as data-driven approaches in both academic and industrial tools.

Given an event log to create the simulation model, we first discover the missing attributes (1). Then, we use time-stratified sampling to create an event log that approximates a steady state (2). Finally, we use our state loader component (3), which allows us to load any event log into a simulation as a start state, to start the simulation from a steady state, see Fig. 3 (top).

Approach Settings and Assumption. Note that the simulation approach and the engine are considered *black boxes*. This allows us to adjust the approach for the general setting and make it applicable to the existing data-driven simulations in process mining. We only assume the simulation contains start and end timestamps. In our approach, we distinguish between the real queuing time, i.e., cases waiting due to a lack of resources, and the traveling time between two activities. Section 5.1 goes into more detail.

5.1 Event Log Completion

For an accurate steady-state estimation, we need a complete input event log with event attributes. Event logs, on the other hand, are limited in size and prone to errors and missing data. We use simulation models to complete an event log. The mined simulation model can be used in two ways. First, it specifies which attributes must be completed, i.e., which attributes in the simulation are relevant. Second, the simulation model can be used to quickly complete existing event logs. Every attribute in the simulation is based on either a situation-independent function (such as sampling from a probability distribution) or a situation-dependent function (e.g., queuing time depending on the number of cases queuing). For completion, we reused the situation-independent functions. The situation-dependent functions imply that the simulation model creation has completed the log, which we will also reuse. Table 2 demonstrates the event log completion provided by two methods for our running example.

Table 2. The preprocessed sample event log of the running example, which is completed with the traveling and queuing time.

Event	...	start	Travel	Queue	Processing	End
...	...	...	...	...	...	...
872	...	11:35	11:37	11:45	12:20	12:50
873	...	11:37	11:42	11:52	12:40	12:55
874	...	12:50	12:50	12:50	12:50	12:50
...	...		...	...	...	...

For our hospital example, the given event log only contained one timestamp, complete timestamps. As our simulation distinguishes travel time, queue time, and processing time (e.g., domain knowledge and user input), those attributes need to be completed, so the position of cases can be identified for creating a steady state. The simulation provides probability distributions for travel and processing times (situation-independent), and assumes that the start time of an event is the end time of the previous case event. Queue time (situation-dependent) can be inferred as the remaining time. Through the reuse of the simulation model, we thus complete our event log, as shown in Table 2.

Definition 1. *(Completed Event Log) Let $\mathcal{E}$ be the universe of event identifiers, $\mathcal{N}$ be the universe of attribute names, and $\mathcal{V}$ be the universe of attribute values. $L = (E, N, f)$ is a completed event log, where $E \subseteq \mathcal{E}$, $N = \{cid, act, travel, queue, processing, start, end\} \subseteq \mathcal{N}$, and $f : E \times N \to \mathcal{V}$ is a function that retrieves values of event attributes. We denote $\mathcal{L}$ as the universe of event logs. For an event $e \in E$ and attribute $n \in N$, $f(e, n) = \perp$, if attribute n is undefined for event e.*

In our case, a completed event log includes travel time, processing time, queuing time, start time, and end time, Definition 1. Note that more attributes, if presented in the event log, such as age of patients, treatments, or resources, can also be included in the set of attributes.

5.2 Steady-State Estimation

In this section, we elaborate on our novel approach for estimating a steady state of processes using their event logs for simulation purposes. We first define the state notation and later apply the designed steps to estimate steady states. The input is assumed to be the completed event log which notably contains start and end timestamps. The state of a process describes the process at a precise moment in time, i.e., given an event log describing the execution over time of a process, the state at any moment in time is the slice of the event log with events currently in the process (Definition 2).

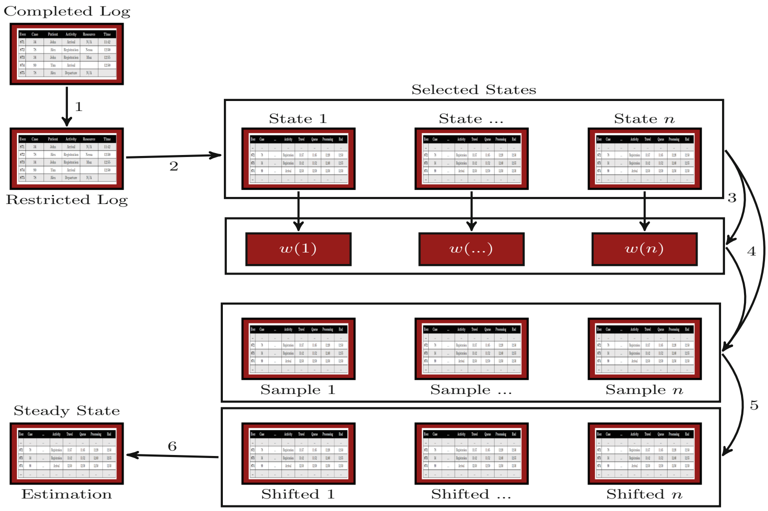


Fig. 4. The overview of steps inside the steady state estimation step.

Definition 2. *(State) Let* $T \in \mathbb{R}^+ \cup \{0\}$ *be the set of timestamps and* $L = (E, N, f)$ *be a completed event log. The* ***state*** *at time* $t \in T$ *is the sub-event-log with events active at time t, i.e.,* $L_t = (E_t, N, f_t)$*, where* $E_t = \{e \in E | f(e, start) \leq t \leq f(e, end)\}$ *and* $\forall_{e \in E_t} \forall_{n \in N} f_t(e, n) = f(e, n)$*.*

The state of the process at time $t = 12:00$ in Table 2 includes events 872 and 873. Six main steps are defined for estimating a steady state from an event log. The steps for estimating a steady state from a completed event log are shown in Fig. 4, along with their relationships and dependencies.

Step 1 - Log Restriction. The first step is for the user to limit the input event log to be used for approximation. Only relevant parts should be considered, and an unnecessarily long event log should be avoided for efficiency. This log restriction is presented using Definition 3.

Definition 3. *(Log Restriction) Let $L = (E, N, f)$ be a completed event log and $[t_{start}, t_{end}] \in T$ be a time duration. The restricted event log is $L' = (E', N, f')$, where $E' = \{e \in E | f(e, start) \geq t_{start} \wedge t_{end} \geq f(e, end)\})$ and $\forall_{e \in E_t} \forall_{n \in N} f'(e, n) = f(e, n)$.*

In the hospital example, having event logs from the last 10 years. The target steady-state approximation is a steady state of the recent hospital process. However, the hospital has evolved over time, older data might represent the hospital when it was a different process. The user thereby provides his estimate of which part of the recorded data is relevant through a time window parameter. If all the data is relevant, smaller subsets can be used for efficiency nonetheless. Here, we use the last three months, [23 May-23 July]. Note that the need to perform these steps is determined by the duration of the event log and the user domain knowledge.

Step 2 - State Selection. Given that we have restricted the event log to parts relevant to the current process, we further restrict it to the states relevant to the simulation starting time, as presented in Definition 5. This allows us to account for the potential patterns in the process. Definition 4 is used to return a set of simulation timestamps in which the states are selected.

Definition 4. *(Simulation Timestamps) Let $[t_{start}, t_{end}]$ be a time duration, where t_{start} and $t_{end} \in T$, $t_{sim} \in T$ be the simulation start timestamp, and $\delta \in T$ be the duration of a pattern. Function $tp \in T \times T \times T \times T \rightarrow P(T)$ returns the set of simulation timestamps such that $tp(t_{start}, t_{end}, t_{sim}, \delta) = \{t \in T | \exists_{m \in \mathbb{N}}\ t = t_{sim} + m.\delta \wedge t_{start} \leq t \leq t_{end}\}$*

Definition 5. *(State Selection) Let L' be a restricted event log in time duration $[t_{start}, t_{end}] \in T$, $t_{sim} \in T$ be the simulation start timestamp, $\delta \in T$ be the duration of a pattern. L'_t is a selected state at time $t \in TP$, where $TP = tp(t_{start}, t_{end}, t_{sim}, \delta)$. We denote $ST = \bigcup_{t \in TP} L'_t$ to be the set of the selected states.*

For instance, in the running example, consider that the simulation starts at 12:00 on Monday, July 24, and there is a weekly pattern within our previously selected range of 3 months. The number and types of patients highly depend on the time and weekday (proportionally more emergencies at night, alcohol intoxication on the weekend, etc.). Mondays at noon are more similar to Mondays at noon in different weeks than other moments. We therefore set a $\delta = 1$ *week* parameter. If no pattern is observed, all moments are equally relevant, and any random δ can be used.

Step 3 - Recency Weight. The relevance of the selected relevant states to the process's steady state is not equal. As the process evolves over time, older states may be less informative than current ones. A recency weight function that assigns a timestamp a relevance score based on how recently it occurred in relation to the simulation start time is provided for the user as an estimate. This step is considered to give the user the option to increase the influence of the process' more recent states on the simulation.

An estimate in the form of a recency weight function assigns an importance score to a timestamp based on its recency relative to the simulation start time. Based on that, we derive a weight function in Definition 6 for states (normalized to sum to 1), which determines their impact on the average.

Definition 6. *(Recency Weight) Let $r \in T \rightarrow [0, 1]$ be the time recency weight function that assigns a weight score depending on the distance to simulation start time t_{sim}, i.e., $r(t_{sim} - t) \in [0, 1]$, and TP be the set of simulation timestamps. For any $t \in TP$, $w(L, t) = r(t_{sim} - t)/\sum_{t' \in TP} r(t_{sim} - t')$ is the normalized state weight.*

In the running example, we observe a slow and steady increase in the number of patients. While the data from previous months is still relevant, the data from this month should have more impact. For the state L_t, function $r((t_{sim} - t)) = 1/((t_{sim} - t).weeks + (t_{end} - t_{start}).weeks)$, t_{sim} is the start of the simulation, $\delta.weeks$ returns the number of weeks in a duration, and t_{start}, t_{end} are the start and end of the restricted event log. As a result, Monday, July 17, has a normalized weight of 0.11; Monday, July 10, has a normalized weight of 0.10; and so on until Monday, May 23, has a normalized weight of 0.06.

Step 4 - Weighted Sampled State. With the selected states and their importance, we can now build a representative average by sampling the states. We bias the sampling by assigning a weight to each event in a state as the selection probability using Definition 7.

Definition 7. *(Weighted Sampled State) Let $L_t = (E_t, N, f_t)$ be a selected state, w be the state weight recency function, and $\rho : 2^E \rightarrow 2^E$ be a function that randomly selects a subset of events, i.e., given $E_1 \subseteq E_t$, $\rho(E_1) \subseteq E_1$, where each event has probability $w(L_t)$ of getting selected. The weighted sample of L_t is $L_t^s = (\rho(E_t), N, f_s)$, where $\forall_{e \in \rho(E_t)} \forall_{n \in N} f_s(e, n) = f(e, n)$.*

In the example hospital, with the previously defined weight, around 11% of events from Monday, July 17, 12:00 will be selected, and 6% of events from Monday, May 23, 12:00.

Step 5 - Sample Shifting. The selected events from the different states do not form a state together, since they originate from different times. To build one consistent state, we first shift their positions at the simulation start time to be the same w.r.t. the sampled moment. Definition 8 is designed to shift the samples.

Definition 8. *(Sample Shifting) Let $t_{sim} \in T$ be the simulation start time and $L_t^s = (E_s, N, f_s)$ be the sampled state. The shifted sampled state to the* *start of simulation is* $L^s_{t \longrightarrow t_{sim}} = (E_s, N, f_s^{t_{sim}})$ *such that for the two attributes* $start, end \in N, \forall_{e \in E_s} f_s^{t_{sim}}(e, start) = (t_{sim} - f_s(e, start)) + f_s(e, start)$ *and* $f_s^{t_{sim}}(e, end) = (t_{sim} - f_s(e, end)) + f_s(e, end)$, *for* $n \notin \{start, end\}$, $f_s^{t_{sim}}(e, n) = f(e, n)$.

In our example, Table 2, assume event 872 was selected within the state on July 17 at 12:00 w.r.t. this state, and the patient was queuing for another 20 min. We shift the timestamps so that the event has now started queuing on July 24 at 11:45 and will do so until July 24 at 12:20.

Step 6 - Sample Merging. We finally merge the samples into a new state using Definition 9.

Definition 9. *(Sample Merging) Let* TP *be the set of simulation timestamps.* $L_S = \bigcup_{t \in TP} L^s_{t \longrightarrow t_{sim}}$ *is the set of merged states.*

In Definition 9, by merging the states, the tuple of merged events, attributes, and functions of the states is created. For our running example, we created an event log that is also a state, since all events are active at simulation time. This event log is an average representation of Monday 12:00 from the last three months, with a stronger influence from recent data.

5.3 State Loader

The final step loads a state as the starting state of the simulation. We omit details, as this step depends strongly on the simulator. The intuition is that since our events are all complete, we know their position w.r.t. the start timestamp at each moment. For example, event 872 from Table 2 for a simulation starting at 12:00 should be loaded into its activity with 20 min remaining. This can be done by reusing the functions the simulation engine uses to advance cases through the system.

6 Evaluation

To evaluate our approach, we designed a framework to compare the simulation results in three different settings. The goal of the evaluation is to assess the ability of our approach to capture the accurate steady states of processes using both real and synthetic event logs, as well as the use of such an estimation as a simulation starting point. We simulate the three scenarios for each event log, starting from a cold start, using tail management, and starting from the estimated steady state. The results enable us to compare and demonstrate the impact of our approach in practice. Furthermore, we discuss and illustrate the various situations in which steady-state estimation is applicable and should be considered. As discussed in Sect. 5, the simulation engine and the quality of the simulation results are not the focus of the evaluation. Owing to privacy considerations, the public sharing of both the data and the implemented codes within commercial tools are limited. Nonetheless, the evaluation details of the presented datasets, including performance metrics, are reported in the evaluation section.

Evaluated Event Logs. The simulation models of the two real-world event logs were created and validated jointly by process analysts and the companies. We reuse them to show that our steady-state start presents a significant improvement due to the slow convergence of the process in one situation, and that the simulation with a steady state is not necessary given the process characteristics in the other one. Our synthetic hospital model is designed to show a different use case, instability, i.e., the incoming cases exceed the capacities of the process. In this case, a steady-state start is an improvement and the only viable solution.

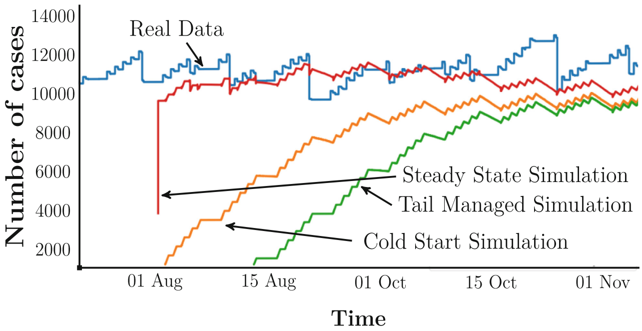


Fig. 5. The comparisons of the simulated number of cases in the process for the medical company.

The number of cases in the process is the primary metric used to represent simulations. We detail whether other metrics, such as case throughput time, are affected in the various data sets. The longer the case throughput time, the more effective the use of steady-state estimation is expected to be. If the throughput time is short, the simulation with a cold start also reaches the steady state quickly, and the tail-management strategies appear plausible.

6.1 Real-World Event Log

We tested our method on two real-world event logs with different characteristics, such as different throughput time of cases within the process. Because of privacy concerns, the event logs have been anonymized.

Medical Company Event Log. The throughput time for a single case in the medical company process is around 24 days. The process has a high case load (10000-12000 cases at all times). Most of the time is spent traveling in between activities, time that can be attributed to waiting for external processes (e.g., waiting for customer payment). Furthermore, the process is very stable, with a consistent case load throughout the years. Figure 5 shows the results for the real medical invoice process.

Cold start simulation performs poorly under these conditions. Due to the high case load and long throughput times, the simulation converges slowly toward a stable result. It takes 60 simulated days for the cold start simulation to stabilize. The tail-managed simulation must thus simulate 60 additional days to eliminate the warm-up phase. Also, the required warm-up time is only obvious in hindsight. The common tail-managed simulation uses a warm-up time of 20%, which proves insufficient. Our approach shows a striking improvement. Due to the stability of the process, almost no warm-up is required. The simulation is immediately in a state that is representative of the real process. Since a majority of the throughput time of cases is spent traveling between activities in the process, results of the simulation other than process load are not impacted (for example, throughput time), all simulations deliver the same results.

Sensitivity Analysis with 95 Confidence Interval and 50 Repetitions. Each simulation was performed 50 times. We compute the 95 confidence interval for each moment by taking the 50 computed values for cases in the system. For each simulation, we plot the lower and upper bounds as shown in Fig. 6. Because of the small size of the confidence interval in relation to the scale, the lower and upper bounds are mostly indistinguishable. The simulations remain consistent, and the current state start and steady-state start outperform the cold and tail simulations significantly. Even the worst steady or current simulation outperforms cold and tail simulations. Furthermore, the nontrivial state start adds 17.37 s of overhead. Event log completion, on the other hand, only needs to be done once and can be reused for all simulations. 3.14 s is the worst overhead for a nontrivial state start after completion.

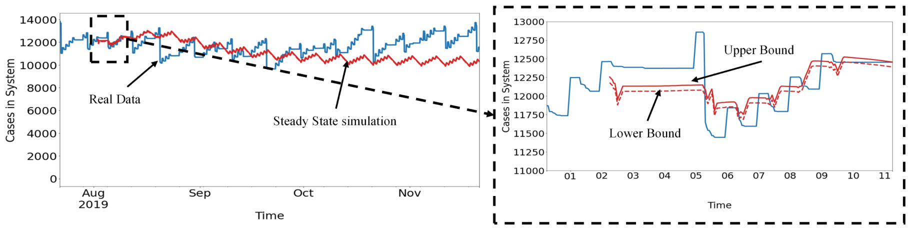


Fig. 6. A sample lower and upper bounds for one of the simulations.

Car Production Event Log. Figure 7 shows the results for a car production company. The event log has a short duration (three months), and the process has a throughput time of 2 h. As such, the initial state loaded into the simulation is processed after 2 h. In such cases, a steady-state start is not required, but the improvement can be seen by starting from a more realistic state compared to the real process.

6.2 Running Example Event Log

We deliberately designed our running example process to have different characteristics as a showcase. The process has a lower case load and short throughput times (a few hours). However, the process is unstable. The amount of patients slightly exceeds the capacities of the hospital's registration process, resulting in a slowly increasing queue. Patients spend the majority of their time queuing.

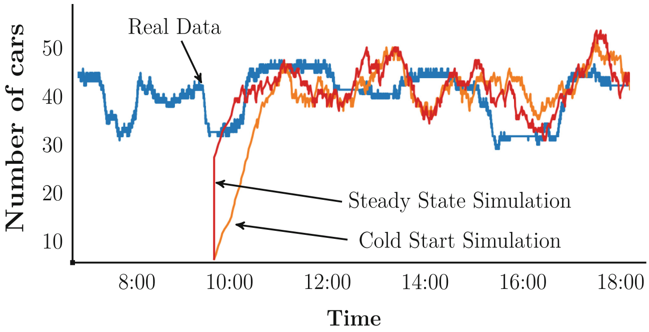


Fig. 7. The number of cases in the car production process w.r.t. three designed simulation strategies.

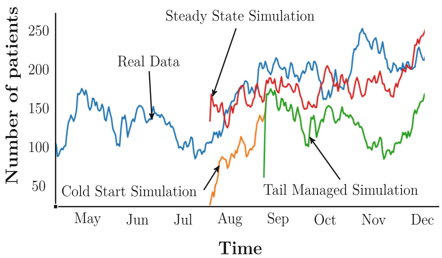


Fig. 8. The number of patients (cases) in the running example (hospital example) and three designed simulation strategies.

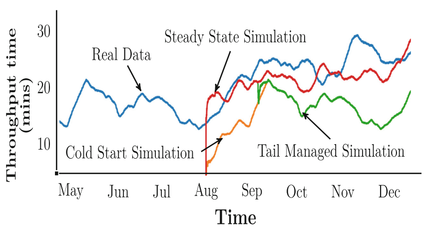


Fig. 9. The throughout time of cases in the running example (hospital example) process w.r.t. three designed simulation strategies.

Figure 8 depicts the results for our hospital example. The cold start simulation starts in an unrealistically empty state. It never reaches the real data because of the existing concept drift. Because the simulation starting from the cold start will be inaccurate, the tail-managed simulation will be inefficient in terms of providing more accurate simulation results. As such, it also never reaches the real process load. Our steady-state estimation is a significant improvement. Our simulation starts close to the real process and remains an improvement at all times. Figure 9 graphs the throughput time, i.e., how long it takes patients to complete the process. As this value depends on the queue time, and hence the process load, the same problem with cold start simulation can be observed, and the same improvement can be observed with steady-state start. All metrics that do not depend on process load remained identical for all simulations.

In such scenarios, the cold start simulation is the worst case. By extension, tail-managed simulation cannot solve this. Since the cold start simulation never stops warming up, no amount of cut-off will improve the result. This problem is solved by using a steady-state start. However, in our example, due to the process's instability, the approximated starting point is not as good as in the previous examples. Yet, it remains vastly superior to an empty start. Our steady-state start simulation is an accurate representation of the process. These results extend to other metrics. Because patients spend the majority of their time queuing, the number of patients correlates with the throughput time, resulting in the same issues for cold start and tail-managed simulation.

In order to assess the performance of our approach in practice, we ran the workflow several times. All simulations took approximately 30 s. The steady-state estimation module takes about 2 s on average, and the state loading module takes about 0.1 s.

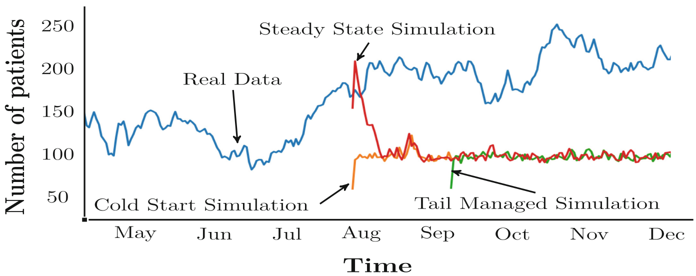


Fig. 10. The comparison of simulation results for three starting points in what-if analysis scenarios for the running example (hospital).

What-If Simulation. So far, we replicated the real data, and steady-state start reduced warm-up time. For what-if simulations, our approach does not aim to minimize warm-up, but make it a valuable part of the simulation instead. In a cold start simulation, the warm-up indicates the states the system goes through from an empty state until stabilization. Unless the system is actually regularly empty, this is irrelevant. In a what-if simulation (the simulation is modified to represent a change in the system), starting from an estimation of the new steady state would optimize convergence. However, when starting from the steady-state approximation of the current process, warm-up represents the transitory period that changes create until a new equilibrium is found.

In our example, the hospital aims to stop the increase in queues. They assign one additional nurse to registration, who assists patients in between activities.

The simulations for this situation are shown in Fig. 10. Although cold start simulation happens to converge quicker than steady-state start simulation, the latter indicates that the changes would need two weeks to break down backlog and reach the new equilibrium.

6.3 Discussion

Our goal was to demonstrate the significance of the starting point in process simulations and propose a method to estimate it using event logs. This step in the process simulation is either ignored, i.e., assumed to begin from an empty state, or in practice employs a tail management strategy, i.e., simulating from a cold start and disregarding the considered duration as a warm-up period. Patterns are one of the crucial pieces of information that should be represented in the simulation models and be reproducible, as they affect the extraction of steady states. In the case of our running example, a few example patterns are as follows. On the weekend, there are more cases of alcohol intoxication. In the winter, there are more flu cases. During the night, the proportion of severe cases is higher. We considered the existence of patterns in our approach by state selection to sample states. This allows us to include patterns in our steady states.

Our results show that simulating from steady states is effective, especially when the process has a long throughput time as well as scenarios with a long convergence time (high case load). We showed that the warm-up period for one of the event logs would be 60 days of simulation to reach the steady state. For unstable simulations (with no convergence), we are not aware of other general purpose solutions. Our work is most similar to [14], in which the simulation begins from the current state. Steady state includes more general process behavior because it is sampled across the process rather than just the current moment, e.g., the current state can be an anomaly in the process.

Estimating process states as a part of our approach is practical in different use cases. Instead of having a lot of time-dependent parameters in the simulation, chaining simulation models by using the end state of one simulation as the start state of the next is one of the effective scenarios. The difference is that only one method is required, and then any time-related change can easily be expressed. This method also scales beautifully; new elements in the simulation will be directly expressible in time.

Example of Mixed Models Using Steady States. There are other scenarios in which the extracted steady states are effective. Consider an x-ray machine breakdown, creating an increasing queue. There are two potential replacements, one available quickly but slowly, and one available in two weeks but efficiently. The current simulation cannot compare these two solutions, as it can only simulate them as if they have always existed. Our steady state can then be used to simulate what would happen if we started from there. However, by chaining models, we can start from the steady state, simulate the problem scenario, and then explore the different solution scenarios. As an alternative, suppose we discover through process mining that the first and second Mondays of each month are less efficient. Adding parameters to every resource and arrival rate is cumbersome, but we can very easily chain together simulations to express this.

7 Conclusion

In this paper, we proposed an approach for estimating the steady state start point for the simulation of processes. The steady state estimator is innovative and efficient in the context of data-driven process simulation. The approach is designed to be independent of the simulator. The evaluation shows that it is efficient, as it scales only with the number of sampled moments, and the number of cases in the system. We extended it to include weights, for concept drift, as well as a curated selection of existing patterns. Since the event log is directly sampled, anything represented in the log is already included in the steady state. The idea of using non-trivial starting states still holds much untapped potential. As data collection, process mining and simulation models evolves and improve, starting from the steady or current state will open more efficient and diverse analysis potential.

References

1. van der Aalst, W.M.P.: Process mining and simulation: a match made in heaven! In: Computer Simulation Conference, pp. 1–12. ACM Press (2018)
2. Camargo, M., Dumas, M., González, O.: Automated discovery of business process simulation models from event logs. Decis. Support Syst. **134**, 113284 (2020)
3. Chapela-Campa, D., Dumas, M.: Modeling extraneous activity delays in business process simulation. In: 2022 4th International Conference on Process Mining (ICPM), pp. 72–79 (2022)
4. Conway, R.W.: Some tactical problems in digital simulation. Manage. Sci. **10**(1), 47–61 (1963)
5. Eickhoff, M., McNickle, D., Pawlikowski, K.: Detecting the duration of initial transient in steady state simulation of arbitrary performance measures. In: Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools. ValueTools 2007 (2007)
6. Estrada-Torres, B., Camargo, M., Dumas, M., Yerokhin, M.: Discovering business process simulation models in the presence of multitasking. In: Dalpiaz, F., Zdravkovic, J., Loucopoulos, P. (eds.) RCIS 2020. LNBIP, vol. 385, pp. 381–397. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50316-1_23
7. Law, A.M., Kelton, W.D., Kelton, W.D.: Simulation modeling and analysis, vol. 3. Mcgraw-hill New York (2007)
8. Mahajan, P., Ingalls, R.: Evaluation of methods used to detect warm-up period in steady state simulation. In: Proceedings of the 2004 Winter Simulation Conference 2004, vol. 1, p. 671 (2004)
9. Pawlikowski, K.: Steady-state simulation of queueing processes: survey of problems and solutions. ACM Comput. Surv. **22**(2), 123–170 (1990)
10. Pourbafrani, M., van der Aalst, W.M.P.: Interactive process improvement using simulation of enriched process trees. In: Hacid, H., et al. (eds.) Service-Oriented Computing – ICSOC 2021 Workshops. ICSOC 2021. LNCS, vol. 13236, pp. 61–76. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-14135-5_5

11. Pourbafrani, M., van der Aalst, W.M.P.: Data-driven simulation in process mining: introducing a reference model. In: Communications of the ECMS, vol. 37, issue 1, Proceedings, pp. 411–420 (2023)
12. Robinson, S.: A statistical process control approach for estimating the warm-up period. In: Proceedings of the Winter Simulation Conference, vol. 1, pp. 439–446. IEEE (2002)
13. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering simulation models. Inf. Syst. **34**(3), 305–327 (2009)
14. Rozinat, A., Wynn, M.T., van der Aalst, W.M.P., ter Hofstede, A.H.M., Fidge, C.J.: Workflow simulation for operational decision support. Data Knowl. Eng. **68**(9), 834–850 (2009)
15. Tumay, K.: Business process simulation. In: Proceedings of the 28th Conference on Winter Simulation, pp. 93–98 (1996)
16. William, N., Tracy, C., Nick, B.: Improving the accuracy of random waypoint simulations through steady-state initialization. In: Proceedings of the 15th International Conference on Modeling and Simulation, pp. 319–326 (2004)

Analytics Pipeline for Process Mining on Video Data

Arvid Lepsien[1(✉)], Agnes Koschmider[2], and Wolfgang Kratsch[3]

[1] Department of Computer Science, Kiel University, Kiel, Germany
ale@informatik.uni-kiel.de
[2] Chair of Business Informatics and Process Analytics, University of Bayreuth, Bayreuth, Fraunhofer FIT, Bayreuth, Germany
agnes.koschmider@uni-bayreuth.de
[3] Technical University of Applied Sciences Augsburg, Fraunhofer FIT, Augsburg, Germany
wolfgang.kratsch@fim-rc.de

Abstract. Process mining has shown that it provides valuable insights in terms of uncovering bottlenecks and inefficiencies in processes or identifying tasks for automation. However, process mining techniques expect structured input data that is at a high (business) level of abstraction. Recently, the benefits of process mining for unstructured data which is at a much lower level of abstraction have been demonstrated, e.g., for IoT data or time series data. It can be expected that the demand for methods efficiently processing these kinds of data for process mining will continuously increase. Hence, in this paper, we present an approach that allows the translation of video data into higher-level, discrete event data, thus enabling existing process mining techniques to work on data tracked in videos. Particularly, we used a combination of object tracking, spatio-temporal action detection, and techniques for raising the abstraction level of events. The evaluation results show that meaningful event logs can be extracted from an unlabeled video dataset, validating both the implementation and the feasibility of our approach.

Keywords: Process mining · Event log extraction · Unstructured data · Activity recognition

1 Introduction

Process mining (PM) strives to discover, monitor, and improve processes by extracting knowledge from structured event logs typically sourced from core information systems (e.g., ERP systems) [1]. However, for many processes (e.g.,

We thank Jan Bosselmann for his support with the implementation, and the Institute of Agricultural Engineering at Kiel University for providing a dataset and use case. This project has received funding from the State of Schleswig-Holstein under the Datencampus project grant no. 220 21 016.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 196–213, 2023.
https://doi.org/10.1007/978-3-031-41623-1_12

highly manual processes), structured data is not available, resulting in blind spots that are not covered by traditional PM approaches. Consequently, PM often does not deliver a full, end-to-end process analysis, but only insights into digitized parts of processes. However, process-related behavior may also be captured by various types of unstructured data, i.e., data that is not organized in a pre-defined manner, or at least no data scheme that is directly applicable for PM purposes. Video cameras are an especially promising source of process-related unstructured data because (1) video cameras are inexpensive and easy to set up and (2) a high amount of diverse information can be extracted from video recordings using modern computer vision techniques.

First approaches concerned with the analysis of video data for PM have been proposed, which were used to examine structured processes from highly specific contexts in logistics and production, recorded in laboratory settings [11,14]. In reality, however, processes are often chaotic and unstructured, due to irrational actors or unpredictable internal and external disorders. To the best of our knowledge, no existing approach offers a solution to analyze these rather chaotic and unstructured processes using video data. Therefore, we present a novel, end-to-end analytics pipeline for performing PM using video data, that enables the video-based analysis of unstructured processes with the help of systematic event abstraction and event log processing. By evaluating our pipeline on videos capturing the daily activities of fattening pigs, we employ a chaotic process with a limited set of known activities. This allows us to perform a technical evaluation of activities but also allows for the incorporation of feedback from domain experts (i.e., agricultural scientists) to confirm the validity of our automatically mined insights.

The paper is structured as follows. First, we introduce our end-to-end analytics pipeline for PM on video data (Sect. 2). Then, we introduce our implementation of this pipeline (Sect. 3) and our use case (Sect. 4), and report the evaluation of our pipeline including the implementation and application to real-world data (Sect. 5). Finally, we outline related work (Sect. 6) and conclude our research and discuss its limitations as well as potential avenues for future research (Sec. 7).

2 Process Analytics Pipeline

This section presents our analytics pipeline for PM for video data, which consists of six steps, as depicted in Fig. 1. The main objective of this pipeline is the discovery of process models from unstructured data, specifically video data. The pipeline enables the identification of structure in terms of a process model from unstructured data, facilitating the detection and explanation of bottlenecks, anomalies, and causalities. In order to apply the pipeline, an analysis goal or problem statement must be defined, which significantly influences the method selection and parameters for each step of the pipeline. The design of the pipeline is based on the pipeline originally presented in [17]. Compared to this pipeline, the proposed approach was extended by (1) renaming of the preprocessing steps

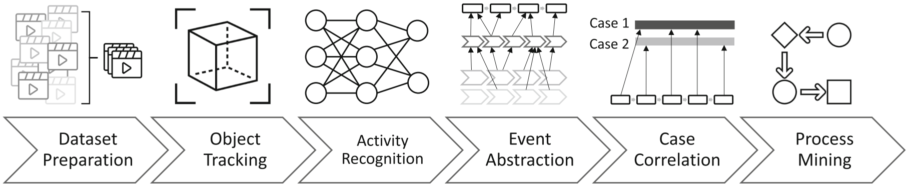


Fig. 1. Process analytics pipeline for process mining on video data (based on [12,17])

to better reflect the tasks related to *Dataset Preparation* and *Object Tracking*, (2) refinement of the *activity recognition* step into the three separate steps of *Activity Recogniton*, *Event Abstraction*, and *Case Correlation*, and (3) removal of the step *mine domain specific knowledge* and the refinement loop. Additionally, this paper contains details on how to select appropriate methods for each pipeline step, a full implementation of the analytics pipeline, and the evaluation of the approach in a real-world use case.

A critical and necessary requirement of the pipeline is the extraction of *events* with *timestamps* from unstructured data. Events are defined as any relevant observations related to a process and do not need to be linked to a specific process activity at the outset. Without the notion of *events* with *timestamps*, the pipeline cannot generate any meaningful results.

2.1 Dataset Preparation

In the first step of the pipeline, a raw video dataset is processed in order to transform it into a unified format. The challenges of unifying video data are to bridge the gap between different resolutions as well as frame rates that arise from using different cameras or recording devices and to combine contiguous recordings that are split across multiple video files.

First, relevant video segments are selected based on the analysis goal. The relevancy of video segments depends on factors such as the presence of specific objects or activities, or on their context. For instance, if the goal is to analyze the performance of a process in a specific part of the day, only video recordings from that part of the day are relevant. Once the relevant segments have been identified, they are resampled to a common frame rate and their resolution is scaled down to reduce the computational load. Furthermore, the degree of resolution reduction depends on the detail required by subsequent pipeline steps, which, in turn, depends on the analysis goal. It is important to reduce the resolution in a controlled manner, as otherwise this could result in the loss of utility of the video segments.

2.2 Object Tracking

In the second step of the pipeline, relevant objects are detected and tracked within the selected video segments. Objects are relevant to the analysis if they

Table 1. Exemplary inputs and outputs of spatio-temporal action detection

Frame	Bounding Box	Track ID	Activities
...	...	...	...
13759	[261, 314, 453, 432]	1	{A: 0.601, B: 0.102}
13791	[274, 315, 566, 410]	1	{A: 0.629}
13823	[314, 295, 638, 399]	1	{A: 0.489, C: 0.222}
13855	[459, 248, 696, 374]	1	{A: 0.699, C: 0.217}
...	...	...	...

are related to the observed process, which are typically the actors performing the process activities. The objective of this step is to extract context information for subsequent steps. Specifically, the bounding boxes of relevant objects are required as input for *Activity Recognition*, and the movement trajectories for *Event Abstraction* as well as *Case Correlation*. The quality of object tracking significantly influences the quality of the subsequent steps.

A tracking-by-detection approach is used for *Object Tracking*, which consists of separate object detection and tracking models [19]. Tracking-by-detection is a suitable approach for this pipeline step, because it has state-of-the-art quality, and only the object detection model needs to be customized for different use cases. Deep learning models, which have been pre-trained on large and heterogeneous image datasets, can be adapted for object detection in order to detect customized objects with few labeled examples, even if these objects were not contained in the original training dataset [26]. *Object Tracking* outputs the bounding boxes of relevant objects in all frames of the selected videos, as well as movement trajectories that identify these objects throughout video segments with tracking IDs.

2.3 Activity Recognition

In the third pipeline step, low-level events are extracted from the selected videos using deep learning. Specifically, a spatio-temporal action detection technique [8] is used, which can detect multiple actions performed concurrently by multiple objects in the same video. Since this is a supervised learning task, a dataset of videos labeled with actions related to the analysis goal must be manually created to train the deep learning model. To simplify the subsequent step of *Event Abstraction*, the training dataset is labeled with actions that directly correspond to the activities of the analyzed process.

An excerpt of inputs and outputs of this pipeline step is listed in Table 1. Inputs for the detector include the previously detected bounding boxes and tracking IDs of relevant objects for each frame. The trained model then analyzes segments of the videos at a fixed interval of frames to detect the activities performed by each object. As seen in the *Activities* column, multiple activities are detected with varying confidence scores for each object and frame, rather than

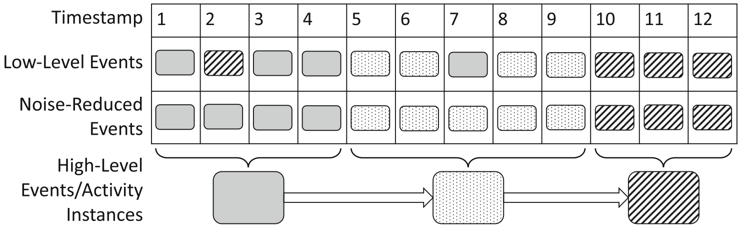


Fig. 2. Abstraction from low-level events to high-level activity instances

single specific activities, which needs to be addressed in the following pipeline step. In summary, *Activity Recognition* outputs a list of low-level events, each referring to the activities that were observed for an object at a specific time.

2.4 Event Abstraction

In the next step of the pipeline, the low-level events from *Activity Recognition* are abstracted to high-level events and finally correlated with activity instances (see Fig. 2). *Activity Recognition* generates multiple, fine-granular events for each observed activity. However, PM algorithms expect abstract, high-level events that denote only the status of the activity, such as the start and end times of execution. Because the activity classes used for *Activity Recognition* directly correspond to the activities of the analyzed process, conceptual abstraction of low-level events is not required. Rather, the low-level events need to be aggregated *temporally* to high-level events corresponding to entire activity instances. Consequently, *Event Abstraction* is done in two steps: first, in order to reduce local noise caused by imperfectly detected activities, a moving average is applied across the confidence scores assigned to the activities of each object, which removes implausibly short or sudden changes of the detected activities. An appropriate size for the moving average window needs to be determined based on the typical duration of relevant process activities, in an effort to avoid removing short activities unintentionally, while simultaneously removing as much noise as possible. After noise reduction, if multiple activities are detected for the same object within the same low-level event, the activity with the highest score is selected. Second, the noise-reduced events are abstracted to high-level events by aggregating repeated observations of the same activity in the sequence of events for each object. Occurrences of activities, i.e., activity instances, can then be identified through the contextualization of high-level events into the realm of a specific process [12,32]. For this purpose, we assume that a one-to-one mapping can be constructed from high-level events to activity instances, and directly treat the high-level events as activity instances. The output of *Event Abstraction* is a log of high-level activity instances, which we consider as an event log without a case allocation.

2.5 Case Correlation

The objective of the fifth pipeline step is to organize the activity instances identified from *Event Abstraction* into cases. Case correlation is a pivotal step of the pipeline since the cases are interpreted as process instances, which implicitly define the process that will be discovered. Case correlation can be approached in three distinct ways: (1) in certain scenarios, case identifiers may be provided externally, either explicitly (e.g., through manual annotations) or implicitly (e.g., one case per video file, as in [11]). (2) Depending on the analysis goal, case correlation can also be performed using information extracted in the previous steps together with domain-specific assumptions. For instance, if the analyzed process is characterized by well-defined start as well as end activities and each process instance is executed by precisely one object, the sequence of activities for each object can be split into cases that run from each start activity to the first following end activity [14]. (3) When domain-specific assumptions are inadequate to construct cases that align with the analysis goal, advanced algorithmic techniques for case correlation, such as those described in [7], can be applied. Nevertheless, knowledge about the analyzed process is still required to choose an algorithm with appropriate properties, such as support for loops or parallelisms inside cases. The output of this step is an event log suitable for PM.

2.6 Process Mining

The final step of the pipeline addresses the application of PM algorithms for process discovery or conformance checking. Usually, PM techniques are designed for structured processes with a limited set of process variants [6], and may not provide satisfying results when directly applied to event logs of unstructured processes. For instance, the application of process discovery algorithms on event logs of unstructured processes will commonly result in either highly overgeneralized or complex process models, which allow little insights into the process.

To address this issue, we divide the event log of an unstructured process into multiple, more structured sub-logs of similar process variants using trace clustering [31]. For trace clustering approaches, it is essential to define features related to the event log, which are then used to cluster the event log of interest. The way the features are defined and selected significantly impacts how the event log is split into clusters. Trace clustering divides an event log into multiple independent sub-logs, and PM techniques are then applied to each sub-log separately, enabling more accurate capturing of the underlying structure of an unstructured process, and providing more valuable insights than an unclustered log.

3 Implementation

We implemented a framework that consists of modular components, each corresponding to one step of the analytics pipeline as described in Sect. 2. For each module, users can select an appropriate method out of a set of methods. These methods are implemented in a generalized way and are not limited to a specific use case. Our implementation is available on GitHub[1]. In the following, the implemented methods are described for each module.

Dataset Preparation. In order to combine video sequences split into multiple files, rescale video resolution and resample video frame rate, we use FFmpeg[2]. To filter out areas of content not relevant to the analysis, we provide an option for using a static mask.

Object Tracking. We integrated interfaces for YOLOv7 [28] and Detectron2 [30] as object detectors. ByteTrack [33] and OC-SORT [4] were implemented as detection-based trackers. Labelme [27] is used to create training datasets for object detection. In addition, we implemented filters to address common issues in object detection such as low confidence, strongly overlapping, and implausibly small bounding boxes.

Activity Recognition. We employ MMAction2 [21] and SlowFast [8] for spatio-temporal action detection. To allow integration of custom training datasets, we added a feature in a labeling tool [2] that allows transferring training datasets into the format of the Atomic Visual Actions (AVA) dataset [9].

Event Abstraction. For event abstraction, we implemented the temporal aggregation technique as described in Sect. 2.4.

Case Correlation. We offer three options for case correlation: (1) tracking ID-based correlation to construct one case for each object for a complete video sequence (e.g., a day), (2) temporal segmentation-based correlation to construct multiple cases for each object based on the time of day, and (3) correlation based on pre-defined start/end activities (see Sect. 2.5). In the third option, further filters can be applied, for example, to require multiple repeats of the end activity before a case terminates or define the length of a start activity instance. If required, the interfaces of the Case Correlation module also support the application of advanced case correlation techniques.

Process Mining. We implemented multiple case-level features, which can be extracted from an event log, and used as input for trace clustering using the algorithms provided by scikit-learn [23]. In particular, we use a combination of standard scaling, principal component analysis (PCA), and k-means clustering. We use the heuristic miner [29] and inductive miner infrequent [15] algorithms for process discovery and token-based replay for conformance checking, both provided by pm4py [3]. Moreover, we also support the export of event logs for use by external PM tools such as Disco[3].

[1] https://github.com/arvidle/video-process-mining-public.

[2] https://ffmpeg.org/.

[3] https://fluxicon.com/disco/.

4 Use Case

We applied the analytics pipeline to a real-world video dataset of surveillance recordings from a conventional pig farming environment. The benefits of the use case are as follows: (1) the behavior of pigs is limited to a few activities, which significantly simplifies activity recognition compared to recognition of human activities in smart homes or smart factories. (2) Pig behavior is well-researched, and the knowledge on pig behavior can be used to verify the analysis results. Finally, (3) no privacy concerns need to be considered. In the future, however, we plan to transfer the pipeline to more complex use cases.

The behavior of fattening pigs is generally limited to resting (lying, sitting, and standing), locomotion (moving, and investigating their surroundings), feeding/drinking, defecating, and playing with toys [35]. Fattening pigs in particular spend between 60–85% of the day lying [35]. Pigs autonomously divide their pens into functional areas associated with specific activities [22]. For instance, defecation is typically done in a small (partially) sheltered area, e.g., near walls or in a corner, which is located opposite to the feeding area.

Previously, video-based research of pig behavior was done manually, i.e., a person observed the behavior of pigs. The goal of our data analysis was to *automatically* monitor common behavioral patterns and evaluate the division of the pigpens into functional areas.

We recorded video material of a pigpen with eleven pigs at a resolution of 1920×1080 pixels and 18.75 FPS from 6:00 a.m. to 6:00 p.m. over a period of four weeks. To reduce processing times, the video files of five consecutive days were sampled from the complete dataset. For each of those days, the full recordings were selected as relevant, to capture behaviors throughout the whole day, and the video files were consolidated into one preprocessed video for each day. For this particular use case and analysis goal, video resolution was reduced to 854×480 pixels, significantly reducing processing times while keeping sufficient visual detail.

Although numerous approaches exist to extract behavioral information from surveillance video of pigpens [5], these approaches are limited to the detection of isolated activities and not a process composed of several activities. Therefore, our approach has the novel potential to explain the influences and causalities of activities in this context.

5 Evaluation

This section summarizes the evaluation of all steps from *Event Abstraction* to *Process Mining* for our use case. After detecting activities in the selected videos, we used the procedure shown in Fig. 3 to evaluate our approach. To validate the result stability, we applied conformance checking, while the meaningfulness of the extracted event logs and process models was confirmed by discussing the results of our analysis with domain experts. By synthesizing the findings from these separate evaluations, we can show that our approach is able to extract meaningful, PM-compliant event logs.

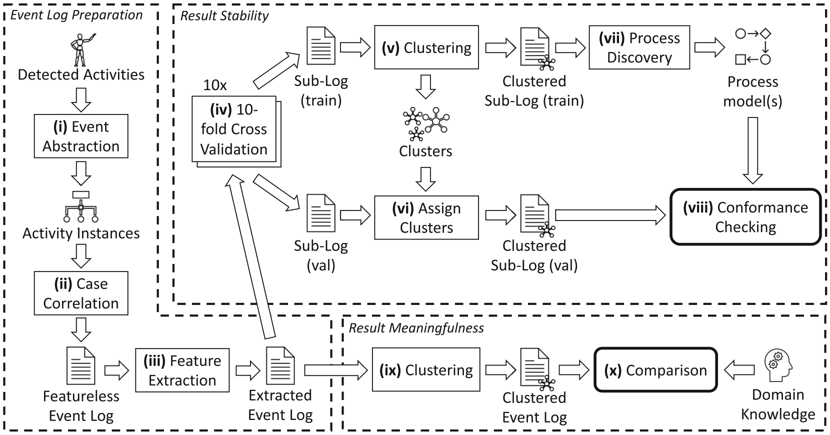


Fig. 3. Procedure to evaluate our analytics pipeline.

5.1 Pre-Processing

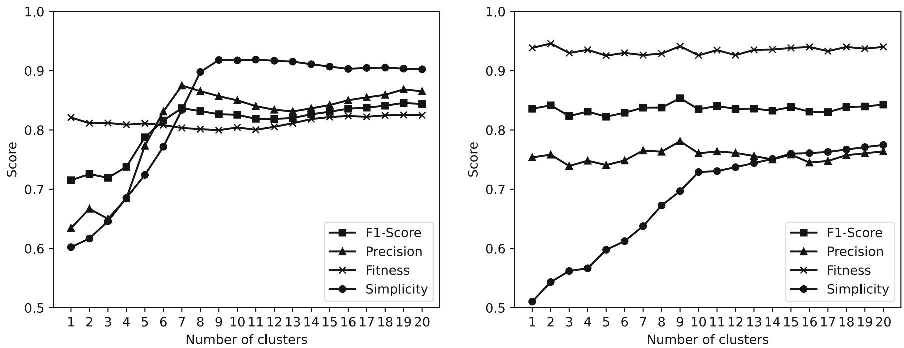
First, activities needed to be detected in the selected videos by conducting the first three steps of the analytics pipeline. A training dataset for object detection was created by sampling a total of 614 frames from the complete dataset, and pigs were labeled with bounding boxes in each frame. This training dataset was used to train a pre-trained YOLOv7 model, which was used in combination with the ByteTrack algorithm to track the pigs in the selected videos. Then, the relevant activities requiring labeling for *Activity Recognition* were determined by interviewing a domain expert. This included the following eight activities: *lying*, *sitting*, *standing*, *moving*, *investigating*, *feeding*, *defecating* and *playing*. Behavior that could not be described by one of these activity classes was categorized to the class *other*. A total of 9240 of these activities were sampled from the complete dataset and manually annotated, and then used to train a pre-trained SlowFast 4×16 model, which was used to detect the pigs' activities in the selected videos.

Preparing the labeled training datasets for the object detection and spatio-temporal action detection models proved to be the most labor-intensive manual tasks required to instantiate the analytics pipeline. Additionally, applying the trained deep learning models to the selected videos was the most computationally-intensive task in the analysis. On average, the object detector recognized 10.6 of 11 pigs per frame, and the pigs were tracked for 18 min before being lost by the tracker. The SlowFast model reported a validation mean average precision (mAP) of 0.7365, which was considered significant to continue the analysis.

5.2 Event Log Preparation

An event log was prepared from the detected activities and used to evaluate both result stability and event log meaningfulness. The same event log was used for both steps of the evaluation.

We used a moving average window size of 20 low-level events for the noise reduction in *Event Abstraction*, which is shorter than the duration of common pig activities, but sufficiently long to reduce local noise. Then, we applied the *Case Correlation* based on specific start/end activities separately for each pig (identified by its respective tracking ID), with *lying* being both the start and end activity. Thus, the process starts with a pig standing up and ends with the pig lying back down. Feature vectors were extracted from each case consisting of activity counts, total duration per activity, case duration, and directly-follows relations. Then, we scaled the feature vectors and reduced them using PCA.



(a) Heuristic Miner (b) Inductive Miner Infrequent

Fig. 4. Aggregated conformance measures by number of clusters for two different process discovery algorithms

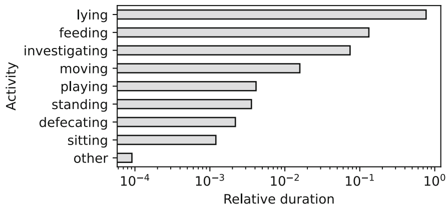
5.3 Result Stability

We assessed the stability of our approach with conformance checking. In particular, we evaluated if our approach was able to repeatably produce PM-compliant event logs. Generally, PM methods assume that all events and cases of an event log refer to the same notion of abstract tasks of the same process [1]. By partitioning the extracted event log into two distinct sub-logs (i.e., a training sub-log used for process discovery, and a validation sub-log to evaluate the discovered process models with conformance checking) conformance checking can be used to evaluate whether events with the same activity refer to the same notion of abstract steps in a process across the complete event log. If this assumption is not fulfilled, process discovery and conformance checking cannot produce reliable results. Instead, analysis results and quality measures would vary unpredictably

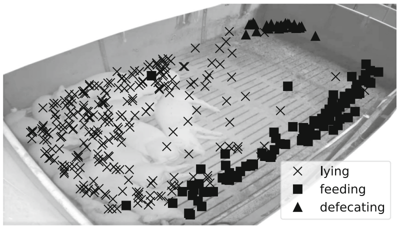
between multiple executions of the analytics pipeline on the same dataset and different partitionings of the same event log. Conversely, if quality measures are stable, it is demonstrated that the pipeline reliably extracts PM-compliant event logs.
To ensure replicability, 10-fold cross-validation was applied for splitting, which itself was repeated 10 times, resulting in a total of 100 runs (see (i) in Fig. 3). For each run, first, the cases of the training sub-log were clustered with k-means (ii). The cases of the validation sub-log were then assigned to the same clusters (iii), and a separate process model was discovered for each cluster of the training sub-log (iv). Fitness, precision, F1-score, and simplicity were then calculated for each cluster, using the cases of the validation sub-log assigned to the clusters accordingly (v). Quality measures for the complete event log were constructed by combining the measures of each cluster using a harmonic mean weighted by the number of cases in each cluster, which were then averaged over all runs. This evaluation scheme was repeated with different cluster configurations of k-means and using both the heuristic miner and inductive miner infrequent algorithms, for which the results are summarized in Fig. 4. Fitness, precision, and the F1-score were largely stable across runs with the same parameters, and their average values are acceptable considering that the analyzed process was inherently unstructured. As expected, model simplicity correlates with the number of clusters. For five or more clusters, the heuristic miner outperformed the inductive miner in F1-score, precision, and simplicity. In summary, the evaluation of result stability shows that the pipeline repeatably extracts activities and cases that are similar over the complete event log, and organizes the cases into behaviorally homogeneous clusters.

5.4 Result Meaningfulness

The approach was further evaluated by analyzing whether the extracted activities, cases, and clusters were *meaningful* with respect to the existing knowledge in the domain (of agriculture). This was evaluated by comparing the event log and process models extracted for one exemplary run of the analytics pipeline with domain knowledge (i.e., confirming the results through domain experts). We organized the resulting event log of this run into 15 clusters.
First, we analyzed the temporal and spatial distribution of the activities. The relative duration of the detected activities (see Fig. 5a) matches the expected ranges. For instance, 75% of all activities are *lying*, which is within the expected range of 60–85%, and the occurrence of *feeding* (13%) is largely similar to the analysis conducted in [2]. The spatial distribution of activities associated with the three functional areas (*lying*, *feeding* and *defecating*) is shown in Fig. 5b. The positions of these three activities reflect three largely separate clusters. *Feeding* was correctly detected among the feeding area, and *defecating* was localized in one corner of the pigpen and separated from the *lying* area. A manual inspection of the selected videos confirmed that these clusters match the actual functional areas commonly used by pigs.



(a) Relative duration of detected activities (total duration an activity is detected over total duration of all activities, logarithmic x-axis)



(b) Positions of a sample of detected *lying, feeding* and *defecating* activities

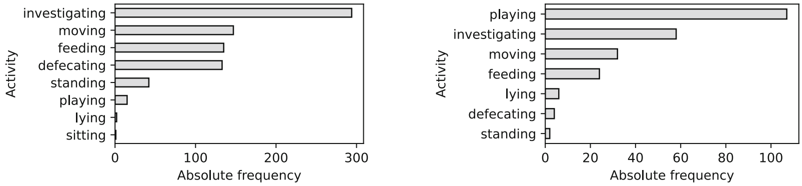
Fig. 5. Spatial and temporal distribution of detected activities

We then evaluated the cases and clusters of the event log, specifically if each cluster contained a set of similar cases of a specific behavioral pattern. The occurrences of activities in two clusters are shown in Fig. 6. Usually, two to four specific activities could define over 90% of all the events in a cluster. For instance, the cluster that was mostly composed of *moving*, *feeding* and *investigating* contained a behavioral pattern that connects these three activities. This implies that the clusters contain specific behavioral patterns. To analyze if the behavioral patterns are meaningful, a process model was discovered for each cluster using Disco. For instance, Fig. 7a shows an excerpt from the process model of a cluster mostly containing *moving*, *standing*, and *investigating*. In this pattern, pigs start with *moving* to a location, and then *investigate* their surroundings with intermittent *standing* pauses, which is indicated by a loop between these two activities. In the model of the cluster that contains 82% of all observations of pigs *defecating* (Fig. 7b), a loop exists between *defecating* and *investigating* (i.e., they are often executed in sequence). This order of activities corresponds to current domain knowledge on pig behavior, as pigs are known to typically investigate their surroundings before and after defecation.

The findings of our analytics pipeline were discussed with two domain experts from agricultural science, who confirmed that the extracted event log and behavioral patterns were meaningful with respect to the knowledge in the domain.

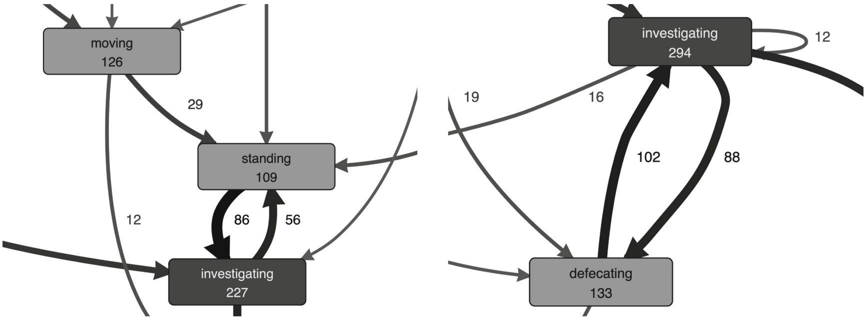
5.5 Reproducibility and Data Availability

Due to copyright restrictions, we are unable to publish the full recorded video dataset. However, we provide the detected bounding boxes of objects, tracking information, and the recognized activities for the videos (see [18]) allowing reproducibility. This also includes the training datasets for object detection and spatio-temporal action detection as well as the trained deep learning models. Also, the implementation includes the extracted event log, scripts to prepare the event log from the recognized activities and reproduce the results, as well as all process models discovered for the evaluation of event log meaningfulness.



(a) Cluster 10 (b) Cluster 11

Fig. 6. Occurrence of activities in two clusters.



(a) Cluster 9 (b) Cluster 10

Fig. 7. Excerpts from process models for two clusters generated using Disco

6 Related Work

In recent years, there has been a growing interest in exploring unstructured data for PM. Thus, many approaches have been proposed for various types of data sources. For instance, Janssen et al. [10] proposed an approach to extract event logs from smart home/IoT motion sensor data. The log of sensor activations is divided into sequential sections, which are clustered into patterns of similar sensor activation sequences. Then, the clusters are mapped to activities, and the activities are grouped into cases by assuming that cases are started and ended by specific sensors. Process discovery is applied to the extracted event log to discover models of human daily routines. Rebmann et al. [25] presented an approach to PM using time series sourced from motion sensors worn by workers. The time series are segmented into fixed-width sub-series, and supervised activity recognition is performed for each sub-series. If an activity cannot be classified from just sensor data, image data is used for disambiguation. An event log is constructed from the recognized activities and used for process discovery. Koschmider et al. [13] proposed an abstract method to extract event logs from general time series data. Similarly to Janssen et al. [10], time series are split into sub-series, similar sub-series are identified using clustering, and the resulting clusters are then mapped to process activities.

In contrast, research on PM specifically using video data is still at an early stage, with the few identified approaches published very recently (i.e., since 2020). Lepsien et al. [17] designed an abstract pipeline for PM on video data. The pipeline outlines the steps from dataset preparation to event log construction and the application of PM to this event log. The steps of the pipeline include the extraction of relevant video data, object and activity recognition, and process discovery. The pipeline also includes the explicit extraction of domain-specific knowledge and a refinement step. The contribution is limited to the abstract approach, and neither an implementation nor evaluation of the steps after pre-processing is provided. Knoch et al. [11] presented an unsupervised approach to process discovery from video recordings of manual assembly tasks. Process activities are recognized in the video recordings from overhead cameras mounted at specially designed assembly workstations by (1) tracking workers' hands throughout videos, (2) clustering hand trajectories and (3) assigning the trajectory clusters to work steps using the location of assembly parts on the workstation. As the clusters are directly assigned to pre-defined work steps, this approach is unable to discover activities not known a priori. Kratsch et al. [14] presented a reference architecture for PM on video data, outlining the steps from raw video data to event logs. They described a selection of different computer vision techniques that can be applied to extract information from video data, and provided guidance on choosing appropriate techniques for specific video PM use cases. The correlation of activity instances to cases is not included in the architecture but needs to be done externally. Further, the authors point out the limitation that their prototypical implementation and evaluation took place in a rather structured process context, and evaluation of more complex (i.e., less structured and more chaotic) processes would be beneficial.

7 Conclusion

In this paper, we presented a pipeline for extracting PM-compliant event logs and process models from unstructured video data. Structure is imposed into this unstructured data step by step, by extracting low-level events from the preprocessed video dataset using spatio-temporal action detection, and raising these events to a higher abstraction level using event abstraction, case correlation, trace clustering, and finally process discovery. We demonstrated the efficiency of our approach by implementing a modular framework that can easily be configured for application on video datasets from different domains and applying the implementation to surveillance footage of fattening pigs. The evaluation indicates that our approach extracts meaningful event logs in a reproducible manner. A review of related literature shows that, to the best of our knowledge, our analytics pipeline is the first fully validated approach that addresses the challenging task of analyzing an unstructured process through unstructured video data.

While evaluating our approach, we identified several limitations that need to be addressed in the future. Firstly, the requirement of supervised activity recognition to pre-define the activities to be detected may hinder the analysis of processes where limited information is available before the analysis. This could be addressed by integrating unsupervised activity recognition into the pipeline, which would also require advanced event abstraction algorithms capable of conceptually abstracting low-level to high-level activities [32]. Secondly, the evaluation confirmed that the pipeline can successfully be instantiated for a real-world application and produces stable and meaningful results, but the benefits added for end users were not quantified. We plan to evaluate the benefits added with user studies. Thirdly, expertise in deep learning is required to prepare the models required for *Object Tracking* and *Activity Recognition*. In the future, we plan to provide an interface to simplify the labeling of datasets and training of the deep learning models. Fourthly, while the evaluation confirmed the general efficiency and validity of our approach, the evaluation on a single use case does not enable conclusions about the generalization of our approach. To address this, a reference dataset and evaluation scheme would be beneficial. However, compiling a dataset that is large and heterogeneous enough for this purpose is highly time-consuming. This could be solved by synthetic evaluation data, which has already been proposed for other types of unstructured data in PM [34]. Fifthly, the current implementation is limited to settings where all process activities are visible from a single camera perspective (i.e., are executed in a single, constrained area) and actors can be tracked without interruption. A solution addressing this limitation would require the implementation of multi-camera tracking or object re-identification [14] to handle actors moving between areas observed by different camera perspectives, and improved event abstraction and case correlation techniques. Finally, the current methods implemented for *Case Correlation* limit the approach to actor-centric processes (objects performing activities). Implementing advanced case correlation methods would enable the analysis of processes that can be characterized with the more general notion of subjects performing activities on objects. Further possibilities to extend the applicability of the pipeline include privacy-preserving analysis techniques to address regulatory limitations [14,20], and techniques to propagate uncertainty (e.g., from *Activity Recognition*) through the pipeline (e.g., [16,24]) to quantify the result confidence.

In conclusion, our work provides a promising approach to integrating unstructured data sources into PM pipelines.

References

1. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_19
2. Bergamini, L., et al.: Extracting accurate long-term behavior changes from a large pig dataset. In: VISAPP 2021, vol. 4, pp. 524–533. SCITEPRESS, Online Streaming (2021). https://doi.org/10.5220/0010288405240533

3. Berti, A., van Zelst, S.J., van der Aalst, W.: Process Mining for Python (PM4Py): bridging the gap between process- and data science (2019). https://doi.org/10.48550/arXiv.1905.06169
4. Cao, J., Weng, X., Khirodkar, R., Pang, J., Kitani, K.: Observation-centric sort: rethinking sort for robust multi-object tracking. arXiv preprint (2022). https://doi.org/10.48550/arXiv.2203.14360
5. Chen, C., Zhu, W., Norton, T.: Behaviour recognition of pigs and cattle: journey from computer vision to deep learning. Comput. Electron. Agric. **187**, 106255 (2021). https://doi.org/10.1016/j.compag.2021.106255
6. Diamantini, C., Genga, L., Potena, D.: Behavioral process mining for unstructured processes. J. Intell. Inf. Syst. **47**(1), 5–32 (2016). https://doi.org/10.1007/s10844-016-0394-7
7. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. WIREs Data Min. Knowl. Discov. **10**(3), e1346 (2020). https://doi.org/10.1002/widm.1346
8. Feichtenhofer, C., Fan, H., Malik, J., He, K.: SlowFast networks for video recognition. In: ICCV 2019, pp. 6201–6210. Seoul, Korea (South) (2019). https://doi.org/10.1109/ICCV.2019.00630
9. Gu, C., et al.: AVA: a video dataset of spatio-temporally localized atomic visual actions. In: CVPR 2018, pp. 6047–6056 (2018). https://doi.org/10.1109/CVPR.2018.00633
10. Janssen, D., Mannhardt, F., Koschmider, A., van Zelst, S.J.: Process model discovery from sensor event data. In: Leemans, S., Leopold, H. (eds.) ICPM 2020. LNBIP, vol. 406, pp. 69–81. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72693-5_6
11. Knoch, S., Ponpathirkoottam, S., Schwartz, T.: Video-to-Model: unsupervised trace extraction from videos for process discovery and conformance checking in manual assembly. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 291–308. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_17
12. 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
13. Koschmider, A., Oppelt, N., Hundsdörfer, M.: Confidence-driven communication of process mining on time series. Informatik Spektrum **45**(4), 223–228 (2022). https://doi.org/10.1007/s00287-022-01470-3
14. Kratsch, W., König, F., Röglinger, M.: Shedding light on blind spots - developing a reference architecture to leverage video data for process mining. Decis. Support Syst. **158**, 113794 (2022). https://doi.org/10.1016/j.dss.2022.113794
15. 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
16. Lepsien, A.: Quantifying uncertainty for explainable process mining. In: Proceedings of the 13th International Workshop on Enterprise Modeling and Information Systems Architectures (EMISA 2023). CEUR Workshop Proceedings, vol. 3397. CEUR-WS.org, Stockholm, Sweden (May 2023). https://ceur-ws.org/Vol-3397/
17. Lepsien, A., Bosselmann, J., Melfsen, A., Koschmider, A.: Process mining on video data. In: Manner, J., Lübke, D., Haarmann, S., Kolb, S., Herzberg, N., Kopp, O.

(eds.) ZEUS 2022. CEUR Workshop Proceedings, vol. 3113, pp. 56–62. CEUR-WS.org, Bamberg, Germany (2022). https://ceur-ws.org/Vol-3113/paper9.pdf
18. Lepsien, A., Koschmider, A., Kratsch, W.: Video process mining evaluation data (2023). https://doi.org/10.5281/zenodo.7763839
19. Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Kim, T.K.: Multiple object tracking: a literature review. Artif. Intell. **293**, 103448 (2021). https://doi.org/10.1016/j.artint.2020.103448
20. Mannhardt, F., Koschmider, A., Baracaldo, N., Weidlich, M., Michael, J.: Privacy-preserving process mining. Bus. Inf. Syst. Eng. **61**(5), 595–614 (2019). https://doi.org/10.1007/s12599-019-00613-3
21. MMAction2 Contributors: openmmlab's next generation video understanding toolbox oand benchmark. https://github.com/open-mmlab/mmaction2 (2020)
22. Nannoni, E., Aarnink, A.J.A., Vermeer, H.M., Reimert, I., Fels, M., Bracke, M.B.M.: Soiling of Pig Pens: a review of eliminative behaviour. Animals **10**(11), 2025 (2020). https://doi.org/10.3390/ani10112025
23. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
24. Pegoraro, M., Uysal, M.S., van der Aalst, W.M.P.: Efficient construction of behavior graphs for uncertain event data. In: Abramowicz, W., Klein, G. (eds.) BIS 2020. LNBIP, vol. 389, pp. 76–88. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53337-3_6
25. Rebmann, A., Emrich, A., Fettke, P.: Enabling the discovery of manual processes using a multi-modal activity recognition approach. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) BPM 2019. LNBIP, vol. 362, pp. 130–141. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37453-2_12
26. Shermin, T., Teng, S.W., Murshed, M., Lu, G., Sohel, F., Paul, M.: Enhanced transfer learning with imagenet trained classification layer. In: Lee, C., Su, Z., Sugimoto, A. (eds.) PSIVT 2019. LNCS, vol. 11854, pp. 142–155. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34879-3_12
27. Wada, K.: LabelMe: image polygonal annotation with Python (2023). https://doi.org/10.5281/zenodo.5711226, https://github.com/wkentaro/labelme
28. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors (2022). https://doi.org/10.48550/arXiv.2207.02696
29. Weijters, A., van Der Aalst, W.M., De Medeiros, A.A.: Process mining with the heuristics miner-algorithm. TU Eindhoven, Tech. Rep. WP 166(July 2017), 1–34 (2006)
30. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)
31. Zandkarimi, F., Rehse, J.R., Soudmand, P., Hoehle, H.: A generic framework for trace clustering in process mining. In: ICPM 2020, pp. 177–184. IEEE, Padua, Italy (2020). https://doi.org/10.1109/ICPM49681.2020.00034
32. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. Granular Comput. **6**(3), 719–736 (2020). https://doi.org/10.1007/s41066-020-00226-2
33. Zhang, Y., et al.: ByteTrack: multi-object tracking by associating every detection box. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022, pp. 1–21. LNCS, Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20047-2_1

34. Zisgen, Y., Janssen, D., Koschmider, A.: Generating synthetic sensor event logs for process mining. In: De Weerdt, J., Polyvyanyy, A. (eds.) CAiSE Forum 2022. LNBIP, vol. 452, pp. 130–137. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-07481-3_15
35. Zoric, M., Johansson, S.E., Wallgren, P.: Behaviour of fattening pigs fed with liquid feed and dry feed. Porc. Health Manag. **1**(1), 14 (2015). https://doi.org/10.1186/s40813-015-0009-7

An SQL-Based Declarative Process Mining Framework for Analyzing Process Data Stored in Relational Databases

Francesco Riva[1,2,3](✉), Dario Benvenuti[4], Fabrizio Maria Maggi[1], Andrea Marrella[4], and Marco Montali[1]

[1] Free University of Bozen-Bolzano, Bolzano, Italy
francesco.riva@unibz.it, {maggi,montali}@inf.unibz.it,
[2] University of Tartu, Tartu, Estonia
[3] Datalane SRL, Verona, Italy
[4] Sapienza University of Rome, Rome, Italy
{d.benvenuti,marrella}@diag.uniroma1.it

Abstract. Recently, the idea of applying process data analysis over relational databases (DBs) has been investigated in the process mining field resulting into different DB schemas that can be used to effectively store process data coming from Process-Aware Information Systems (PAISs). However, although SQL queries are particularly suitable to check declarative rules over traces stored in a DB, a deep analysis of how the existing instruments for SQL-based process mining can be effectively used for process analysis tasks based on declarative process modeling languages is still missing. In this paper, we present a full-fledged framework based on SQL queries over relational DBs for different declarative process mining use cases, i.e., process discovery, conformance checking, and query checking. The framework is used to benchmark different SQL-based solutions for declarative process mining, using synthetic and real-life event logs, with the aim of exploring their strengths and weaknesses.

Keywords: Process Discovery · Conformance Checking · Query Checking · Declarative Process Model · SQL · Relational Database

1 Introduction

The process data recorded by Process-Aware Information Systems (PAISs) is usually stored in multiple and often heterogeneous relational databases (DBs). Several efforts have been done, in the past, in order to solve the data integration

The work of F. Riva was funded by the PRISMA project of the Free University of Bozen-Bolzano and by the Estonian Research Council (PRG1226). The work of D. Benvenuti and A. Marrella was supported by the PNRR MUR project PE0000013-FAIR, the H2020 project DataCloud (Grant number 101016835), and the Sapienza project DISPIPE.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 214–231, 2023.
https://doi.org/10.1007/978-3-031-41623-1_13

problem [4], but also in order to store and query process data in relational DBs in an effective and efficient manner [13,31].

In recent years, the process mining community has investigated how DB theory methods can be used to carry on process analysis on the process behavior recorded in a relational DB. Different DB schemas have been developed [13,31], which are suitable to effectively store process data into a DB.

SQL queries are particularly suitable for process mining based on declarative languages (such as Declare [23], DPIL [33], or DCR Graphs [16]) since it is possible to build a 1-to-1 mapping between the SQL queries and the temporal rules that need to be checked over traces stored in a DB. Although some works have already investigated how to discover Declare rules from a DB using SQL queries [27–29], a full-fledged framework to support the event log storage in a DB and the execution of queries that can be used to support the entire spectrum of declarative process mining use cases is still missing.

In this exploratory paper, we introduce such a framework and we use it to provide a deep analysis of strengths and weaknesses of different SQL-based solutions for declarative process mining. The framework is readily available[1] for researchers and practitioners that need to analyze process data stored in relational DBs.

The paper is structured as follows. Section 2 presents the research problem. Section 3 discusses related work. Section 4 introduces the proposed framework, and discusses the DB schemas and the SQL queries supported. In Sect. 5, the framework is used to benchmark different SQL-based solutions for declarative process mining using synthetic and real-life logs. Section 6 concludes the paper and spells out directions for future work.

2 Research Problem

In this paper, we present an SQL-based framework for declarative process mining and use it to benchmark different SQL-based solutions for declarative process mining. Through the framework, we answer the following research questions:

- **RQ1**: What is the most efficient DB schema in terms of required disk space and population time?
- **RQ2**: What is the most efficient DB schema in terms of query execution time?
- **RQ3**: How does the query execution time vary for datasets with different characteristics?
- **RQ4**: How does the query execution time vary for different types of queries?

RQ1 and **RQ2** aim at understanding which one of the DB schemas existing in the literature has the highest performance in terms of population time, required disk space, and query execution time. To answer these research questions, we also test how some improvements over the existing schemas can increase

[1] https://github.com/francxx96/XEStoDB.

the DB performance. To answer **RQ3**, we show how the query execution time varies using synthetic logs with different characteristics. The query execution time measured for answering **RQ2** and **RQ3** concerns the discovery task. **RQ4** investigates, instead, the query execution time needed to run all the different types of queries provided in the proposed framework.

3 Related Work

The literature on declarative process mining covers a wide range of process mining use cases [21]. In this paper, we solve standard process mining tasks like process discovery (cf. [27,28]), conformance checking (cf. [3]), and query checking (cf. [26]), and we extend them with novel types of analysis that can be easily tackled using queries like instance-spanning process analysis [1], metric temporal rule discovery and checking [20], and local rule checking (i.e., the verification of rules in specific time intervals).

An approach for process discovery similar to the SQL-based one used in this paper is presented in [27,28]. Here, a sub-set of the standard Declare [24] templates (i.e., parameterized temporal rules) is used to define SQL queries that can be used to discover Declare models. Further investigation [29] led to the introduction of a set of queries for the discovery of Multi-Perspective Declare (MP-Declare). Other techniques (that are not based on SQL) for performing declarative process mining are available in state-of-the art process mining toolkits like RuM [2] and Declare4Py [8]. However, in order to use these tools for process analysis, the source data must be first extracted from the PAISs and then arranged in XES files.

In [30], the authors present an investigation that shows that it is possible to make the SQL queries for Declare discovery faster by using DB indexing. As mentioned in Sect. 4, the DB indexing analysis provided in [30] supports the way we designed our queries.

4 SQL-Based Declarative Process Mining Framework

In this paper, we present a full-fledged framework to perform different process mining tasks using SQL queries over relational DBs of PAISs. Figure 1 presents the conceptual overview of the framework. The framework supports two phases of the process data analysis with relational DBs.

Database Creation. A new relational DB is created following a DB schema given as input, the DB is then populated according to an input event log.

SQL-Based Declarative Process Mining. In this phase, the user chooses the process mining (PM) task to perform (i.e., discovery, query checking or conformance checking) and the query type, i.e., one of the task variants that will be introduced in Sect. 4.2. For conformance checking, the Declare model to be checked

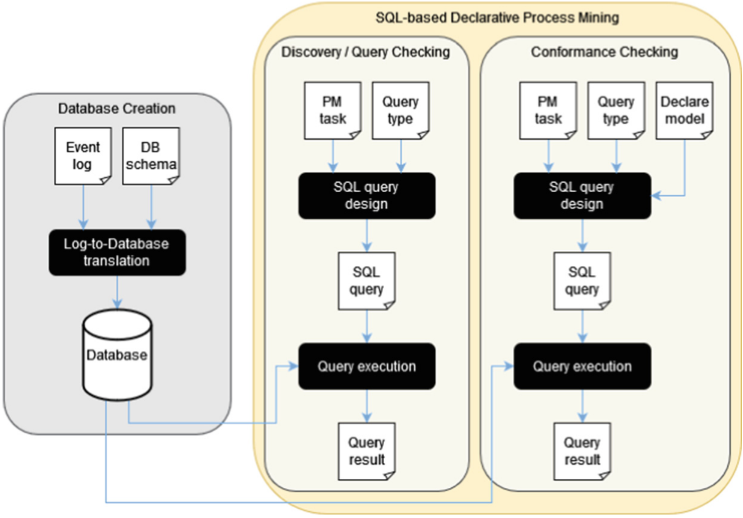


Fig. 1. Conceptual overview of the developed framework.

has to be provided by the user as well. The corresponding SQL query is designed based on the inputs. The query is then executed on the DB.

To support these two phases of the process data analysis, we implemented a wide range of queries. Some of these queries implement different declarative process mining tasks and support the *SQL-Based Declarative Process Mining* phase, others support the *Database Creation* phase using different DB schemas.

4.1 Database Creation

In the literature, different DB schemas have been proposed to store process data with the aim of minimizing both population time and disk space required to store the data. The DB schemas proposed are all fully compatible with the XES standard [32]. We selected four different relational DB schemas to be compared:

– Monolithic, composed of a single table in which each row represents an event of the log;
– DBXES, presented in [31];
– RXES, presented in [13];
– RXES+, which is an adaptation of RXES (see Fig. 2).

The design of RXES+ was intended to optimize not only the population time and the disk space needed to store the process data, but also the query execution time for process mining tasks.

Differently from DBXES and RXES, in RXES+, the *log*, *trace* and *event* tables include the mandatory XES attributes (i.e., *name*, *timestamp*, and *life-cycle transition*), so that we significantly reduce the amount of repetitions in tables $\{log|trace|event\}\_has\_attribute$ and *attribute*. Moreover, RXES is defined

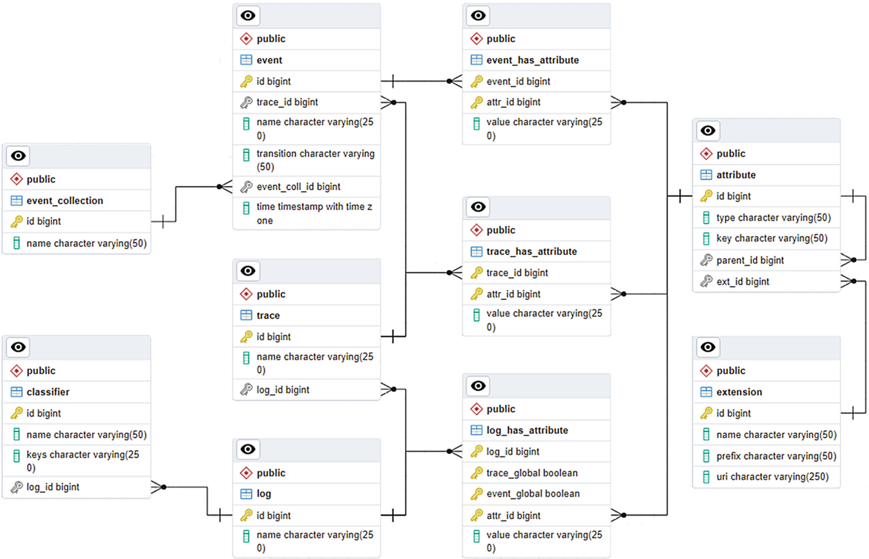


Fig. 2. RXES+ schema.

in a way that it eliminates the duplicate traces/events inside a log, i.e., when populating the DB, if an event or a trace is exactly the same as another event or trace already present in the DB, this element is not repeated but linked to the existing one. However, checking the existence of an event or a trace in the DB is an unnecessarily heavy task. Indeed, from an analysis of the logs existing in the literature (see Table 1), we can see that, since events are always provided with a timestamp, the occurrence of duplicate traces/events is extremely rare (this only happens if multiple events with exactly the same attributes occur exactly at the same time). For this reason, RXES+ has been designed to allow duplicates. As we can see from our experiments, this does not significantly affect the DB size and the query execution time but dramatically reduces the population time.

4.2 SQL-Based Declarative Process Mining

SQL queries are a powerful instrument for implementing in a straightforward way a large range of analysis types over process data recorded in a DB. Here, we present the query types available in our framework. All the queries use a temporary table *@event*, which allows the queries to be formulated exactly in the same way independently of the schema. The temporary table *@event* is built with a different query depending on the schema. For example, for the RXES+ schema *@event* is built with the query:

```
DECLARE @event TABLE ( log_id BIGINT, trace_id BIGINT,
    task VARCHAR(300), [timestamp] DATETIME2(3) )

INSERT @event
```

```
SELECT t.log_id, e.trace_id, e.name + '_' + e.
    transition,
     e.[timestamp]
FROM trace t JOIN event e ON t.id = e.trace_id;
```

Discovery. For process discovery, i.e., for the identification of a set of rules (based on a Declare template specified by the user) satisfied with a minimum support[2] in an input log, we follow the approach introduced in [27,28], where, first, the input template is instantiated into different candidate constraints (obtained by replacing the template parameters with all the possible combinations of activities available in the log), and, then, the candidate constraints are checked to compute their support.

In particular, for each SQL query defined from now on, the *Support* is computed as follows:

```
CAST(COUNT(*) AS FLOAT) / CAST( (SELECT COUNT(*) FROM
    @event WHERE task='TaskA') AS FLOAT)
```

Starting from the queries presented in [27,28] - the Baseline (BS) query set - developed for the discovery of standard Declare rules, we designed a new query set - the Join query set - in order to improve the query execution time for discovery. In particular, the new query set was designed with the aim of reducing possible performance bottlenecks; in order to achieve this, we used the query plan [14] produced when executing each query. From the plans, we noticed that, before running any queries, the DBMS always sorts the events in the DB when the queries contain explicit JOIN statements. This conclusion is similar to the one found in [30], where a systematic DB indexing analysis is conducted. Therefore, we re-designed the queries to benefit of this automatic DB indexing executed by the DBMS.

Example 1. In Declare, the *response* template instantiated with activation A and target B indicates that, when activity A is executed, it must be eventually followed by B. The discovery from an input log of rules of type *response* can be obtained with the following query relying on explicit JOIN statements:

```
SELECT 'Response', TaskA, TaskB, Support
FROM (
    SELECT a.trace_id, a.task AS TaskA, b.task AS TaskB
    FROM @event a JOIN @event b ON (
        a.log_id = b.log_id
        AND a.trace_id = b.trace_id
        AND a.task != b.task
        AND a.[timestamp] < b.[timestamp]
    ) GROUP BY a.trace_id, a.task, a.[timestamp], b.
        task
) subquery
GROUP BY TaskA, TaskB;
```

[2] Here the support corresponds to the *event support* introduced in [7].

Conformance Checking. For conformance checking, we follow the approach introduced in [3]. The input here is not a generic template but a Declare model, i.e., a set of concrete rules, which are instantiations of templates with real activities. The outcome is the support in the log of each rule in the model.

Example 2. The conformance checking of a rule of type *response* instantiated over activities *Receive Payment* and *Send Receipt* wrt. an input log is obtained from the one seen in Ex1 by changing line 7 as follows:

```
7        AND a.task='Receive Payment' AND b.task='Send Receipt'
```

Query Checking. This type of analysis was first introduced in [26]. The input here is a partial instantiation of a template, i.e., a template where only one of the parameters is replaced with a real activity, while the other one remains unspecified. In addition, a minimum support is also specified. The outcome is the discovery from an input log of rules of the specified format, and satisfied in the log with the specified minimum support.

Example 3. The query checking of a rule of type *response* instantiated with activation *Receive Payment* and target left unspecified is obtained from the one seen in Example 1 by changing line 7 as follows:

```
7        AND a.task='Receive Payment' AND b.task!=a.task
```

Additional types of analysis The standard approaches for discovery, conformance, and query checking just introduced can be extended using variants of the standard queries, which provide facilities to solve well-known problems in declarative process mining, such as:

- Instance-Spanning analysis (IS) [1], which considers the whole log as a single trace obtained by ordering the events by timestamp;
- Local Rule Checking (LRC), which checks the validity of a rule only within a given time interval;
- Metric Temporal rule discovery (MT) [20], which enriches the query for the discovery of standard Declare rules with information about the minimum/average/maximum temporal distance between activation and target activities;
- Validity Intervals analysis (VAL), which finds the time intervals in a trace in which a Declare rule is valid;
- Attribute Range (RNG) analysis, which finds for a given attribute the range of values it gets in a given time interval.

These variants can also be easily combined together to build custom queries that are useful for a particular need of the end user, e.g., it is possible to combine IS with LRC in order to have an instance-spanning query restricted to a given time interval.

Example 4. The following query implements the Instance-Spanning discovery of *response* rules:

```
SELECT 'Response', TaskA, TaskB, Support
FROM (
    SELECT a.trace_id, a.task AS TaskA, b.task AS TaskB
    FROM @event a JOIN @event b ON (
        a.log_id = b.log_id
        AND a.task != b.task
        AND a.[timestamp] < b.[timestamp]
    ) GROUP BY a.trace_id, a.task, a.[timestamp], b.task
) subquery
GROUP BY TaskA, TaskB;
```

The following query implements the Local Rule Checking of the *response* template instantiated over activities *Receive Payment* and *Send Receipt* in the time interval spanning from 2020-01-01 00:00:00.000 to 2021-12-31 23:59:59.999:

```
DECLARE @interval_start DATETIME2(3)='2020-01-01 00:00:00.000',
    @interval_end DATETIME2(3)='2021-12-31 23:59:59.999';

SELECT 'Response', TaskA, TaskB, Support
FROM (
    SELECT a.trace_id, a.task AS TaskA, b.task AS TaskB
    FROM @event a JOIN @event b ON (
        a.log_id = b.log_id AND a.trace_id = b.trace_id
        AND a.task = 'Receive Payment' AND b.task = 'Send Receipt'
        AND a.[timestamp] < b.[timestamp]
    ) WHERE a.[timestamp] >= @interval_start
        AND a.[timestamp] < @interval_end
        AND b.[timestamp] >= @interval_start
        AND b.[timestamp] < @interval_end
    GROUP BY a.trace_id, a.task, a.[timestamp], b.task
) subquery
GROUP BY TaskA, TaskB;
```

The following query implements the Metric Temporal discovery of *response* rules:

```
SELECT 'Response', TaskA, TaskB, Support,
    MIN(TD) AS min_TD, AVG(TD) AS avg_TD, MAX(TD) AS max_TD
FROM (
    SELECT a.trace_id, a.task AS TaskA, b.task AS TaskB,
        MIN(DATEDIFF(SECOND, a.[timestamp], b.[timestamp])) AS TD
    FROM @event a JOIN @event b ON (
```

```
            a.log_id = b.log_id AND a.trace_id = b.trace_id
            AND a.task != b.task AND a.[timestamp] < b.[timestamp]
        ) GROUP BY a.trace_id, a.task, a.[timestamp], b.task
) subquery
GROUP BY TaskA, TaskB;
```

For space limitations, we do not report here the queries returning the Validity Intervals of a rule and the Attribute Range of an attribute. The interested reader can find the details about these queries at https://github.com/francxx96/XEStoDB.

5 Benchmarks

To answer the research questions introduced in Sect. 2, we performed experiments on synthetic and real-life logs. In the following sections, we first describe the experimental setting, i.e., describe the logs and metrics used in the experimentation, then, we discuss the experimental results to answer the research questions.

5.1 Experimental Setting

As already mentioned, we validated our SQL-based framework by considering both synthetic and real-life logs. With the real-life logs, we wanted to demonstrate the applicability of the framework to well-known benchmarks in the process mining field. In particular, we considered six logs, most of them presented in past editions of the Business Process Intelligence Challenge (BPIC):

- SEPSIS, recording the treatment of incoming patients with sepsis in a hospital [22];
- ROAD, related to a road traffic fines management process [18];
- FINANC, pertaining to a loan application process (provided for the BPIC 2012) [9];
- LOAN, a richer version of FINANC (provided for the BPIC 2017) [10].
- REIMB, pertaining to a reimbursement process for international declarations (provided for the BPIC 2020) [11];
- TRAVEL, related to the management of travel permits (provided for the BPIC 2020) [12];

Table 1 reports the characteristics of the real-life logs. These logs are widely heterogeneous ranging from simple to very complex, with a log size ranging from 1,050 traces (for the SEPSIS log) to 150,370 traces (for the ROAD log). A similar variety can be observed in the number of event classes (i.e., activities executed in the log), ranging from 11 to 51. Moreover, the trace length also varies from very short traces (containing only two events), to very long traces (containing 185 events). The table also shows the percentages of duplicate events in each log. In this respect, we can see that, except for the ROAD log, the percentage of duplicate events is always equal to zero or very close to it.

Table 1. Descriptive statistics of real-life logs.

Log name	Total traces	Total events	Event classes	Duplicate events	Trace length		
					min	avg	max
SEPSIS	1,050	15,214	16	<0.01%	3	14	185
ROAD	150,370	561,470	11	79.63%	2	4	20
FINANC	13,087	262,200	36	<0.01%	3	20	175
LOAN	31,509	1,160,405	26	0%	9	37	177
REIMB	6,449	72,151	34	0.04%	3	11	27
TRAVEL	7,065	86,581	51	0%	3	12	90

Synthetic logs were created using the *ASP log generator* [6] implemented in the declarative process mining tool RuM [2]. They are intended to prove the scalability of the presented framework wrt. logs with specific characteristics (i.e., number of distinct event classes, number of traces in the log, number of events in a trace) in a controlled environment. We built several different synthetic logs, each named using the format `clsXXXtrcXXXevtXXX`. For example, `cls10trc100evt30` identifies a log containing 10 different event classes and 100 traces, each including 30 events.

The performance metrics we considered in our experimentation to answer the research questions are:

- Required disk space to store a log in a DB, which is a measure of the DB redundancy degree;
- DB population time;
- Event insertion time, which measures the time needed for inserting an event in the DB (when measured for subsequent insertions, it might happen that the insertion time is higher for events inserted later in the DB);
- Query execution time.

All the experimental material can be found in the repository available at https://github.com/francxx96/XEStoDB, which contains:

- Translation scripts (Java 11) from XES-formatted files to each type of DB considered;
- SQL schemas reproducing the (empty) DBs;
- SQL dumps of the DBs already populated with the datasets used in our experiments;
- SQL queries for implementing all the declarative process mining task discussed in this paper.

We performed our experiments on a machine with an Intel Xeon E5-2690 CPU (dual core, 2.60 GHz), Windows Server 2019 OS and 16 GB RAM. The DBMS we used to define DB schemas and queries is Microsoft SQL Server 2019.

5.2 Results

RQ1. What Is the Most Efficient DB Schema in Terms of Required Disk Space and Population Time? To answer this research question, we created, for each considered real-life log, four DBs (one for each considered schema)

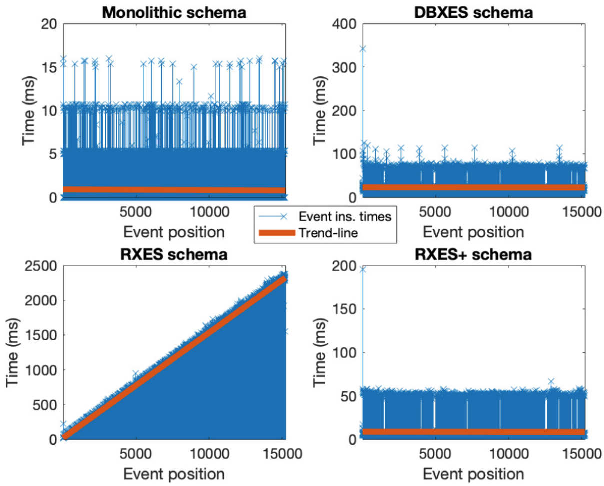


Fig. 3. Subsequent event insertion times for SEPSIS.

Table 2. Disk space for log storage and average population time.

DB schema	Disk space (MB)						Population time (s)					
	SEPSIS	ROAD	FINANC	LOAN	REIMB	TRAVEL	SEPSIS	ROAD	FINANC	LOAN	REIMB	TRAVEL
Monolithic	8.50	184.98	110.03	627.00	53.24	91.78	34.69	1,211.73	560.31	2,795.66	156.03	216.90
DBXES	7.63	180.30	73.93	–	43.55	49.13	364.09	15,551.52	14,440.95	–	6,034.75	7,303.98
RXES	8.50	–	–	–	–	–	18,060.88	–	–	–	–	–
RXES+	5.08	92.59	30.19	318.96	21.91	26.30	134.89	5,063.72	1,203.21	13,406.82	896.47	1,085.52

containing the process data of that log. A comparison in terms of required disk space needed to store the logs using the different DB schemas and their population times[3] (averaged over 5 runs) can be seen in Table 2. Here, we can observe that, as expected, the Monolithic schema has the highest degree of redundancy and occupies in all cases the highest amount of disk space (in some cases more than three times wrt. RXES+). This is a critical issue for large real-life logs. In addition, the large amount of disk space occupied, forces the Monolithic schema to have an upper-bound on the number of attributes that can be stored in the DB given by the limited number of columns admitted by the DBMSs for a single table. The RXES+ schema, instead, uses always less space than other schemas for all the analyzed logs. For what concerns the population time, the Monolithic schema is the fastest one. This is due to its simple structure (it is composed of a single table containing all the log data) that does not require to update DB-related constraints (e.g., foreign keys) when inserting a new event. The population time of RXES is extremely high since its structure requires to check the presence of duplicate events/traces at each insertion. This is confirmed by the more detailed analysis conducted on the SEPSIS log shown in Fig. 6 3. The plots in the figure indicate the time needed for subsequent event insertions

[3] We set a timeout on the population scripts and each script that did not end within 24 h was stopped. Dashes in the tables mean that the corresponding scripts reached the timeout.

Table 3. Query execution time comparison between the query sets.

DB schema	Average query time (s)													
	Response		Alternate Response		Chain Response		Precedence		Alternate Precedence		Chain Precedence		Responded Existence	
	BS	Join	BS	Join	BS	Join	BS	Join	BS	Join	BS	Join	BS	Join
SEPSIS														
Monolithic	0.585	0.506	1.111	0.587	1.215	0.427	0.564	0.543	1.405	0.538	1.595	0.429	1.193	0.975
DBXES	1.176	1.239	1.854	1.259	1.999	1.037	1.319	1.299	2.140	1.208	2.382	1.050	1.949	1.714
RXES	0.676	0.636	1.213	0.720	1.348	0.507	0.698	0.675	1.522	0.649	1.735	0.502	1.318	1.107
RXES+	0.575	0.524	1.105	0.602	1.225	0.452	0.561	0.547	1.394	0.542	1.575	0.447	1.173	0.990
ROAD														
Monolithic	9.40	8.04	25.78	8.31	31.43	4.91	9.06	7.86	25.97	8.06	31.87	4.76	22.78	14.55
DBXES	13.73	12.56	31.08	13.28	36.39	9.85	13.98	12.77	31.39	12.92	37.03	9.75	27.63	19.73
RXES+	9.44	8.09	26.43	8.44	31.74	5.02	9.28	7.90	26.53	8.12	32.44	4.89	22.93	14.57
FINANC														
Monolithic	36.72	32.57	39.99	22.17	57.59	15.39	36.26	33.58	44.83	18.54	74.54	15.56	60.86	68.38
DBXES	46.06	41.63	50.54	31.16	67.62	22.62	45.93	43.08	55.67	27.74	84.30	23.08	70.57	79.73
RXES+	37.25	32.41	40.87	21.75	57.21	15.65	36.45	33.51	45.66	18.41	74.64	16.20	60.13	68.09
LOAN														
Monolithic	542.76	202.87	355.58	166.87	566.16	93.29	533.45	207.00	369.44	143.76	663.95	96.67	843.42	424.74
RXES+	540.37	204.00	350.52	167.32	568.96	95.01	546.44	207.66	374.10	145.01	678.89	97.44	895.67	422.88
REIMB														
Monolithic	8.79	1.65	9.20	1.91	13.44	1.09	8.94	1.62	9.26	1.86	13.98	1.11	16.43	2.71
DBXES	13.20	5.49	14.94	6.38	20.42	4.69	13.47	6.18	15.15	6.50	20.77	4.75	22.49	8.60
RXES+	9.77	2.72	11.67	2.98	15.74	1.30	10.18	2.64	11.79	2.92	16.19	1.28	19.14	4.79
TRAVEL														
Monolithic	22.63	4.22	20.95	4.36	27.72	2.02	22.59	4.09	20.62	4.23	28.03	2.04	32.85	7.45
DBXES	25.60	7.79	24.57	8.63	32.10	6.09	26.75	8.41	24.72	8.64	33.15	6.20	37.34	12.02
RXES+	22.28	4.18	20.35	4.35	27.37	1.98	22.54	4.07	20.67	4.21	27.98	1.99	33.13	7.43

for the four considered schemas. The time required for subsequent event insertions for RXES grows linearly with the number of events already inserted, while it remains constant for the other schemas. RXES+ is faster than both DBXES and RXES. Overall, RXES+ guarantees a good trade-off between the disk space needed to store the process data and the population time.

RQ2. What Is the Most Efficient DB Schema in Terms of Query Execution Time? To answer this research question, we executed all the queries in the two discovery query sets BS and Join for all the considered real-life logs stored using all the considered DB schemas. Table 3 shows the query execution times measured by executing the BS and the Join query set. The gray background in the table indicates the schema that performed best on the same log and the same query. The query execution times displayed in the tables are averaged over 5 runs. The results highlight that the Monolithic and RXES+ schemas achieve very similar performance on both query sets, and perform better than DBXES and RXES. Almost all the queries in the Join set run faster than the baseline queries.

RQ3. How Does the Query Execution Time Vary for Datasets with Different Characteristics? To answer this research question, we performed a set of controlled experiments using synthetic logs generated by varying the number of event classes, the number of traces, and the number of events in each

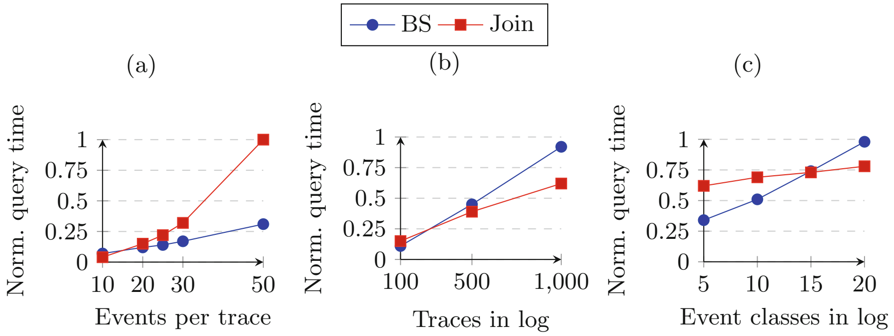


Fig. 4. Query execution time for Discovery task wrt. (4a) number of events per trace, (4b) number of traces in the log, (4c) number of event classes in the log.

trace. In particular, we generated three sets of synthetic logs; each set fixes two of the above parameters, while changing the remaining one.

For testing how the trace size affects the query execution time, we generated five synthetic logs containing 10 event classes and 100 traces of size 10, 20, 25, 30, and 50, respectively. For testing how the log size affects the query execution time, we generated three synthetic logs containing 10 event classes, traces of size 20 and with log size equal to 100, 500, and 1000, respectively. Finally, for testing how a different number of event classes affects the query execution time, we generated four synthetic logs with log and trace size equal to 100 and 20, respectively, and containing 5, 10, 15, and 20 event classes. The query execution time was, again, measured running all the queries in the two discovery query sets BS and Join and was normalized over the results obtained for all the queries in the two query sets, in order to have a single performance indicator for all queries in each set. We used the RXES+ schema for these experiments.

The results are shown in Fig. 4. In Fig. 4a, we can observe that the Join query set is more efficient when traces are shorter, while grows exponentially as the traces become longer. This is due to the structure of the Join queries, which leverages DB indexing (improving the performance for traces with less than 20 events) and a Cartesian product, containing all the possible combinations of the events in a trace, which becomes larger when the trace size increases. Figure 4b shows, instead, that the Join query set scales better than BS as the log size grows, and Fig. 4c shows how the query execution time for both BS and Join grows linearly when the number of event classes increases, with Join becoming more efficient for logs containing more than 15 event classes.

To sum up, for logs with more than 15 event classes and more than 500 traces each containing less than 20 events (which are common characteristics for many real-life logs) the queries in the Join set are more efficient. However, for traces particularly long, it is better to use queries that do not rely on explicit JOIN statements.

Table 4. Query execution time for each query type.

	Average query time (s)						
Query type	Response	Alternate Response	Chain Response	Precedence	Alternate Precedence	Chain Precedence	Responded Existence
SEPSIS							
STD	0.52	0.60	0.45	0.55	0.54	0.45	0.99
IS	471.55	491.96	820.21	605.46	473.83	814.45	0.95
LRC	0.65	0.86	0.74	0.65	0.83	0.76	1.03
MT	0.84	0.84	0.62	0.83	0.77	0.61	0.82
VAL	0.96	1.94	4.16	1.23	6.22	3.48	2.68
ROAD							
STD	8.09	8.44	5.02	7.90	8.12	4.89	14.57
IS	–	–	–	–	–	–	42.69
LRC	7.00	8.94	5.32	6.55	8.68	5.19	11.37
MT	0.84	8.71	5.19	8.25	8.47	5.07	8.40
VAL	0.96	19.66	11.99	11.51	20.02	12.12	26.72
FINANC							
STD	32.41	21.75	15.65	33.51	18.41	16.20	68.09
IS	–	–	–	–	–	–	65.38
LRC	22.13	23.97	16.52	23.32	20.67	17.18	41.76
MT	37.91	23.82	17.38	38.64	20.15	18.09	38.02
VAL	42.84	69.79	105.42	51.92	90.67	109.70	102.22
LOAN							
STD	204.00	167.32	95.01	207.66	145.01	97.44	422.88
IS	–	–	–	–	–	–	677.36
LRC	230.16	193.64	97.39	233.38	166.22	100.56	475.61
MT	242.90	181.59	104.70	249.01	157.58	106.87	249.84
VAL	322.73	434.47	636.92	363.25	482.77	658.32	1,032.00
REIMB							
STD	2.72	2.98	1.30	2.64	2.92	1.28	4.79
IS	–	–	–	–	–	–	15.49
LRC	3.08	3.77	2.09	3.12	3.82	2.21	5.44
MT	3.38	3.59	1.39	3.33	3.57	1.40	3.49
VAL	5.46	7.57	5.63	5.39	7.54	6.27	12.44
TRAVEL							
STD	4.18	4.35	1.98	4.07	4.21	1.99	7.43
IS	–	–	–	–	–	–	29.88
LRC	4.68	5.38	3.05	4.49	5.39	3.24	6.72
MT	5.08	5.16	2.13	4.91	4.90	2.16	4.99
VAL	7.64	10.76	9.15	7.86	11.11	10.44	17.25

Table 5. Query execution time for RNG query type.

Query type	Average query time (s)					
	SEPSIS	ROAD	FINANC	LOAN	REIMB	TRAVEL
RNG	0.14	0.18	0.09	14.25	1.23	0.70

RQ4. How Does the Query Execution Time Vary for Different Types of Queries? To answer this research question, we executed all the additional types of queries defined in Sect. 4 using RXES+ as schema on each log.[4]

Table 4 compares, for each Declare template, the execution time (averaged over 5 runs) of the standard queries in the Join query set (STD) with the IS, LRC, MT, VAL. Here, IS, LRC, and MT are standard discovery tasks, while VAL returns the validity intervals of all the Declare rules obtained by instantiating each template with all the combinations of event classes available in the log. Table 5 shows results for the RNG query execution time. RNG returns all the value ranges of all the attributes available in each log in a fixed time interval. The details about the queries used in this experiment can be found at https://github.com/francxx96/XEStoDB.

We have already seen from the experiments on the synthetic logs that the execution time of the queries in the Join set grows exponentially as the trace size grows. Since the instance-spanning queries consider the whole log as a single trace, the IS queries require much more time to be executed. To solve this issue, in our repository, we provide also for this type of queries a version that does not use explicit JOIN statements.

6 Conclusion

In this paper, we proposed an SQL-based framework for declarative process mining. We proposed different queries to support the three main use cases of declarative process mining, i.e., process discovery, conformance checking, and query checking. We also presented an extensive cross-benchmark comparison we conducted using several synthetic and real-life logs for investigating the performance of different DB schemas and different types of queries.

The evaluation has been conducted using Microsoft SQL Server 2019 as DBMS. Nonetheless, the conclusions drawn are valid in general. In particular, the improved performance of RXES+ in terms of disk space needed to store a log and the insights derived from the experiments on the DB population time are clearly valid independently of the DBMS used. In addition, the improvements in the query execution time obtained with the use of explicit JOIN statements are also valid in general provided that the queries are executed on an indexed DB. Another observation that is worth mentioning is that all the queries presented in this paper are easily applicable to any (proprietary) DB schema with the

[4] We set a timeout of 30 min on the query scripts.

only requirement that the DB contains timestamped events somehow grouped together (into traces), which is the most basic requirement needed to conduct any type of process mining analysis on a dataset (only the query for building the temporary table *@event* must be rewritten when using a new DB schema). Portability is, in general, a significant advantage of the proposed framework and this is the reason why we developed it by relying only on standard SQL clauses. Although other more sophisticated SQL clauses could be used to improve the overall performance of the framework, these solutions could affect its portability across different DBMSs.

We think that the investigations conducted in this paper can be considered as an important basis for researchers who want to develop techniques for process mining based on SQL since they give several insights about the main bottlenecks and possible issues that can come up when DBs are used for process analytics. This work can be, in the future, extended towards several directions. First, a systematic comparison with the techniques for performing declarative process mining available in toolkits like RuM [2] and Declare4Py [8] could be conducted.

Even if we defined basic queries over the data attributes attached to events in a log (the RNG queries), more sophisticated queries could be defined, for example, for checking MP-Declare rules. These queries might represent the basis for novel approaches for discovery, conformance checking and query checking based on MP-Declare. Also, it would be easy to compute, using SQL queries, metrics for measuring the "interestingness" of a Declare rule [5] that go beyond the support we use in this paper (e.g., confidence).

The use of different query languages like PQL [25] could be investigated in the context of declarative process mining. Smarter strategies for storing event data in DBs like the ones investigated in [7] could help improving the query execution time. Other DB schema can be built for other standards for storing process information in event logs like the recent object-centric standards, such as XOC [19] and OCEL [15].

Another avenue for future work is the development of conformance checking SQL queries providing richer feedback to the user like trace alignments. Finally, the use of more advanced instruments from DB theory, such as the use of temporal DBs [17], could be investigated with the aim of improving the performance of the proposed framework.

References

1. Aamer, H., Montali, M., Van den Bussche, J.: What Can Database Query Processing Do for Instance-Spanning Constraints? CoRR abs/2206.00140 (2022)
2. Alman, A., Di Ciccio, C., Haas, D., Maggi, F.M., Nolte, A.: Rule Mining with RuM. In: 2nd International Conference on Process Mining, ICPM 2020, Padua, Italy, 4–9 October 2020, pp. 121–128 (2020)
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. Calì, A., Calvanese, D., Lenzerini, M.: Data integration under integrity constraints. In: Seminal Contributions to Information Systems Engineering, 25 Years of CAiSE, pp. 335–352. Springer (2013). https://doi.org/10.1007/978-3-642-36926-1
5. Cecconi, A., De Giacomo, G., Di Ciccio, C., Maggi, F.M., Mendling, J.: Measuring the interestingness of temporal logic behavioral specifications in process mining. Inf. Syst. **107**, 101920 (2022)
6. Chiariello, F., Maggi, F.M., Patrizi, F.: ASP-Based Declarative Process Mining. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2022)
7. Di Ciccio, C., Mecella, M.: On the discovery of declarative control flows for artful processes. ACM Trans. Manag. Inf. Syst. **5**(4), 24:1–24:37 (2015)
8. Donadello, I., Riva, F., Maggi, F.M., Shikhizada, A.: Declare4Py: a python library for declarative process mining. In: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2022, pp. 117–121 (2022)
9. van Dongen, B.F.: BPI Challenge 2012 (Apr 2012)
10. van Dongen, B.F.: BPI Challenge 2017 (Feb 2017)
11. van Dongen, B.F.: International Declarations Log. BPI Challenge 2020 (Mar 2020)
12. van Dongen, B.F.: Travel Permits Log. BPI Challenge 2020 (Mar 2020)
13. van Dongen, B.F., Shabani, S.: Relational XES: data management for process mining. In: CAiSE Forum 2015, pp. 169–176 (2015)
14. Ewen, S., Kache, H., Markl, V., Raman, V.: Progressive query optimization for federated queries. In: Advances in Database Technology - EDBT 2006. vol. 3896, pp. 847–864 (2006)
15. Ghahfarokhi, A., Park, G., Berti, A., van der Aalst, W.M.P.: OCEL: A Standard for Object-Centric Event Logs, pp. 169–175 (07 2021)
16. Hildebrandt, T.T., Mukkamala, R.R., Slaats, T., Zanitti, F.: Contracts for cross-organizational workflows as timed Dynamic Condition Response Graphs. J. Log. Algebr. Program. **82**(5–7), 164–185 (2013)
17. Kulkarni, K.G., Michels, J.: Temporal features in SQL: 2011. SIGMOD Rec. **41**(3), 34–43 (2012)
18. de Leoni, M., Mannhardt, F.: Road Traffic Fine Management Process (Feb 2015)
19. 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: Information Systems in the Big Data Era, pp. 182–199 (2018)
20. Maggi, F.M.: Discovering metric temporal business constraints from event logs. In: Johansson, B., Andersson, B., Holmberg, N. (eds.) BIR 2014. LNBIP, vol. 194, pp. 261–275. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11370-8_19
21. Maggi, F.M.: Declarative Process Mining. In: Sakr, S., Zomaya, A.Y. (eds) Encyclopedia of Big Data Technologies. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-77525-8_92
22. Mannhardt, F.: Sepsis Cases - Event Log (Dec 2016)
23. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: IEEE International EDOC Conference 2007, pp. 287–300 (2007)
24. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), pp. 287–300 (2007)
25. Polyvyanyy, A., ter Hofstede, A.H.M., La Rosa, M., Ouyang, C., Pika, A.: Process query language: Design, implementation, and evaluation. CoRR abs/1909.09543 (2019)

26. Räim, M., Di Ciccio, C., Maggi, F.M., Mecella, M., Mendling, J.: Log-based understanding of business processes through temporal logic query checking. In: Meersman, R. (ed.) OTM 2014. LNCS, vol. 8841, pp. 75–92. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45563-0_5
27. 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
28. Schönig, S.: SQL Queries for Declarative Process Mining on Event Logs of Relational Databases. CoRR abs/1512.00196 (2015)
29. Schönig, S., Di Ciccio, C., Maggi, F.M., Mendling, J.: Discovery of multi-perspective declarative process models. In: Sheng, Q.Z., Stroulia, E., Tata, S., Bhiri, S. (eds.) ICSOC 2016. LNCS, vol. 9936, pp. 87–103. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46295-0_6
30. Schönig, S., Di Ciccio, C., Mendling, J.: Configuring SQL-based process mining for performance and storage optimisation. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 94–97. SAC 2019 (2019)
31. Syamsiyah, A., van Dongen, B.F., van der Aalst, W.M.P.: DB-XES: enabling process discovery in the large. In: Ceravolo, P., Guetl, C., Rinderle-Ma, S. (eds.) SIMPDA 2016. LNBIP, vol. 307, pp. 53–77. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74161-1_4
32. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES, XESame, and ProM 6. In: Information Systems Evolution - CAiSE Forum. vol. 72, pp. 60–75 (2010)
33. Zeising, M., Schönig, S., Jablonski, S.: Towards a Common Platform for the Support of Routine and Agile Business Processes. In: Collaborative Computing: Networking, Applications and Worksharing (2014)

Optimizing the Solution Quality of Metaheuristics Through Process Mining Based on Selected Problems from Operations Research

Alexander Kinast[1(✉)], Roland Braune[2], Karl F. Doerner[2], and Stefanie Rinderle-Ma[3]

[1] Forschungsverbund Data Science, University of Vienna, Kolingasse 14-16, 1090 Wien, Austria
alexander.kinast@univie.ac.at
[2] Department of Business Decisions and Analytics, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Wien, Austria
[3] TUM School of Computation, Information, and Technology, Technical University of Munich, Boltzmannstrasse 3, 85748 Garching, Germany
stefanie.rinderle-ma@tum.de

Abstract. Methods from Operations Research (OR) are employed to address a diverse set of Business Process Management (BPM) problems such as determining optimum resource allocation for process tasks. However, it has not been comprehensively investigated how BPM methods can be used for solving OR problems, although process mining, for example, provides powerful analytical instruments. Hence, in this work, we show how process discovery, a subclass of process mining, can generate problem knowledge to optimize the solutions of metaheuristics to solve a novel OR problem, i.e., the combined cobot assignment and job shop scheduling problem. This problem is relevant as cobots can cooperate with humans without the need for a safe zone and currently significantly impact transitions in production environments. In detail, we propose two process discovery based neighborhood operators, namely process discovery change and process discovery dictionary change, and implement and evaluate them in comparison with random and greedy operations based on a real-world data set. The approach is also applied to another OR problem for generalizability reasons. The combined OR and process discovery approach shows promising results, especially for larger problem instances.

Keywords: Process Discovery · Operations Research · Metaheuristics · Memetic algorithm · Industry 4.0

1 Introduction

The application of techniques from Operations Research (OR) has been identified as promising *"avenue to obtain better processes"*, although *"OR techniques*

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 232–248, 2023.
https://doi.org/10.1007/978-3-031-41623-1_14

have not been systematically applied to solve process improvement problems yet" [2]. One example of the application of OR techniques to BPM is the allocation of resources to process tasks, e.g., [10]. Another example is the use of a memetic algorithm (MA) to mine change propagation behavior in process collaborations under confidential information, i.e., details on private processes [9]. Less attention has been paid to the reverse direction, i.e., how BPM methods such as process discovery (PD) can contribute to solve OR problems, even though PD provides powerful analytical instruments. [23] uses conformance checking to improve a scheduling problem in a hospital setting. In [13], we propose to use PD for visualization and exploration of solutions for a combined cobot assignment and job shop scheduling problem. In this case, the solutions that are generated by an MA are represented as process event logs. The discovered process models are then enriched by attributes such as cost and time for visual inspection and comparison of the solutions. In this work, we study how to exploit PD techniques for generating knowledge to optimize metaheuristics solutions based on two selected OR problems from the production domain, i.e., the cobot assignment and job shop scheduling problem [12] and flexible job shop scheduling problem [6] with an extended cobot assignment. As both problems are NP-hard optimization problems [29], metaheuristics offer promising solutions that are highly relevant in industry. MAs are one kind of metaheuristics that have proven useful for solving the cobot assignment and job shop scheduling problem [12]: a genetic algorithm explores the search space, and for promising solutions, a variable neighbourhood search (VNS) is performed. To investigate the potential of PD to metaheuristics, in this paper, we investigate i) how PD can be used in order to generate problem knowledge to optimize the solutions of the MA and ii) how much the solution quality can be increased. For this, we propose two PD-based neighborhood operators, namely process discovery change and process discovery dictionary change. Both operators are implemented and evaluated alongside two standard neighbourhood operations, i.e., basic change and greedy change, based on three data sets for the two problems described above. The results underpin the potential of PD-based neighbourhood operations, especially for large data sets and many cobots.

Figure 1 depicts an overview of the overall idea and algorithm. On the left side in the operations research part of Fig. 1, it can be seen that an optimization problem is loaded and initial solutions for the problem are generated. After loading the problem, the main loop of the MA (detailed description in Sects. 3 and 4) starts, and all individuals are evaluated. Whenever a new best solution is found, this solution is stored and a log file of this solution is created. In the BPM part of the Fig. 1, a local process model (LPM) is mined out of this log file and an LPM dictionary (described in Sects. 4.3 and 4.4) is created. Knowledge generated with these models can now be used to boost the performance of the MA. The paper has the following outline. Section 2 discusses related work. Section 3 explains fundamental concepts for the work, such as memetic algorithm and local process models. In Sect. 4, the solution for the selected OR problems, including the PD-based neighbourhood operators, is described. Section 5 presents the computational experiments to show the performance of the algorithm. Section 6 concludes the paper.

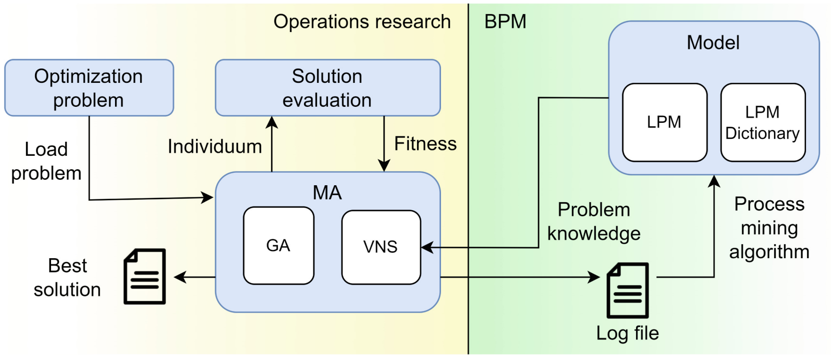


Fig. 1. MA with process discovery knowledge

2 Related Work

Scheduling problems in production are one of the hardest and most studied NP-hard optimization problems [29]. In [7], it is described that in the last decades, a lot of research has been done on developing efficient heuristic optimization algorithms for the (flexible) job shop scheduling problem due to its relevance for the industry. Especially local search methods like tabu search [25] have proven successful in this area. By combining the exploitation capabilities of local search with the explorative power of genetic algorithms [4], so-called MAs represent a hybrid between these two search paradigms. One of the most recent effective applications of such an algorithm to job shop scheduling is described in [30]. For process (re-)design, different OR methods have been used, e.g., mathematical programming [15]. [17] provides an overview of questions and approaches for automated planning in process design. OR methods are also used to determine the optimal data flow in process choreographies [14]. [5] put process model optimization to runtime based on formulating and solving a declarative process model plus temporal constraint as constraint satisfaction problem. Stochastic Petri nets [16,21] can be employed to model, simulate, and analyze dynamic process settings. PD has been mainly used to visualize and explore the results of the OR method to a given problem, so far. [8] uses process mining to analyze logs before scheduling in a hospital environment. In [13], we suggest using process mining to visually explore the results of cobot assignments by translating the schedules into logs. In [9], process mining is used to visualize and compare the solutions of an MA to predict change propagation behaviour in distributed process settings with and without confidentiality requirements. In [23] processes from a real-world clinic are improved: existing logs are analyzed, and a schedule is created with OR methods which is used to analyze the cause of deviations and to improve the process. However, the aforementioned approaches do not exploit process mining techniques to optimize OR methods.

3 Fundamentals

This section presents background information on memetic algorithms and local process model mining as the two fundamental concepts combined in the solution method presented in this work.

3.1 Memetic Algorithms (MA)

In order to understand how MAs work, we introduce the fundamentals of genetic algorithms (GA) and local search (LS) methods. GAs are an abstraction of biological evolution. A set of solutions (population) is the basis. Selection, crossover, and mutation operations transform this initial set of solutions to the next generation. A selection operator selects two parents for the next generation. The idea is that fitter individuals are selected more often. The crossover operator now combines these two individuals and therefore mimics biological reproduction. The mutation operator can slightly change the produced offspring, similar to a natural mutation. By representing a problem as an individual of the population and creating a fitness function, that can assign fitness to new individuals, this basic genetic algorithm can solve a broad range of optimization problems [18]

LS methods start with a single solution. A set of local changes are applied to the starting solution, which will improve the starting solution until a local optimum has been found. Basic local search methods will stop once a local optimum has been found. However, there are algorithms that can escape local optima and continue the search. One of these algorithms is a VNS. This algorithm explores increasing neighbourhoods (a k^{th} neighbouring solution can be reached with k changes to the base solution), e.g. neighbourhoods with 1, 2, or 3 changes to the base solution. If an improvement to the best solution has been found, the algorithm is restarted from the newfound solution [19].

A GA has a population and explores large parts of the search space. These GAs can be combined with LS so that the LS is applied to promising solutions that the GA finds. This MA combines global and local search methods and was able to provide good results for many practical problems [20].

3.2 Local Process Model Mining

Process models allow to *specify, describe, understand, and document processes more effectively than they can do using text*. Process models can be used to understand processes and make decisions [11]. Due to high concurrency and complex dependencies, simple sequence mining techniques do not work well on modern processes. However, process discovery (PD) algorithms have proven to capture processes adequately based on event logs [3]. Local process model (LPM) mining is a PD technique introduced in [24] that aims at extracting the best LPM from an event log. LPMs are generated based on an initial set of process trees containing only one node, i.e., one workstation. These trees are assigned a fitness value based on five quality criteria, such as the number of traces that can be considered an instance of the LPM (support) and the harmonic mean of all explainable event occurrences divided by not explainable event occurrences (confidence) [24]. All or a subset of the process trees are selected for the next generation based on their fitness. The process trees are then expanded with different operators and nodes. This is necessary, as one node might be no good presentation for a large process. In the expansion step, a leaf is replaced by an operator. The original leaf is the first child of the new operator, and a second random node is the second child of the operator. This expansion step is done multiple times, until a stopping criterium is reached, and the best process tree is stored. These generated process trees can be converted into LPMs at any time.

4 Solution Method

This section describes the OR problems and the MA that has been used to solve them (see [12]). Moreover, this section defines the novel PD-based neighbourhood operators.

4.1 Operations Research Problems

We present the necessary details of the two OR problems based on which we investigate the potential of employing PD in metaheuristics. [12] describes orders and tasks. However, for clarity, orders and tasks will be called jobs and operations. [6] describes machines. However, for clarity, machines will be called workstations, as human workers can interact with machines and cobots on these workstations.

OR Problem 1 (Cobot assignment and job shop scheduling problem [12]**)** *In this problem, a list of jobs is given. Each job contains multiple operations that are subject to precedence constraints. These operations should be executed on a given set of workstations.*

All workstations that can do similar operations are grouped, e.g. all drilling workstations. Each workstation has a speed and cost factor. An example would be a new drilling workstation. This new workstation might have more expensive drills, but it is also faster than an old one. Furthermore, a predefined number of cobots can be deployed to workstations in order to speed up production as introduced in [28]. For each operation, a base cost and duration are available. Additionally, a workstation group (e.g. a drilling operation can be done on any drilling workstation) as well as precedence relations are given.

The objective function of this problem is a combination of normalized production cost and normalized makespan. An extension to the classical job-shop-scheduling problem is that operations can be split and assigned to different workstations for quicker processing. Additionally, operation fragments can be executed in an arbitrary order.

OR Problem 2 (Flexible job shop scheduling [6]**)** *In this problem, a list of jobs is given. Each job contains multiple operations that are subject to precedence relations. Each operation can be processed on one out of a set of eligible workstations, and the processing time depends on the selected workstation.*

This base problem is extended by a cobot-to-workstation assignment.

The objective function of this problem is to minimize the makespan to finish all jobs. Therefore, operations must be assigned to the workstations, and a production order must be defined. The main difference to the first defined problem is the flexibility of operations (producible on many workstations instead of small workstation groups and the objective function).

Problems 1 + 2 are NP-hard problems extended by an additional decision aspect, i.e., a cobot-to-workstation-assignment. In [12], an MA has already been used to solve the cobot assignment and job shop scheduling problem. In this paper, we extend the MA with PD neighbourhood operators, i.e., if the genetic algorithm finds a promising solution, the VNS is started from this solution.

4.2 Encoding and Evaluation

In Fig. 2, the encoding for Problems 1 + 2 is shown. The first part of the encoding is the operation to workstation assignment. If an operation can be produced on multiple workstations, the upper bound equals the number of these workstations. The value represents which of the possible workstations is used for production. E.g. if workstations 0 to 4 are possible for production, the upper bound is 4, and a value of two would mean that the second workstation is used.

In the second part of the encoding in Fig. 2, each operation's priority is encoded. If multiple operations can be produced simultaneously at the same workstation, the operation with the highest priority gets produced first. Each number between zero and the largest possible integer is possible. E.g. two tasks, task 1 with priority 5 and task 2 with priority 10, should be produced on the same workstation. The task with the highest priority, namely task 2, is produced first.

The final part of the encoding is the cobot-to-workstations-assignment that can be seen on the right side of Fig. 2. The upper bound of this value is the number of workstations that have no cobot assigned. The value represents which of these workstations a cobot should be assigned. E.g. if workstations 0 to 10 have no cobot assigned, the upper bound of the value is 10 and a value of 5 would mean that a cobot is assigned to workstation 5.

In [28], it is described that a cobot speeds up production by 30%. This value is used for the evaluation.

Details regarding the evaluation of the extended cobot assignment and job shop scheduling problem can be found in [12]. During the evaluation of one solution, two objective values (production cost and makespan) are generated. The objective function F that is used as fitness is a combination of normalized cost and normalized makespan, i.e., $F = n_{cost} + n_{makespan}$.

Instances of the extended cobot assignment and flexible job shop scheduling problem do not define production costs. Therefore, the makespan should be minimized in these instances.

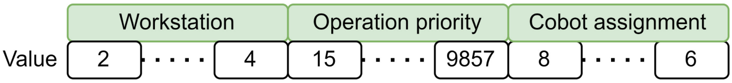


Fig. 2. Integer-based encoding

4.3 Local Process Model Mining

In this work, an LPM represents highly used workstations and relations between these workstations in the currently best solution of the algorithm. More precisely, the best-rated LPM is mined whenever the MA finds a new best solution. Section 4.4 will explain how one or multiple LPMs are used inside the MA to improve the algorithm's performance. However, the conceptual idea is that solutions that are close to the best-found solution might improve if more operations are assigned to those highly used workstations of the best solution.

Compared to [24], the computational effort is crucial in the context of this paper. Hence, the LPM mining algorithm is adjusted to be executable in the MA. For this, the number of operators to build the LPMs is limited to sequence and xor operators. Regarding the described problems, this deviation from the originally proposed mining algorithm does not have disadvantages, as the problems are defined without loops and concurrency. Additionally, generating all possible solutions in the selection step is not feasible. Therefore, a random subset is generated before the expansion step. Figure 3 shows the process of generating an LPM during the run of the MA. Green parts have been developed or adjusted for this paper. The basis for the LPM is the event log representing the current best solution found by the MA.

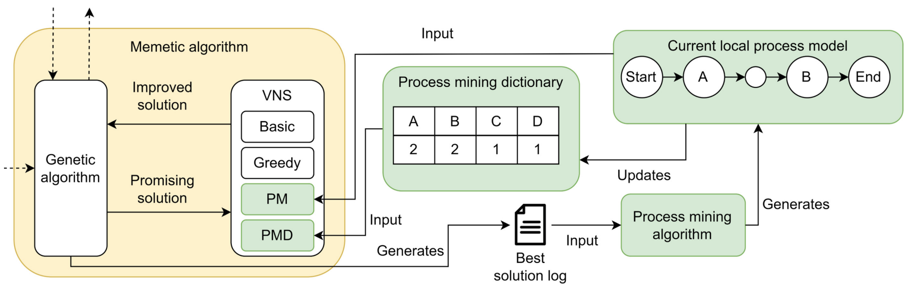


Fig. 3. Generation and usage of local process models

During the evaluation of one solution, information regarding jobs, order of operations, workstations, and cobot placement is available. The generation of an event log in the xes format [1] is triggered every time a new best solution is found. The xes file contains a trace for each job. The traces, in turn, contain events for each operation. Each event defines a start and end timestamp (start and end of the operation), a job ID, an operation ID, which workstation has been used and the information if a cobot is currently assigned to this workstation. Using the adjusted LPM mining algorithm, the best LPM for this given log file is created. An example would be the sequential execution on workstations A and B in a problem with four workstations A-D. Section 4.4 will explain how LPMs are used in the VNS.

4.4 Memetic Algorithm

To generate neighbourhood solutions in a MA as described in Sect. 3, a neighbourhood operator applies k changes to an initial solution.

Independent of the neighbourhood operator, each part of the encoding, described in Fig. 2, has an equal chance of being selected for change (operation-to-workstation assignment, operation priority, cobot assignment). The first neighbourhood operator is the **Basic change (B)**. In this change, one value of the selected part is randomized within its bounds. The second operator is the **Greedy change (G)**. Regarding operator priority and cobot assignment, this change equals the basic change. In the operation-to-workstation assignment, all workstations that have a cobot assigned are calculated. Workstations with cobots have a threefold probability of being selected during the operation-to-workstation assignment. The third operator is the **Process discovery change (PD)**. Regarding operator priority and cobot assignment, this change equals the basic change. In the operation-to-workstation assignment, the latest LPM is used. This can be seen in Fig. 3. Workstations that are part of the latest LPM have a threefold probability of being selected compared to other workstations. The final operator is the **Process discovery dictionary change (PDD)**. Regarding operator priority, this change is equal to the basic change. In the operation-to-workstation assignment and the cobot-to-workstation assignment, the weight of each workstation is the weight of the entry in the process mining dictionary. This can be seen in Fig. 3. This means that workstations that greatly impact the process over multiple generations of LPMs have a higher chance of being selected.

In Algorithm 1, the evaluation of the MA (cf. [12]) with PD-based VNS is described. In line 0, a solution and one neighbourhood operator are passed to the evaluation method. A fitness value for this solution, called solutionFitness, is generated. This can be seen in Algorithm 1 in line 1. In line 2, it is checked if the VNS should be applied. It is applied to solutions within a given range of the best solution that has been found so far. Lines 3, 4, 5, 14, and 15 indicate the minimum number of individuals generated whenever the VNS is started. An example would be k_{max}=5, where at least 50 solutions with $k \in \{1, 3, 5\}$ changes are generated. In line 6, k changes are made to the existing best solution based on the passed neighbourhood operator of line 0. Four neighbourhood operators, i.e., basic change, greedy change, PD change, and PDD change, are used in line 6 and will be explained in detail after the algorithm description. In line 7, the

Algorithm 1. Pseudo code - MA with process discovery

Parameters/Methods	Description
BestSolution	Best solution found so far
BestSolutionLog	Log file describing the best solution found so far
BestFitness	Best fitness found so far
VnsThreshhold	Threshold to check if the VNS should be applied
LPM	Local process model of the best solution found so far
LPMDictionary	Dictionary that is updated based on mined LPMs
UpdateDictionary()	Method that updated the current LPMDictionary
EvaluateSolution()	Method to get the quality of a passed individual
SolutionLog()	Method to get the log file of a solution
MineLocalProcessModel()	Method to mine a LPM out of a process log
NeighbouringSolution()	Method that generates a solution with k changes

```
0  Evaluate(solution, neighbourhood, k_max)
1      solutionFitness ← EvaluateSolution(solution)
2      if(solutionFitness ≤ BestFitness * VnsThreshhold)
3          k ← 1
4          while(k ≤ k_max)
5              for(i = 0, i ≤ 50, i++)
6                  newSolution ← NeighbouringSolution(solution, neighbourhood, k)
7                  newSolutionFitness ← EvaluateSolution(s')
8                  if(newSolutionFitness < solutionFitness)
9                      solutionFitness ← newSolutionFitness
10                     solution ← newSolution
11                     k ← 1
12                     goto line 3
13                 end if
14             end for
15             k += 2
16         end while
17     end if
18     if(solutionFitness < BestFitness)
19         BestFitness = solutionFitness
20         BestSolution = solution
21         BestSolutionLog = SolutionLog(solution)
22         LPM = MineLocalProcessModel(BestSolutionLog)
23         LPMDictionary = UpdateDictionary(LocalProcessModel)
24     end if
25     return x
```

fitness of the new changed solution is evaluated.

The VNS is restarted on a first-improvement basis. This can be seen in lines 8 to 13. The found improved solution replaces the current best solution for this variable neighbourhood.

Lines 18 to 24 show that the MA also stores the best-found solution. If a new best solution is found and the neighbourhood operator is one of the PD operators, a solution log of this solution is generated and a LPM mining algorithm is applied. The mined LPM replaces the LPM of the currently best solution. It is assumed that key workstations of existing solutions are part of the LPM. Examples are workstations that are used frequently in the best existing solution. Utilizing this information in the genetic algorithm might be good for already promising solutions to assign operations to these workstations.

Figure 3 shows the development of a PDD. A dictionary with all workstations is created to utilize information extracted from multiple LPMs. Each workstation has a base weight of 1. Each time a new LPM is mined, the weight of all workstations that are part of this LPM is increased by one.

5 Numerical Experiments

5.1 Data and Code

The problem files for the cobot assignment and job shop scheduling problem can be found at https://doi.org/10.5281/zenodo.7691316 and the problem files for the cobot assignment and flexible job shop scheduling problem at https://doi.org/10.5281/zenodo.7691455. Algorithm 1 is implemented in C# and embedded into HeuristicLab, a framework for heuristic optimization [26]. The simulation framework Easy4Sim[1] was used to evaluate solutions. The code is provided at https://zenodo.org/badge/latestdoi/614876607.

The evaluation of the approach necessitates large-scale computational experiments. For this, the HPC3 cluster[2] in Vienna was used. All calculations were executed on nodes with a Xeon-G 6226R CPU 2.9 GHz. To execute the C# code on a Linux cluster, the mono framework [27] was utilized. To evaluate the overhead of the runtime environment, preliminary experiments were conducted. Stretching the computation time by a factor of 1.6 allows for a similar number of solutions to be evaluated compared to the same code running on a native.NET platform (MS Windows). All runs of the MA have been done on the HPC. Therefore, this factor has been used for all runs of the MA, and the original runtime is reported in this paper.

5.2 Constraint Programming Formulation

A constraint programming (CP) formulation for all solved problems has been done to measure the implemented algorithm's performance. If the CP model terminates, it finds the global optimum of a problem. Therefore the CP model gives an overview of the complexity of the problem (can the optimum be found in a reasonable time?). If no optimum is found, it gives a good base quality which can be compared to the solutions found by the developed algorithm.

The exact formulation for the first two data sets can be found in [12]. To solve the third problem, minor adjustments have been made to the CP model. Since

[1] https://www.risc-software.at/.
[2] https://w3.vdc.univie.ac.at/wiki/index.php/Slurm.

the CP formulation of the problem requires a lot of space, it is not presented here. IBM ILOG CP Optimizer has been employed for implementing the model and for solving the example problems. The model definition can be found at https://doi.org/10.5281/zenodo.7754794.

5.3 Data Dimensions

Three different data sets were solved with all neighbourhood operators. In [12], the first two data sets are explained. The first data set is a combined cobot assignment and job shop scheduling problem from the industry. It has 54 workstations, 210 jobs, and 1265 operations. This instance is split into two halves and four quarters to create additional smaller instances. The second data set is inspired by this real-world data set and has 50 artificial instances. These instances are similar in size compared to real-world instances (full, halves, quarters). In [6], the third data set is introduced. This data set contains large flexible job shop scheduling instances that are extended with a cobot-to-workstation-assignment in this paper. The instance size ranges from 30×10 (jobs $\times$ workstations) to 100×20.

5.4 Real-World Cobot Assignment and Job Shop Scheduling Problem

The real-world problem described in [12] has been solved with the following parameters:

- Runtime: Short (100 min), medium (200 min), and long (300 min) runtime
- Cobots: 0, 5, and 10
- neighbourhood operator: Basic change, greedy change, PD change, PDD change
- Instances: Full, halves, quarters
- Repetitions: 10

These settings result in 2520 runs of the MA. The reported runtime is used for the full instance and 30%, and 10% of this runtime is used for the half and quarter instances, respectively. In [12], the CP model has been used to solve the real-world data set with zero and five cobots.

In Table 1, the average normalized objective value of all runs of the MA is reported and compared to the CP results. Both values of the objective function (makespan, cost) are normalized so that a higher normalized value represents a better value. The maximum of each normalized value is 1, which means the closer the objective value gets to 2, the better the result is. The values in the cells represent the average for 10, 20, and 40 runs of the algorithm for the full instance, the half instances, and the four quarters, respectively.

The colored cells mark the best neighbourhood for each combination of runtime, instances size, and the number of cobots. This highlights the advantages of the different neighbourhood operators.

Table 1. Detailed results for the real-world problem

		0 Cobots			5 Cobots			10 Cobots		
NBOp	Time	Quarters	Halves	Full	Quarters	Halves	Full	Quarters	Halves	Full
B	Short	1.2698	0.9654	1.0644	1.7878	1.5955	1.6491	1.7926	1.5995	1.6414
G		1.2213	0.9574	1.0423	1.7565	1.5714	1.6583	1.7884	1.6245	1.7343
PD		1.2475	0.8933	1.0262	1.7604	1.5790	1.6714	1.8168	1.6277	1.7423
PDD		1.2766	0.9653	1.0722	1.6111	1.5351	1.5504	1.8176	1.6200	1.7279
B	Medium	1.2876	0.9915	1.0854	1.8227	1.6115	1.7227	1.8489	1.6563	1.7329
G		1.2886	0.9845	1.0646	1.8074	1.6018	1.7164	1.8343	1.6516	1.7457
PD		1.2838	1.0436	1.0821	1.8134	1.5967	1.7071	1.8431	1.6746	1.7511
PDD		1.2877	1.0349	1.1154	1.6463	1.6624	1.7268	1.8540	1.6776	1.7919
B	Long	1.3004	1.0897	1.1099	1.8322	1.6244	1.7506	1.8584	1.7019	1.7617
G		1.2968	1.0587	1.0685	1.8215	1.6738	1.7394	1.8455	1.6829	1.7461
PD		1.2946	1.0322	1.0973	1.8257	1.6738	1.7237	1.8616	1.6936	1.7975
PDD		1.2954	1.0520	1.0930	1.6660	1.6756	1.7814	1.8672	1.6921	1.8086
CP		0.9057	1.1493	1.0216	-2.4404	-1.0479	0.9258			

NBOp: neighbourhood Operator; B: Basic; G: Greedy; PD(D): Process discovery (Dictionary)

The PD operators try to identify important workstations in generated solutions. The PDD operator even learns over a large number of generations. Since applying PD operators comes with an overhead, the instance must be hard enough that this generated knowledge has enough impact in the remaining time. In Table 1, it can be seen that the PD operators, especially the PDD operator, outperform other neighbourhood operators on complex problems (large instance, high number of cobots) if enough runtime is given.

Once the number of cobots increases, the CP model has difficulties finding a valid solution. This can be seen in the last line of Table 1. The CP approach delivers good zero cobot results, especially for the half instances. However, with five cobots, the CP formulation already has trouble finding valid solutions.

5.5 Generated Cobot Assignment and Job Shop Scheduling Problem

The second data set solved is the artificial data set described in [12]. In this data set, 50 instances in 3 sizes have been created:

- Small instances (10 instances)
  300 operations / 30 workstations
- Medium instances (20 instances)
  600 operations / 30 workstations
  600 operations / 50 workstations
- Large instances (20 instances)
  1200 operations / 30 workstations
  1200 operations / 50 workstations

All instances have been solved with the following parameters:

- Cobots: 0, 5, and 10

Table 2. Detailed results for the artificially generated instances

	0 cobots			5 cobots			10 cobots		
neighbourhood	Small	Medium	Large	Small	Medium	Large	Small	Medium	Large
B	0.8140	0.5258	0.3657	0.9704	0.6342	0.5450	1.1592	0.7708	0.6781
G	0.8124	0.4991	0.3483	0.9463	0.6188	0.5399	1.1331	0.7570	0.6792
PD	0.8192	0.5243	0.3883	0.9767	0.6473	0.5576	1.1462	0.7799	0.6801
PDD	0.8123	0.5355	0.3897	1.0100	0.6632	0.5426	1.1903	0.7953	0.6815
CP	0.4139	0.2989	0.1240	0.4898	0.2989	-0.2926	0.5690	0.2989	-0.2926

NBOp: neighbourhood Operator; B: Basic; G: Greedy; PD(D): Process discovery (Dictionary)

– neighbourhood operator: Basic change, greedy change, PD change, PDD change
– Repetitions: 10

These settings result in 6000 runs of the algorithm. The runtime was 60, 180, and 300 min for the small, medium, and large instances, respectively.

Table 2 summarizes the performance of the neighbourhood operators compared to the CP model on the artificial instances. The value in each cell represents the normalized objective value (normalized cost + normalized makespan) with an upper bound of 2. A larger value means that on average better solutions have been found. The coloured cells represent the best neighbourhood operator with regard to the instance size and the number of cobots.

If the CP model did not find a solution for a cobot setting, the solution with fewer cobots is taken. It can be seen that the MA outperforms the CP model for this problem. This is independent of the used neighbourhood.

Table 2 shows that the PD neighbourhood operators outperform the basic and greedy neighbourhood operators over the whole data set. This is again especially true for the PDD operator. Which performs, on average, 2.4% better than the basic neighbourhood, 4.5% better than the greedy neighbourhood, and 1.5% better than the PD neighbourhood.

The values in the table represent the average over 100 and 200 instances for the small and medium/large instances, respectively. Due to the larger number of instances, results from this data set are less prone to errors than the real-world instances.

5.6 Cobot Assignment and Flexible Job Shop Scheduling Problem

The third data set solved is the flexible job shop scheduling problem described in [6], cf. Problem 2. For the previous two problems, it could be seen that the CP results have performed better for simpler instances and worse for complex instances. In this problem, CP delivers good results due to the simple, makespan-only objective. To compete with the CP formulation with an equal runtime, minor adaptations had to be done in the MA.

A fraction of the initial population of the MA has been initialized with solutions generated using priority dispatching rules. These priority rules allow the generation of acceptable initial solutions that can be further improved with the

MA. A considerable amount of priority rules are described in [22]. The following have been used in the operation-to-workstation assignment in this paper:

– Most work remaining
– Shortest processing time remaining
– Most operations remaining
– Operational flow slack per processing time
– Flow due date per work remaining
– Shortest processing time per work remaining
– Shortest processing time and work in next queue

Additionally, the *Highest number of assignable operations* and the *Largest amount of assignable work* priority rules have been designed for the cobot-to-workstation assignment. In the first cobot-to-workstation rule, workstations are sorted by the number of operations that can be assigned and the available cobots are assigned to the top workstations. In the second rule, the sorting is done by the duration of all assignable operations on a specific workstation.

Additionally, generating LPMs has been stopped until the first generation is finished. Four problem files (0, 10, 20, 30) of two categories (smallest and largest) were selected. The smallest category has 30 jobs with 10 workstations, and the largest has 100 jobs with 20 workstations. These problems were solved with cobots assigned to 0%, 20%, and 40% of the workstations. Each problem was solved with all four described neighbourhood operators and 20 repetitions. This resulted in 1920 runs of the algorithm. The CP solver and the MA had a runtime limit of 60 min.

Table 3. Detailed makespan results for the cobot assignment and FJSP

	B	G	PD	PDD	CP
small 0%	764	764	765	764	762
small 20%	702	703	703	702	699
small 40%	650	650	650	649	646
large 0%	3906	3906	3906	3906	3904
large 20%	3587	3587	3587	3587	3587
large 40%	3314	3314	3314	3314	3317

Table 3 reports the average solution quality for the MA and the CP model of the flexible job shop scheduling instances. The values represent the average objective value (makespan) across each instance group. Hence, smaller values indicate a better solution quality. The CP solver delivered good results for all numbers of cobots for the small instances. With growing problem difficulty (increasing number of cobots and instance size), the performance of the MA increased.

For the large instances with 40% cobots, a slight advantage of the MA over the CP model can be observed. Even though the performance of the neighbourhood operators is pretty similar, the PDD operator outperforms the other neighbourhood operators again and delivers the best results for the most complex (largest, highest amount of cobots) instances.

6 Summary and Outlook

This paper introduces a novel combination of an MA with a feedback loop that utilizes LPM mining. This MA uses different neighbourhoods that utilize the information generated with this PD algorithm, and the results are compared to traditional neighbourhood operators and a CP model.

Two problems from OR were tackled to show the algorithm's generalizability. The algorithm should be easily adaptable to new problems due to the flexibility of the base algorithm, the genetic algorithm. Additionally, it has been implemented in HeuristicLab, which can, due to its plugin-based architecture, easily be extended with new problem formulations.

Running additional code like the PD algorithms during the execution of a genetic algorithm to generate knowledge comes with overhead. This knowledge can help identify important parts of the process.

A series of experiments on different problems were started to quantify the impact of this generated knowledge. This paper reports the results of 10440 runs of the MA. A CP formulation was employed for all problems to have a baseline performance measure.

For small instances and simple problems, the overhead incurred through PD inhibits the competitiveness of our approach. However, it was shown that neighbourhood operators that utilize PD algorithms to generate knowledge outperform other neighbourhood operators and the CP model on large and complex instances. In further research, different metaheuristics, feedback variants, problems, and PD algorithms can be reviewed. In the current version, the order and connections between workstations in the LPM is not utilized, however, utilizing this information might be helpful in upcoming research.

Acknowledgments. This work has been partly funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 277991500. Additionally, we are thankful that the RISC Software GmbH allowed us to use the simulation framework, Easy4Sim.

References

1. IEEE standard for eXtensible event stream (XES) for achieving interoperability in event logs and event streams. IEEE Std **1849**, 1–50 (2016). https://doi.org/10.1109/IEEESTD.2016.7740858
2. van der Aalst, W.M.P., Rosa, M.L., Santoro, F.M.: Business process management - don't forget to improve the process! Bus. Inf. Syst. Eng. **58**(1), 1–6 (2016). https://doi.org/10.1007/s12599-015-0409-x
3. van der Aalst, W.: Process discovery: capturing the invisible. IEEE Comput. Intell. Mag. **5**(1), 28–41 (2010). https://doi.org/10.1109/MCI.2009.935307

4. Affenzeller, M., Wagner, S., Winkler, S., Beham, A.: Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications. Chapman and Hall/CRC, 1 edn. (2009). https://doi.org/10.1201/9781420011326
5. Barba, I., Jiménez-Ramírez, A., Reichert, M., Valle, C.D., Weber, B.: Flexible runtime support of business processes under rolling planning horizons. Expert Syst. Appl. **177**, 114857 (2021). https://doi.org/10.1016/j.eswa.2021.114857
6. Braune, R., Benda, F., Doerner, K.F., Hartl, R.F.: A genetic programming learning approach to generate dispatching rules for flexible shop scheduling problems. Int. J. Prod. Econ. **243**, 108342 (2022). https://doi.org/10.1016/j.ijpe.2021.108342
7. Chaudhry, I.A., Khan, A.A.: A research survey: review of flexible job shop scheduling techniques. Int. Trans. Oper. Res. **23**(3), 551–591 (2016). https://doi.org/10.1111/itor.12199
8. Cunzolo, M.D., et al.: Combining process mining and optimization: A scheduling application in healthcare. In: Business Process Management Workshops, pp. 197–209 (2022). https://doi.org/10.1007/978-3-031-25383-6_15
9. Fdhila, W., Rinderle-Ma, S., Indiono, C.: Memetic algorithms for mining change logs in process choreographies. In: Franch, X., Ghose, A.K., Lewis, G.A., Bhiri, S. (eds.) ICSOC 2014. LNCS, vol. 8831, pp. 47–62. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45391-9_4
10. Ihde, S., Pufahl, L., Völker, M., Goel, A., Weske, M.: A framework for modeling and executing task-specific resource allocations in business processes. Computing **104**(11), 2405–2429 (2022). https://doi.org/10.1007/s00607-022-01093-2
11. Kesari, M., Chang, S., Seddon, P.B.: A content-analytic study of the advantages and disadvantages of process modelling. In: ACIS 2003 Proceedings, p. 2 (2003)
12. Kinast, A., Braune, R., Doerner, K.F., Rinderle-Ma, S., Weckenborg, C.: A hybrid metaheuristic solution approach for the cobot assignment and job shop scheduling problem. J. Ind. Inf. Integr. **28**, 100350 (2022). https://doi.org/10.1016/j.jii.2022.100350
13. Kinast, A., Doerner, K.F., Rinderle-Ma, S.: Combining metaheuristics and process mining: improving cobot placement in a combined cobot assignment and job shop scheduling problem. Procedia Comput. Sci. **200**, 1836–1845 (2022). https://doi.org/10.1016/j.procs.2022.01.384
14. Köpke, J., Franceschetti, M., Eder, J.: Optimizing data-flow implementations for inter-organizational processes. Distrib. Parallel Databases **37**(4), 651–695 (2019). https://doi.org/10.1007/s10619-018-7251-3
15. Kumar, A., Liu, R.: Business workflow optimization through process model redesign. IEEE Trans. Eng. Manage. **69**(6), 3068–3084 (2022). https://doi.org/10.1109/TEM.2020.3028040
16. Leemans, S.J.J., Maggi, F.M., Montali, M.: Reasoning on labelled petri nets and their dynamics in a stochastic setting. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds) Business Process Management. BPM 2022. Lecture Notes in Computer Science, vol 13420. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16103-2_22
17. Marrella, A., Chakraborti, T.: Applications of automated planning for business process management. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 30–36. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_4
18. Mitchell, M.: Genetic algorithms: An overview. In: Complex, vol. 1, pp. 31–39. Citeseer (1995)

19. Mladenović, N., Hansen, P.: Variable neighborhood search. Computers & Operations Research **24**(11), 1097–1100 (1997). https://doi.org/10.1016/S0305-0548(97)00031-2. https://www.sciencedirect.com/science/article/abs/pii/S0305054897000312
20. Moscato, P., Cotta, C.: An accelerated introduction to memetic algorithms. In: Gendreau, M., Potvin, J.-Y. (eds.) Handbook of Metaheuristics. ISORMS, vol. 272, pp. 275–309. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-91086-4_9
21. Rogge-Solti, A., Weske, M.: Prediction of business process durations using non-markovian stochastic petri nets. Inf. Syst. **54**, 1–14 (2015). https://doi.org/10.1016/j.is.2015.04.004
22. Sels, V., Gheysen, N., Vanhoucke, M.: A comparison of priority rules for the job shop scheduling problem under different flow time- and tardiness-related objective functions. Int. J. Prod. Res. **50**, 4255–4270 (2012). https://doi.org/10.1080/00207543.2011.611539
23. Senderovich, A., et al.: Conformance checking and performance improvement in scheduled processes: a queueing-network perspective. Inf. Syst. **62**, 185–206 (2016). https://doi.org/10.1016/j.is.2016.01.002
24. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.: Mining local process models. J. Innovation Digit. Ecosyst. **3**(2), 183–196 (2016). https://doi.org/10.1016/j.jides.2016.11.001
25. Vilcot, G., Billaut, J.C.: A tabu search algorithm for solving a multicriteria flexible job shop scheduling problem. Int. J. Prod. Res. **49**(23), 6963–6980 (2011). https://doi.org/10.1080/00207543.2010.526016
26. Wagner, S., et al.: Architecture and design of the heuristiclab optimization environment. In: Klempous, R., Nikodem, J., Jacak, W., Chaczko, Z. (eds) Advanced Methods and Applications in Computational Intelligence. Topics in Intelligent Engineering and Informatics, vol 6. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-319-01436-4_10
27. Mono framework. https://www.mono-project.com/. Accessed 15 Feb 2023
28. Weckenborg, C., Kieckhäfer, K., Müller, C., Grunewald, M., Spengler, T.S.: Balancing of assembly lines with collaborative robots. Bus. Res. **13**(1), 93–132 (2020). https://doi.org/10.1007/s40685-019-0101-y
29. Xie, J., Gao, L., Peng, K., Li, X., Li, H.: Review on flexible job shop scheduling. IET Collaborative Intell. Manuf. **1**(3), 67–77 (2019). https://doi.org/10.1049/iet-cim.2018.0009
30. Zhang, G., Sun, J., Lu, X., Zhang, H.: An improved memetic algorithm for the flexible job shop scheduling problem with transportation times. Meas. Contr. **53**(7–8), 1518–1528 (2020). https://doi.org/10.1177/0020294020948094

Resource Allocation in Recommender Systems for Global KPI Improvement

Alessandro Padella(✉) and Massimiliano de Leoni

University of Padua, Padua, Italy
alessandro.padella@phd.unipd.it, deleoni@math.unipd.it

Abstract. Process-aware Recommender systems are information systems designed to monitor the execution of processes, predict their outcomes, and suggest effective interventions to achieve better results, with respect to reference KPIs (Key Performance Indicators). Interventions typically consist of suggesting an activity to be assigned to a certain resource. State of the art typically proposes interventions for single cases in isolation. However, since resources are shared among cases, this might impact the effectiveness of the available interventions for other cases that would require one. As result, the overall KPI improvement is partially hampered. This paper proposes an approach to assign resources to needed cases, aiming to improve the overall KPI values for all cases together, namely the summation of KPI values for all cases. Experiments conducted on two real-life case studies illustrate that globally considering all needing cases together allows a better global KPI improvement, compared with a more greedy approach where interventions are proposed one after the other.

Keywords: Process Improvement · Process Prescriptive Analytics · Recommender Systems · Resource Allocation · Resource Experience

1 Introduction

Process-aware Recommender Systems are a class of information system that aims to monitor whether executions are predicting to achieve the expected goals, and, whenever this is not the case, they propose interventions to try to take those executions back on track. In literature interventions are typically based on advising what activity to perform as next, possibly paired with a suggestion of the resource that will carry it out (see Sect. 2).

However, resources are shared among all running process instances, a.k.a. cases, and typically they can carry on one activity a time. As a result, if the interventions (activity and resource) are determined for each instance without considering the other instances that also require intervention, the overall effectiveness, namely for all instances that require interventions, is limited. For instance, if one process instance P1 is assigned a resource R1, R1 cannot work a different instance P2 that requires intervention. It might be the case that it

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 249–266, 2023.
https://doi.org/10.1007/978-3-031-41623-1_15

is more beneficial to assign a different resource R2 to P1, because R2 can work almost as good for P1, and let R1 work on P2, for which no resource exists that is almost equally good. This consideration illustrates how the decision of the interventions is a global decision for all running cases.

Section 2 reports how literature does not propose approaches for resource allocations to cases with the aim of improving the whole set of process instances. This paper proposes an approach to overtake this limitation. In our paper, the achievement of the goal is measured through a measurable Key Performance Indicator (KPI): cases associated with values outside the acceptable range are considered worth of an intervention. The problem that we tackle is clearly an optimization problem: given the likely process' scenario of hundreds of resources and running cases, an exact solution is practically unfeasible. We thus propose two greedy approaches, one faster and one slower, that respectively provide a worse and better approximation.

In a nutshell, the idea is to create an initial resource profile, in which resources are allocated to cases: each profile is associated with an expected overall KPI improvement. The expected overall KPI improvement is computed, using machine-learning techniques for prescriptive process analytics. Then, this initial profile is altered, by changing resources allocated to cases, thus obtaining further profiles. At the end, a number of resource profiles is generated, from which those with higher expected KPI improvements are retained. The ultimate outcome is a set of resource profiles, which are deemed valid to improve KPIs. In different resource profiles, the same resource is assigned to a different case and intervention: resources are thus given a certain degree of freedom on which case to pick up and work on as next. This is beneficial, because a rigid resource-to-case imposition is against the principle of resource-aware recommender systems [1], such as how the problem of task-to-resource assignments is typically tackled in operation research.

The framework has been assessed on real-life case studies with processes with hundreds of running cases and resources. This allowed us to perform a stress test on the practical feasibility. The typical behavior of resources was simulated, on the basis of behavior patterns observed in human-computer interaction literature. The results show that our resource-allocation framework enables a significantly higher total KPI improvements, i.e. considering all running cases, if compared with scenarions in which each case is recommended in isolation.

Section 2 discusses related works in the domain of prescriptive process analytics and resource-aware recommender systems. Section 3 introduces the necessary background concepts: event logs, KPI definitions, and process prescriptive analytics that consider single cases in isolation. Section 4 puts forward our approach for resource allocation for global KPI improvement, Sect. 5 reports on the evaluation setup and results, while Sect. 6 concludes the paper, summarizing the contribution in our paper.

2 Related Works

Literature has focused on using recommender systems in business processes to improve the future outcome of process instances. This has often been translated into being focused on recommending which activities to work on next to improve the process' Key Performance Indicators (KPIs) [2–4].

The growing interest in recommender systems for process mining has led the community to explore how to determine when to intervene with recommendations [5] and whether an intervention is cost-wise worth [6]. A body of research has also focused on ensuring that recommendation are well explained to human-resources [7], using Shapley Values theory [8].

Moreover, several research works have focused on considering which resources should perform specific activities in various contexts. Cabanillas et al. in [9, 10] propose two approaches to define a language to equip BPMN models with complex resource-allocation policies, as well as to discover those policies.

A few works focuses on suggesting a resource allocation for a set of activities that need to be performed, without - though - focusing on recommending which activity to perform. Zhao et al. in [11] provide a framework based on a system of convex equations that encode a system of constraints on time and cost. Huang et al. in [12] leverage on Reinforcement Learning to propose a resource-allocation algorithm based on a Markov decision process. Park et al. in [13] integrate offline prediction model construction, using Long Short-Term Memory models to predict the next activity to perform and, subsequently, employing a minimum-cost-and-maximum-flow algorithm to allocate resources. Dumas et al. in [14] focus on recommending resource-activity pair, as we aim here. However, they recommend for single cases in isolation, which was previously mentioned to provide a lower degree of overall KPI improvement, namely for the whole set of running cases.

In conclusion, no previous research works have pursued the goal to provide an global KPI improvements, while leaving a certain degree of freedom to resources on which case (and activity) to work on as next.

3 Preliminaries

The starting point for a process mining-based system is an *event log*. An event log is a multiset of *traces*. Each trace is a sequence of events, each describing the life-cycle of a particular *process instance* (i.e. a *case*) in terms of the *activities* executed, *resources* that execute it, and the process *attributes* manipulated.

Definition 1 (Events). *Let $\mathcal{A}$ be the set of process activities. Let $\mathcal{R}$ be the set of possible resources. Let $\mathcal{V}$ be the set of process attributes. Let $\mathcal{W}_\mathcal{V}$ be a function that assigns a domain $\mathcal{W}_\mathcal{V}(x)$ to each process attribute $x \in \mathcal{V}$. Let $\overline{\mathcal{W}} = \cup_{x \in \mathcal{V}} \mathcal{W}_\mathcal{V}(x)$. An event is a tuple $(a, r, v) \in \mathcal{A} \times \mathcal{R} \times (\mathcal{V} \not\rightarrow \overline{\mathcal{W}})$ where a is the event activity, r the resource that performs it and v is a partial function assigning values to process attributes with $v(x) \in \mathcal{W}_\mathcal{V}(x)$.*

A trace is a sequence of events. The same event can occur in different traces, namely attributes are given the same assignment in different traces. This means that the entire same trace can appear multiple times and motivates why an event log is to be defined as a function which assigns a trace to a given identifier:

Definition 2 (Traces & Event Logs). *Let $\mathcal{E} = \mathcal{A} \times \mathcal{R} \times (\mathcal{V} \nrightarrow \overline{\mathcal{W}})$ be the universe of events. Let $\mathcal{I}$ be the universe of the case identifiers. A trace σ a sequence of events, i.e. $\sigma \in \mathcal{E}^*$. An event-log $\mathcal{L}$ is here modeled as a function that, given an identifier i of a log trace returns the sequence of events related to the process instance with the identifier i, i.e. $\mathcal{L} : \mathcal{I} \rightarrow \mathcal{E}^*$.*[1]

Given an event $e = (a, r, v)$, the remainder uses the following shortcuts: $activity(e) = a$, $resource(e) = r$ and $variables(e) = v$. Also, given a trace $\sigma = \langle e_1, \ldots, e_n \rangle$, $prefix(\sigma)$ denotes the set of all prefixes of σ, including σ, namely $prefix(\sigma)=\{\langle\rangle, \langle e_1 \rangle, \langle e_1, e_2 \rangle, \ldots, \langle e_1, \ldots, e_n \rangle\}$.

For building our recommender system, we need to define what we aim to optimize, i.e. the goal of our recommendation: hereafter, this is named Key Performance Indicator (KPI) and depends on the specific process domain.

Definition 3 (KPI Function). *Let $\mathcal{E}$ be the universe of events. A Key Performance Indicator (KPI) is a function $\mathcal{K} : \mathcal{E}^* \times \mathbb{N} \rightarrow \mathbb{R}$ such that, given a (prefix of a) trace $\sigma \in \mathcal{E}^*$ and an integer $1 \leq i \leq |\sigma|$,[2] $\mathcal{K}(\sigma, i)$ returns the KPI value of σ after the occurrence of the first i events.*

Therefore $img(\mathcal{K})$ is the set of all possible KPI values. With abuse of notation, we indicate $\mathcal{K}(\sigma) = \mathcal{K}(\sigma, |\sigma|)$, namely the KPI value after the occurrence of events in trace σ. Note that our KPI definition is assumed 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 the occurrence of each event, i.e. after each activity execution. We aim to be generic and account for all relevant domains. Given a trace $\sigma = \langle e_1, \ldots, e_n \rangle$ that records a complete process execution, the followings are examples of two potential KPI definitions:

- *Total Time.* We opted to consider the task in which the objective is to reduce the total time. Given a σ's prefix of i events, $\mathcal{K}_{total}(\sigma, i)$ measures the difference between the timestamp of the trace's last future event and the first event's timestamp.
- *Activity Occurrence.* It measures if a certain activity is going to occur in the future, such as an activity eventually *Open Loan* in a loan-application process. The corresponding KPI definition for the occurrence of an activity a is $\mathcal{K}_{occur\_a}(\sigma, i)$, which is equal to 1 if the activity a occurs in $\langle e_{i+1}, \ldots, e_n \rangle$, 0 otherwise.

[1] The operator * refers to the Kleene star: given a set A, A^* contains all the possible finite sequences of elements belonging to A.
[2] Given a trace σ, $|\sigma|$ indicates the number of events in σ.

Table 1. Example of the output of the Prescriptive Analytics Oracle Function in a tabular form, for a given trace when the KPI is the total time of a case. It provides the recommended activity *Back-Office Adjustment Requested* associated with a set of pairs of resources and delta in KPI. For instance, if the resource $BOCSER$ executes an adjustment to the Back-Office, the expected total time of the procedure will decrease by 195 h.

Activity	Resource	Δ
Back-Office Adjustment Requested	CE_UO	208 h
	$BOCSER$	195 h
	BOC	112 h
	...	...

The goal of the recommender system is to provide recommendations on both the activities to be performed and the resources best suited to perform them, with the aim of enhancing the final outcome of running process instances in terms of the identified KPIs. To achieve this, a **Prescriptive Analytics Oracle Function** must be developed. This function will enable the prediction of the KPIs of the final outcome of a running process instance, and will identify the best activity to be performed and the most suitable resource to perform it.

Definition 4 (Prescriptive Analytics Oracle Function). *Let $\mathcal{E}$ be the universe of events and $\sigma \in \mathcal{E}^*$ a (running) trace belonging to it, $\mathcal{A}$ the set of possible activities, $\mathcal{K} : \mathcal{E}^* \times \mathbb{N} \rightarrow \mathbb{R}$ a KPI function and $\mathcal{R}$ the set of the possible resources. A Prescriptive Analytics Oracle Function is a function $\psi : \mathcal{E}^* \times \mathcal{K} \rightarrow \mathcal{A} \times 2^{(\mathcal{R}\times\mathbb{R})}$ such that $\psi(\sigma, \mathcal{K})$ returns $(a, \{(r_1, \Delta_1), \ldots, (r_m, \Delta_m)\})$ with $m \leq |\mathcal{R}|$ to indicate that activity a is recommended and, if performed by r_i, will lead to a Δ_i improvement of KPI $\mathcal{K}$. Also, $\forall i, j \in \{1, \ldots, m\}$, $r_i = r_j \iff i = j$, meaning that a resource can only be recommended once.*

Since not all resources can perform the recommended activity a, the number m of recommended resources does not necessarily coincide with the number $|\mathcal{R}|$ of all resources [15]: also, some resources might not be available at a certain point for other reasons, e.g. on holidays or on sick leave.

In the example in Table 1, the oracle function ψ takes as input the KPI function as defined in Definition 3, with $\mathcal{K}$ modelling the KPI . For a certain trace, the recommended activity is *Back-office Adjustament Requested*. The function also returns a set of pairs (r, Δ) to indicate that, if the activity is performed by resource r, the final KPI is predicted to change by Δ. Concretely, if the activity *Back-Office Adjustment Request* is performed, e.g., by `CE_UO`, the total time will reduce by `208hours`.

In the remainder, we will make use of a helper function $max_{\psi,\mathcal{K}}(\sigma)$ that for each (running) trace $\sigma \in \mathcal{E}^*$ returns the maximum achievable improvement.

Definition 5 (Helper function). *Let $\mathcal{E}$ be the universe of events. $\sigma \in \mathcal{E}^*$ a (running) trace belonging to it, and $\mathcal{A}$ the set of possible activities. Let $\psi(\sigma, \mathcal{K}) = (a, \{(r_1, \Delta_1), \ldots, (r_m, \Delta_m)\})$ the Prescriptive Oracle Function. The Helper function $max_{\psi,\mathcal{K}} : \mathcal{E}^* \rightarrow \mathcal{A} \times \mathbb{R}$ is a function that returns a pair $(a, \Delta) = (a, max(\{\Delta_1, \ldots, \Delta_m\}))$*

The oracle function can be implemented in multiple ways, using several of the prescriptive-analytics algorithms in literature (cf. Sect. 2). This paper does not aim to propose any specific prescriptive-analytics algorithms. However, for the implementation and testing, we opted to use the prescriptive-analytics proposal discussed in [7], which has been extended to also return the pairs of resources and KPI's deltas.[3]

4 Global Activity-Resource Allocation

The purpose of this paper is to provide resources with tailored recommendations regarding which actions to take and to which process instance, while also allowing for a degree of choice and autonomy. To achieve this goal, it is essential to establish a framework that can generate interdependent recommendations while simultaneously accommodating the individual decision-making processes of the resources involved. It is important to note that the best recommendation for a case, when viewed from the perspective of optimising a specific KPI, may not necessarily be the best recommendation for that case in the context of global optimisation. In the task of our work, the KPI to be optimised is not pointwise: an individual case may get the best recommendation for him, but this makes a resource busy and so unavailable for other cases that could improve their KPI more. Hence, we want to optimise the sum of the KPIs for all resources and cases on which they act, ensuring the single recommendations provided to resources interact with each other without conflict. This leads to the definition of a **Profile**.

Definition 6 (Profile). *Let $\mathcal{L} : \mathcal{I} \rightarrow \mathcal{E}^*$ be an event log, $\mathcal{A}$ the set of its possible activities, and $\mathcal{R}$ the set of the possible resources. A profile $\mathcal{P} \subset (\mathcal{I} \times \mathcal{A} \times \mathcal{R} \times \mathbb{R})$ is defined as a set of tuples (i, a, r, Δ) where for the (running) case with identifier i, the activity a is to be assigned to the resource r for improving its expected KPI of Δ. There is an additional constraints: there cannot be two tuples with the same case identifier or the same resource.*

A profile aims to allocate the set of available resources to a set of cases with the aim of improving the overall KPI values for all running cases. The generation of a profile is challenging: it is a combinatorial problem that would require one to potentially try every combination of case ids, activities, and resources. This is practically unfeasible. In Sect. 4.1, we illustrate a greedy algorithm to compute a profile.

[3] Code available at https://github.com/Pado123/prescriptive_global_optimization.

Id	Activity	ΔKPI
DD-45678	Pending Liquidation Request	93434 h
CC-34567	Back-Office Adjustment Requested	85014 h
BB-23456	Pending Liquidation Request	42543 h
...	...	...

Id	Activity	Resource	ΔKPI
DD-45678	Pending Liquidation Request	BOCSER	93434 h
BB-23456	Pending Liquidation Request	BOC	21944 h
CC-34567	Back-Office Adjustment Requested	CE_UO	10433 h
...	...	...	...

Fig. 1. The table on the left is an example of tabular form of the $\Delta RANK$ sequence: the columns shows the case ids, the recommended activities for the respective cases, and the maximum KPI improvement. In this example, the employed KPI is the case total time. The profile is obtained from $\Delta RANK$ by allocating resources to cases (see the right-hand side table). Since the best resource cannot be assigned to every case, the assigned resource might cause a drop in the KPI's improvements.

The creation of a single profile is also poorly applicable in practice because it would impose activities to resources, without considering external factors. The novelty of our framework is also linked to providing process actors with some degree of freedom, while still aiming to improve the overall KPI. This requires generating several profiles: different profiles assign different resources to a certain resource. A resource can pick one of the activities available for him/her in any of the generated profiles.

The resource's choice naturally filters out profiles that are incompatible with the choice made. The subsequent resource to choose will then have fewer profiles according to which to choose. Section 4.2 illustrates how to generate the profiles additional to the first.

4.1 Generation of the First Profile

To create an initial profile $\mathcal{P}_0$, we first create a sequence $\Delta RANK \subseteq (\mathcal{I} \times \mathcal{A} \times \mathbb{R})^*$. It can be constructed using the Helper function defined in Definition 5 as follows. First, we build the set of triples $(i_1, a_1, \Delta_1), \ldots, (i_n, a_n, \Delta_n) = \bigcup_{i \in dom(\mathcal{L})}(i, max_{\psi,\mathcal{K}}(\mathcal{L}(i))$, which later are sorted descending by the third component, namely $\Delta_1, \ldots, \Delta_n$. An example of $\Delta RANK$ is given in the left-hand side table in Fig. 1.

The first profile $\mathcal{P}_0$ is obtained by extending $\Delta RANK$ with resources (cf. the right-hand side table in Fig. 1). To achieve this, we start from the first element $(i_1, a_1, \Delta_1) \in \Delta RANK$, i.e. the one with the greatest expected improvement. Then, we evaluate $\psi(\mathcal{L}(i_1), \mathcal{K}) = (a_1, \{(r_1^1, \Delta_1^1), \ldots, (r_1^m, \Delta_1^m\}))$, with $\mathcal{K}$ be the KPI function of interest, and we associate resource r_1^1 to (i_1, a_1, Δ_1) the first pair (r_1, Δ_1), thus resulting to add $(i_1, a_1, r_1, \Delta_1^1)$ to the profile. Resource r_1 is removed from the set $\mathcal{R}$ of the resources available.

We then move to the second element $(i_2, a_2, \Delta_2) \in \Delta RANK$, and evaluate $\psi(\mathcal{L}(i_2), \mathcal{K}) = (a_2, \{(r_2^1, \Delta_2^1), \ldots, (r_2^q, \Delta_2^q\}))$. If $\{r_2^1, \ldots, r_2^q\} \cap \mathcal{R} = \emptyset$, no element is added to profile $\mathcal{P}_0$ for instance i_2. Otherwise, we look for the smallest j such

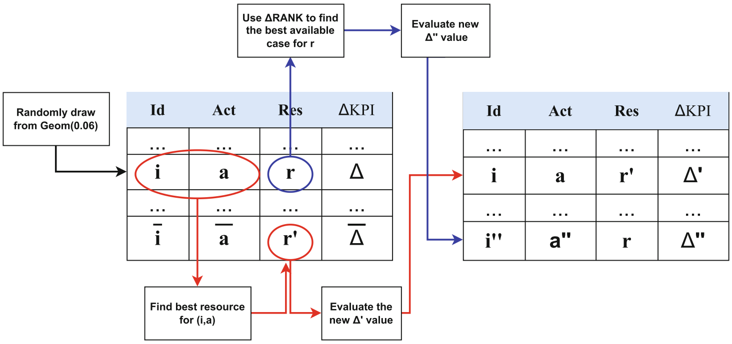


Fig. 2. A visual representation of the algorithm to perturb profiles. The right table depicts the original profile, while the left table shows the perturbed profile. A random element $p = (i, a, r, \Delta)$, according to a geometric distribution. The algorithm identifies the best new resource r' for the corresponding trace identifier i and activity a, resulting in a new element $p' = (i, a, r', \Delta')$ (indicated in red in the picture). The resource r' is unassigned from the previous assignment: the element for r' is thus removed from the profile $(\bar{i}, \bar{a}, r', \bar{\delta}$ in figure). Resource r is free, and is given a different assignment (element i'', a'', r, Δ'' in figure). (Color figure online)

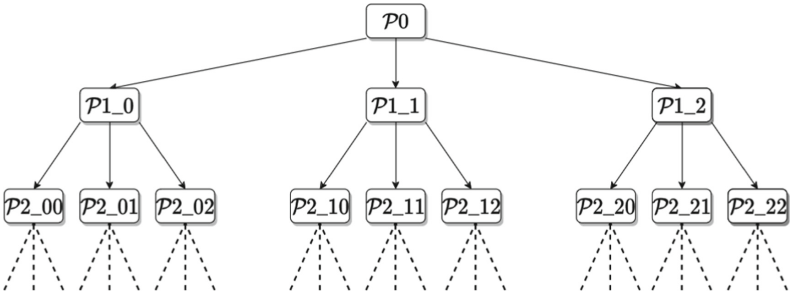


Fig. 3. Schematic of how profiles are generated via perturbation. Starting from the initial profile $\mathcal{P}_0$, a number of perturbations are created (three for the example in figure), namely: $\mathcal{P}_{1\_0}$, $\mathcal{P}_{1\_1}$ and $\mathcal{P}_{1\_2}$. This is then repeated for each of the obtained profiles, until a certain number of profiles are overall constructed. If a perturbation generates a profile that has already been created, this is discarded.

that $r_2^j \in \mathcal{R}$. Tuple $(i_2, a_2, r_2^j, \Delta_2^j)$ is added to profile $\mathcal{P}_0$. Note that Δ_2^j might be lower that Δ_2 because the allocated resource might not yield the maximum improvement: by construction, it is only guaranteed that $\Delta_2 = \Delta_2^1$. Resource r_2^j is removed from $\mathcal{R}$. This procedure is repeated for every tuple in $\Delta RANK$, as long as set $\mathcal{R}$ is not empty (i.e. activities and cases can be allocated to resources).

4.2 Generation of Additional Profiles

The first profile $\mathcal{P}_0$ is certainly the valuable starting point, but it falls short in two main aspects. It is generated considering the traces in the descending order of potential improvements: it is in fact a greedy approach, which might still returned solutions relatively far from the potential, optimal solution. Using approaches based on local search, solution $\mathcal{P}_0$ is perturbed to obtain more solutions of profiles. As discussed, we want to grant freedom to resources on the choice of which cases (and consequently activities) to work on as next: therefore, all profiles generated by perturbation are retained.

Figure 2 illustrates how one profile $\mathcal{P}$ is perturbed into $\mathcal{P}'$: elements are visualized in tabular form. Initially $\mathcal{P}' = \mathcal{P}$. The elements in $\mathcal{P}$ are sorted by descending values of the KPI improvement (see column ΔKPI in figure). An element $p = (i, a, r, \Delta)$ is randomly selected from the sorted list according to a geometric distribution with $p = 0.06$. Let $p = (i, a, r, \Delta)$ the selected element, with oracle function $\psi(\mathcal{L}(i), K) = (a, \{(r_1, \Delta_1), \ldots, (r_m, \Delta_m\})$. Elem p is removed from $\mathcal{P}'$, while we add a certain $p' = (i, a, r', \Delta')$ such that, if $r = r_1$, then $r' = r_2$ an $\Delta' = \Delta_2$, otherwise $r' = r_1$ and $\Delta' = \Delta_1$. Since every resource is assigned to the activity of some case, $\mathcal{P}$ contains some element for r': $\overline{p} = (\overline{i}, \overline{a}, r', \overline{\Delta}) \in \mathcal{P}$. Tuple $\overline{p}$ is removed from $\mathcal{P}'$. Resource r is now free: we pick the top element $(i'', a'', \Delta'') \in \Delta RANK$ such that r is allowed to execute a'' and there is no element in the $\mathcal{P}'$ that refers to the case with id i''. Element (i'', a'', r, Δ'') is added to $\mathcal{P}'$.

In sum, the above procedure is able to perturb a profile and, thus, create a new one. This is iterated, until a given target of profiles is created. This can be visualized as in Fig. 3: we started from the initial profile $\mathcal{P}_0$. A certain number of perturbations is created from $\mathcal{P}_0$: profiles $\mathcal{P}_{1\_0}$, $\mathcal{P}_{1\_1}$ and $\mathcal{P}_{1\_2}$ in figure with three perturbations. This is then repeated for each of the obtained profiles. In general, two subsequent perturbations can result in the original profiles; however, we discard the perturbed profiles that were already previously obtained. This motivates why Fig. 3 has a tree-like structure, in place of a graph-like.

4.3 Assign Recommendations

Once we generate the entire set of profiles, we create a Profiles Ranking $\mathscr{P}$, by sorting them down by global KPI improvements (i.e., summing up the KPI improvements for all elements in the profiles).

which is used to effectively provide recommendations to resources. In fact, in organizational reality, they receive the range of choices provided for them based on the profile's order in which the Profiles Ranking $\mathscr{P}$ has been sorted. At the end of the assignment procedure, every resource r will have selected an activity a and a case identifier i, resulting in a final set that, from this point onwards, we will refer to it as **Resource-task Assignment Set** $\mathcal{S} \subset (\mathcal{I} \times \mathcal{A} \times \mathcal{R} \times \mathbb{R})^{|\mathcal{R}|}$ where $\mathcal{R}$ is the set of available profiles.

Once the first resource $\overline{r}$ has to select its task, the profiles in $\mathscr{P}$ are scanned till 3 different pairs activity-identifier may be assigned to it. This allows the system to provide the resource (at most) three different choices. Then $\bar{r}$ picks the case with identifier $\bar{i}$, and $\bar{r}$ executes the accordant activity $\bar{a}$. At that time, we remove all profiles in the set $\mathscr{P}$ in which the element $(\bar{i}, \bar{a}, \bar{r}, \overline{\Delta})$ is not present for some $\overline{\Delta}$. Then, the other resources can pick activities in order according to the retained profiles. This procedure, called **Exact Assignment (EA)**, provides a certain degree of freedom to the first resources, which can go quickly down as more and more resources pick cases for performance.

To overcome this problem, an alternative framework called **Approximate Assignment (AA)** is proposed: the only difference is that no profile is removed from $\mathscr{P}$. So the resulting Resource-task Assignment Set $\mathcal{S}$ may not coincide to those of any profile in $\mathscr{P}$. On the Approximate Assignment procedure, when the first resource $\bar{r}$ makes a choice of case with id $\bar{i}$ and activity $\bar{a}$, no profile is removed from $\mathscr{P}$. When the second resource makes a choice, (s)he presented the three best options in $\mathscr{P}$ without considering identifiers related to cases previously selected. Approximate assignment thus provides further freedom of choice, at the cost of potentially a lower global KPI improvement

5 Evaluation

The evaluation focuses on assessing the overall KPI improvements for two case studies (see Sect. 5.1). In particular, the comparison is done with respect to existing approaches, with specific emphasis on the work by Dumas et al. [14] where the framework exhibits a similar operational approach to that outlined within this documentation, although lacking the provision of multiple profiles, thereby terminating at the first greedy solution. We also focus on quantifying the degree of freedom, specifically assessing the extent to which resources possess the ability to choose their task, even if limited to a choice between two feasible options. This freedom's degree is a novel of our approach if compared with the state of the art. It indeed allows resources to pick among multiple alternatives.

Section 5.2 details the procedure for partitioning event logs into a training log and a test log. The training log is used to train the oracle function ψ, which plays a crucial role in our proposed methodology. Meanwhile, the test log is employed to evaluate the performance and effectiveness of our approach.

Subsequently, in Sect. 5.3, we discuss the assessment of recommendation quality and the level of resource autonomy achieved. Furthermore, Sect. 5.4 presents our evaluation method, which involves comparing the outcomes to a real-world scenario. Lastly, in Sect. 5.5, we analyze and interpret the results generated by our methodology.

5.1 Introduction to Use Cases

The validity of our approach was assessed using two different event logs with their associated use case. The first is so-called **Bank Account Closure** (BAC), a log referring to an Italian Bank Institution process that deals with the closures of bank accounts. From the bank's information system, we extracted an event log containing 212,721 events containing 15 activities, 654 resources and 32,429 completed traces, divided into 14,593 for train and 17,836 for testing. For this log, we opted to consider the task in which the objective is to reduce the execution time of the instances, i.e. the KPI function $\mathcal{K}$ is equal to *Total Time* and the total number of generated profiles is 650,000.

The BPI challenge used the second log in 2013[4]. It is provided by Volvo Belgium and contains events from an incident and problem management system called **VINST**. We extracted 7,456 completed traces and 64,975 events. It contains 13 different activities that can be accomplished with 649 resources. In selecting traces from the log for training and testing, we get a training log of 3,355 traces and a test log of 4,101 traces.

For this case, we aim to avoid the occurrence of the activity *Wait-User*. The KPI value can be 1 or 0 if the activity occurs or not, while the Δ values related to the oracle function are evaluated as the difference by the probability of the activity occurring (i.e. $\Delta \in [0, 1]$). Note that one wants to reduce the activity-occurrence probability: the activity *Wait-User* is considered detrimental in terms of time and customer satisfaction. The total number of generated profiles is 140,000.

5.2 Train-Test Splitting Procedure

The starting point for an evaluation is an event log $\mathcal{L}$. In this section, to lighten the notation, we refer to $dom(\mathcal{L})$ as $\mathcal{L}$, and so referring to a log not as a function but as a set of trace identifiers in its domain. We first extract the training log $\mathcal{L}^{comp}$ for training the oracle function and, consequently, the recommender system. Then, we aim at creating the log $\mathcal{L}^{run}$ used for testing our system. To extract the training log $\mathcal{L}^{comp} \subset \mathcal{L}$ we compute the earliest time t_{split} such that 45% of the identifiers related to traces of $\mathcal{L}$ are completed. This allows us to define $\mathcal{L}^{comp}$ as the set of traces of $\mathcal{L}$ completed at time t_{split}, and consequently, define $\mathcal{L}^{run}$ as $\mathcal{L} \setminus \mathcal{L}^{comp}$. The traces of $\mathcal{L}^{run}$ are then truncated to a set $\mathcal{L}^{trunc}$ obtained from $\mathcal{L}^{run}$ by maintaining only a random percentage of events in each trace[5], this has been done for simulating running instances to which provide recommendations, using the set $\mathcal{L}^{run}$ for the evaluation of them.

5.3 Evaluation Metrics

The accuracy of recommending the resource r performs the activity a for the running case with identifier $i' \in dom(\mathcal{L}^{trunc})$ is evaluated as the average KPI of traces similar to it. Analytically, if $\mathcal{L}(i') = \sigma'$ and e such that $activity(e) = a$ and $resource(e) = r$:

$$score(\sigma', e) = avg_{\sigma \in Sim(\sigma', e, \mathcal{L}^{run})} \mathcal{K}(\sigma) \tag{1}$$

[4] https://www.win.tue.nl/bpi/doku.php?id=2013:challenge.

[5] The random percentage p was drawn from a uniform distribution $\mathcal{U}[25, 75]$, repeating the experiment for its stochastic validity.

where $Sim(\sigma', e, \mathcal{L}^{run})$ is the set of traces similar to $\sigma' \oplus \langle e \rangle$, namely

$$\begin{aligned} Sim(\langle e'_1, \ldots, e'_m \rangle, e, \mathcal{L}^{run}) = \{\sigma \in \text{cod}(\mathcal{L}^{run}) : \exists \sigma^p = \langle e_1, \ldots, e_{m+1} \rangle \in \text{prefix}(\sigma), \\ (\mathit{activity}(e_{m+1}), \text{resource}(e_{m+1})) = (\text{activity}(e), \text{resource}(e)), \\ \text{activity}(e_i) = \text{activity}(e'_i) \ \forall i \in \{1, \ldots, m\}\} \end{aligned}$$

The score of the recommended action a to a resource r performing the running trace σ' is so the average KPI of traces similar to σ' for which the activity a has been performed by the resource r. This procedure is similar to the one used by de Leoni et al. in [2] and by Padella et al. in [7], adding the constraint about the recommended resource r to the similarity concept.

Typically, in the machine learning literature, the dimension of the train set is larger than the dimension of the test set. We chose this split ratio because using the accuracy evaluation proposed in Eq. 1, we evaluate a mean value on the output set of the function Sim that embodies the constraints related to the resource and the activity: this may lead to a small number of items on it, making the mean value evaluated statistically not significant.

As already mentioned, we also aim to give resources freedom in choosing which case and, consequently, activity to work on as next. Therefore, in our experiments, our goal is also to measure the resource freedom, hereafter defined as the number of resources that have given the freedom to choose the case to work on within a set that contains at least two cases. On this aim, we introduce the concept of **Freedom Score**, that is the ratio between the resources that had the possibility of choosing between at least two case-activity options in our assignment procedure (cf. Sect. 4.3) and the number of resources that can act on more than two running cases of $\mathcal{L}^{run}$. Analytically

$$Freedom\ Score(Assignment) = \frac{|\{r \in \mathcal{S} : r\ has\ chosen\ in\ Assignment\}|}{|\{r \in \mathcal{S} : r\ can\ act\ on\ more\ than\ one\ i \in\ dom(\mathcal{L}^{run})\}|} \quad (2)$$

The purpose of this function is to assess the degree of freedom of choice afforded to the resources by comparing it to the level of choice they typically have. A Freedom Score of 100% indicates that resources are granted complete freedom, while a score of 0% corresponds to no freedom.

5.4 Evaluation Methodology

The assessment of the system was carried out by trying to replicate actual organizational conditions. Therefore, we want to simulate how resources realistically interact with a recommender system.

1. Not all resource work at the same time, due to various factors such as shifts, vacations or other circumstances. Therefore, a Bernoulli distribution with a parameter of $p = 0.75$ is used to stochastically select a subset of resources: each individual element in the complete set of resources has a 75% probability of being designated as active and thus included in the subset.
2. Not all resources pick up a case to work on the same time, then we randomly shuffled the list of resources obtained at point 1, generating random arrival orders randomising the order in which resources pick their task.

Table 2. In the second column, the table presents the time values associated with the generation of the complete Profiles Ranking $\mathscr{P}$, representing approximately 10% of the total number of profiles that the framework can generate. The third column displays the absolute count of the generated profiles.

Case Study	Time needed	Total number of generated profiles
Total time on BAC	1 h and 40 min	650,000
Wait-User Occurrence on VINST	40 min	140,000

3. Resources are provided with a ranking of cases allowed to work on, ordered by expected KPI improvement. However, they do not necessarily pick the top element: research in Human-Computer Interaction has demonstrated a consistent pattern of user behavior when presented with a ranked list of options, as documented in [16]. In line with this study, we have adopted a stochastic resource selection behavior: Specifically, the probabilities of selecting the first, second, and third options are 61%, 24%, and 15%, respectively.

Since the points 1–3 in the list above rely on sampling from distributions (e.g. the Bernoulli distribution at point 1), the procedure has been repeated: we extracted 10 values from the Bernoulli distribution described at point 1 and, for each of these values, the random shuffling has been done 10 times. It follows that, in total, we repeated the evaluation 100 times.

The improvements by our framework has been evaluated by applying the formula in Eq. 1 to the recommendations provided to the traces relatives to the identifiers in $\mathcal{L}^{trunc}$ using the two assignment procedures defined in Sect. 4.3 and then comparing this scores with the real process executions from $\mathcal{L}^{run}$.

5.5 Results Analysis

For each of the 10 subsets of existing resources obtained at point 1 of the evaluation methodology (cf. Sect. 5.4), we computed the total number of potential profiles. However, the time needed to compute them all is practically not feasible and hence we use our framework to only generate up to 10% of them in experiments, with increments of 1%.

Due to differences in the number of activities, resources, and cases in the logs, the computational times varied. All the generations were executed on a workstation equipped with a 16-core AMD Ryzen 7 4700G processor unit and 16 GB RAM, which were divided into 12 different threads. Table 2 shows for each case study, the time to generate this 10% of profiles. This threshold represents a justifiable value since the tree procedure described in the Sect. 4.2 follows a greedy approach: on it, profiles are initially generated in a stochastic manner and subsequently filtered to eliminate duplicates. As the number of generated profiles augments, the likelihood of encountering new profiles decreases, leading to an exponential rise in the time required for generating new profiles.

The experiments' results in terms of KPI values are shown in Fig. 4, which illustrates how the improvement is linked to the percentage of profiles that are

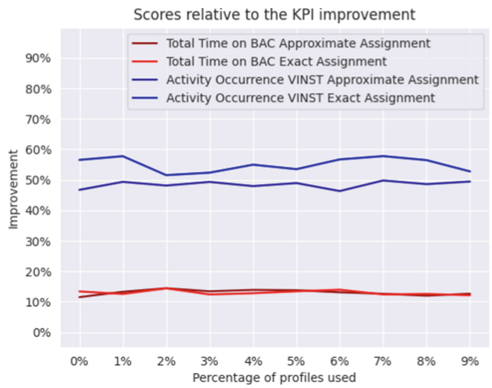


Fig. 4. Results related to the KPI improvement for both case studies and the two assignment techniques. On the x-axis there is the percentage of the profile used for running the two algorithms, while in the y-axis the average KPI improvement on the whole $\mathcal{L}^{run}$ evaluated as defined in Eq. 2 is shown.

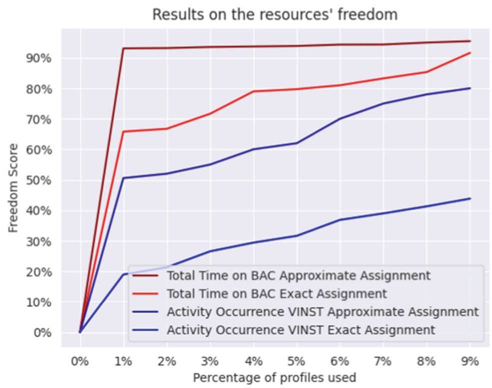


Fig. 5. Results related to the freedom left to the resources for both case studies and the two assignment techniques. On the x-axis there is the percentage of the profile used for running the two algorithms, while on the y-axis the Freedom Score as defined in Eq. 2 is shown.

generated, for the BAC and VINST case studies, and using our two approaches to compute the set of profiles. The results show that it is sufficient to generate few profiles to obtain significant KPI improvements. For the case study of reducing total execution time on BAC, there is no significant variation in outcomes between the Exact and Approximate assignment techniques. In the conducted case study focusing on optimizing process performance in the BAC system, we achieved an improvement of 58%. This improvement translates to a reduction in the total execution time of all active traces from 179, 430 h to 76,873 h. We successfully minimized the overall processing time by implementing the proposed measures, leading to significant efficiency gains. Furthermore, our investigation targeted the reduction of the *Wait-User* activity occurrences within the VINST dataset. The initial analysis identified 631 traces in which this activity took place. Through the implementation of optimized strategies, the occurrence of *Wait-User* decreased to 486 traces. This reduction highlights the effectiveness of the proposed approach in streamlining the process and minimizing potential bottlenecks associated with this specific activity of 12%.

A larger number of profile generation may still remain relevant to allow resources a larger degree of freedom to choose the case, and hence the intervention, to work on. Figure 5 shows how the Freedom Score increases with larger number of profiles that are generated. The algorithm for approximate assignments seem to consistently allow larger freedom. It even achieves a Freedom Score of 90% after generating only 1% of the profiles for the BAC case study.

The Approximate Assignment method was indeed designed to provide resources with more freedom of choice, a goal successfully achieved in both case studies. The approximate assignment was also able to achieve the same amount of KPI improvement as the exact assignment: therefore, the Approximate-Assignment algorithm is certainly preferable for the BAC case study. For the VINST case study, the Exact-Assignment method provides results that are around 10% better than the Approximate-Assignment method, which suggests the Exact Assignment method is preferable.

It follows that opting for the Exact-Assignment or for the Approximate-Assignment method may depend on the case study. Therefore, the choice requires to conduct a prior assessment based on training and testing phases, as conducted in the experiments discussed in this paper.

Last but not least, *we aim to compare our results with those obtained by the latest advantages in prescriptive process analytics, and we have carried on a comparison with respect to the approach by Dumas et al.* [14]*. The approach by Dumas et al. corresponds to the scenario in which only the first profile is generated, analogously to what discussed in Sect.* 4.1*. The conducted experiments have shown that the creation of multiple profiles and their ranking provides a further improvement by 6.2% and 1.9% for the VINST and BAC case studies, respectively.* It is worthwhile noting here that Dumas et al. use a different prescriptive-analytics oracle function. However, a fair comparison requires to use the same oracle function, to put aside any difference due to the choice of the oracle function. This motivates why we use our oracle function in both of scenarios, namely only using the first profile, or conversely leveraging on the profile ranking.

6 Conclusions

Process-aware Recommendation systems are a class of information systems that provide support to process stakeholders to achieve better results for the running cases. The module that suggests effective interventions is obviously the core module in this class of systems. The intervention for a running case typically consists in suggesting a certain activity to be performed as next, as well as the resource to which this activity should be given for performance. Existing techniques propose interventions to single running cases in isolation, making the choice of interventions local to single cases. However, resources are shared among cases, and hence an allocation of resources and interventions should be deal with as a global optimization problem, where all cases requiring interventions are considered altogether.

This paper has put forward a framework that tackles the global optimization problem. It is clear that the complexity of the problem is NP-hard, and hence finding an optimal solution is intractable when hundreds of cases are running at the same time, and also hundreds of resources are involved. We thus propose two approximated algorithms that aim to find sub-optimal solutions. The algorithm returns a number of alternative user profiles, each of which consists in a set of assignments of activities to resources, with the constraint that a resource can only work on with a case within a profile. These profiles are then ranked with the expected outcome improvement, measured in terms of KPIs. Each resource is then offered the interventions ordered by descending ranking of the profile of which those interventions are part.

Among the advantages of our proposal, it is worthwhile mentioning that, while most of existing approaches impose an assignment of cases and activities to resources, we provide process actors with a certain degree of freedom in choosing what to work on. This freedom is clearly very beneficial in the context of recommender systems: vice versa, imposing an assignment may potentially incur in the risk of having resources to act on cases independently and regardless of the recommendations.

Experiments were conducted with two real-life case studies, emulating how humans would pick offered interventions in an order list. This emulation was based on behavioral models described in the human-computer interaction literature [16]. The results illustrate a significant improvement with respect to frameworks that aim to improve the outcome of running cases in isolation. So did we compare with the approach by Dumas et al. [14], which is the only approach that we found that is able to provide a global KPI improvement: our framework provides a further improvement by 6.2 and 1.9% for the two case studies.

Acknowledgement. The PhD. scholarship of Mr. Padella is partly funded by IBM Italy, and by the BMCS Doctoral Program, University of Padua. This research is also supported by the Department of Mathematics, University of Padua, through the BIRD project "Data-driven Business Process Improvement" (code `BIRD215924/21`).

References

1. Comuzzi, M.: Ant-colony optimisation for path recommendation in business process execution. J. Data Semantics **8**(2), 113–128 (2019)
2. de Leoni, M., 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) (2020)
3. Weinzierl, S., Dunzer, S., Zilker, S., Matzner, M.: Prescriptive business process monitoring for recommending next best actions. In: Business Process Management Forum (2020)
4. 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
5. Fahrenkrog-Petersen, S., Tax, N., Teinemaa, I., Dumas, M., de Leoni, M., Maggi, F., Weidlich, M.: Fire now, fire later: alarm-based systems for prescriptive process monitoring. Knowl. Inf. Syst. **64**, 02 (2022)
6. 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) (2021)
7. Padella, A., de Leoni, M., Dogan, O., Galanti, R.: Explainable process prescriptive analytics. In: 2022 4th International Conference on Process Mining (ICPM), pp. 16–23 (2022)
8. Shapley, L.S.: A value for n-person games. RAND Corporation, no. 28 (1953)
9. Cabanillas, C., Schönig, S., Sturm, C., Mendling, J.: Mining expressive and executable resource-aware imperative process models. In: Gulden, J., Reinhartz-Berger, I., Schmidt, R., Guerreiro, S., Guédria, W., Bera, P. (eds.) BPMDS/EMMSAD -2018. LNBIP, vol. 318, pp. 3–18. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91704-7_1
10. Havur, G., Cabanillas, C.: History-aware dynamic process fragmentation for risk-aware resource allocation. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) OTM 2019. LNCS, vol. 11877, pp. 533–551. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33246-4_33
11. Zhao, W., Yang, L., Liu, H., Wu, R.: The optimization of resource allocation based on process mining. In: Huang, D.-S., Han, K. (eds.) ICIC 2015. LNCS (LNAI), vol. 9227, pp. 341–353. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22053-6_38
12. Huang, Z., van der Aalst, W., Lu, X., Duan, H.: Reinforcement learning based resource allocation in business process management. Data Knowl. Eng. **70**(1), 127–145 (2011). https://www.sciencedirect.com/science/article/pii/S0169023X1000114X
13. Park, G., Song, M.: Prediction-based resource allocation using LSTM and minimum cost and maximum flow algorithm. In: International Conference on Process Mining (ICPM) 2019, pp. 121–128 (2019)
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.) Business Process Management Forum, pp. 207–223 Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16171-1_13

15. de Leoni, M.: Foundations of Process Enhancement, pp. 243–273. Springer, Cham (2022)
16. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR 2005, pp. 154–161. Association for Computing Machinery, New York (2005)

Zooming in for Clarity: Towards Low-Code Modeling for Activity Data Flow

Ali Nour Eldin[1,2](✉), Jonathan Baudot[2], and Walid Gaaloul[1]

[1] Telecom SudParis, Institut Polytechnique de Paris, Palaiseau, France
{ali.nour_eldin,walid.gaaloul}@telecom-sudparis.eu
[2] Bonitasoft, Grenoble, France
{ali.nour-eldin,jonathan.baudot}@bonitasoft.com

Abstract. Business Process Modeling and Notation (BPMN) is a widely used standard workflow language for modeling business processes. However, there is a growing need to integrate data and process models to enable a more holistic view of business processes to reducing implementation time through a clear understanding of the mode, and BPMN has limitations in representing data-related concepts. To address this, we propose an extension to BPMN called DataFlow BPMN (DF-BPMN), which is a low-coding visual solution, for modeling and analyzing the relationship between process and data. Low-code is a growing development approach supported by many platforms. It fills the gap between business and developer. Indeed, it enables quick generation and delivery of business applications with minimum effort. We introduce the Activity DataFlow, an extension of activity that allows zooming into the data manipulated within it, which enable different levels of granularity: control-flow perspective and data perspective. Additionally, we developed a tool for creating a model with the DF-BPMN. Our approach has been evaluated quantitatively and qualitatively, and the results demonstrate that DF-BPMN offers significant advantages over BPMN.

Keywords: BPMN · Low-Code · Process Modeling · data perspective · data models

1 Introduction

A business process is a set of activities within an enterprise that follows a defined logical order and dependency, with the objective of producing a desired result. Process models provide a comprehensive understanding of a process [3], and can be understood from different perspectives [2]. The *control-flow perspective* describes tasks and their execution order through different constructors, which can be modeled using BPMN [8]. The *data perspective* deals with business and processing data, similar to UML [9]. Integrating these perspectives within the

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 267–282, 2023.
https://doi.org/10.1007/978-3-031-41623-1_16

same workflow modeling language is essential for success, as it enhances information representation, reduces implementation time, and enables effective decision-making and resource allocation.

Low-Code Development Platforms (LCDPs) provide development environments for creating software applications using visual interfaces, rather than traditional manual coding methods. These platforms use process modeling languages to define business processes and automate related tasks, such as data modification, scheduling, or linking data flows to external services. Process executions help bridge the gap between business and developer [10]. Figure 1 illustrates the steps involved in LCDP development compared to the Business Process Lifecycle: (i) language specification, (ii) language compilation, and (iii) execution and monitoring. Furthermore, BPMN elements are commonly used as the language specification in LCDP-based business processes [10].

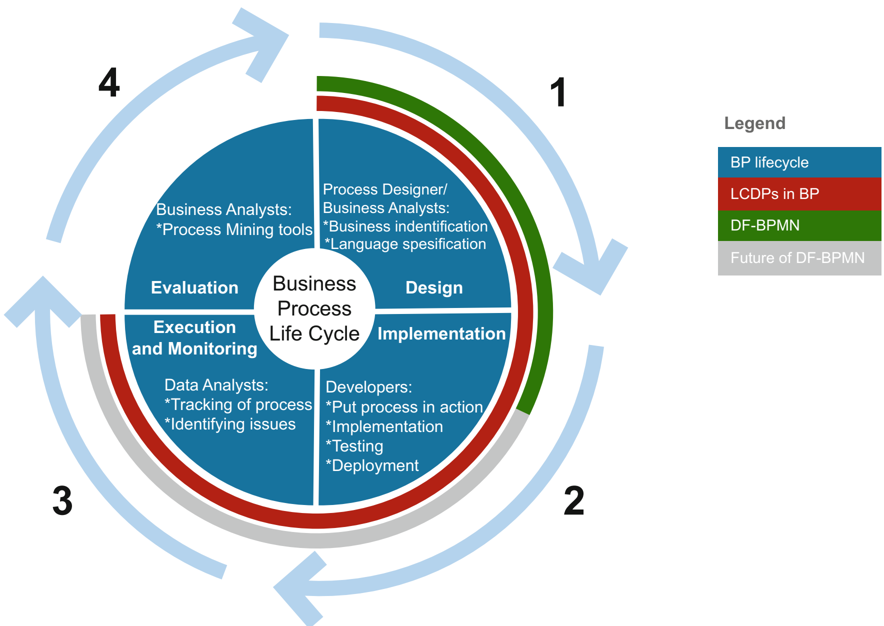


Fig. 1. Comparison of the Business Process Lifecycle vs. Low-Code Process Development (LCPDs) in Business Process (BP) vs. the DF-BPMN approach

Business Process Modeling and Notation (BPMN) is a widely-used standard for modeling business processes, designed to be comprehensible by various stakeholders such as analysts, developers, and business people [8]. However, BPMN process models have limited detail on persistent data structures and struggle to represent interactions between data objects and data stores. As a result, the data flow perspective has been neglected in BPMN automation, in comparison to the control flow. This can lead to misunderstandings and errors during implementation by technical developers. For example, in the BPMN model shown in Fig. 2, the activity "Complete quotation" has an input and output "DB" data store representing a database, without detailing the object name and their attributes. Therefore, the representation of "DB" can be ambiguous and prone to mistakes during process implementation.

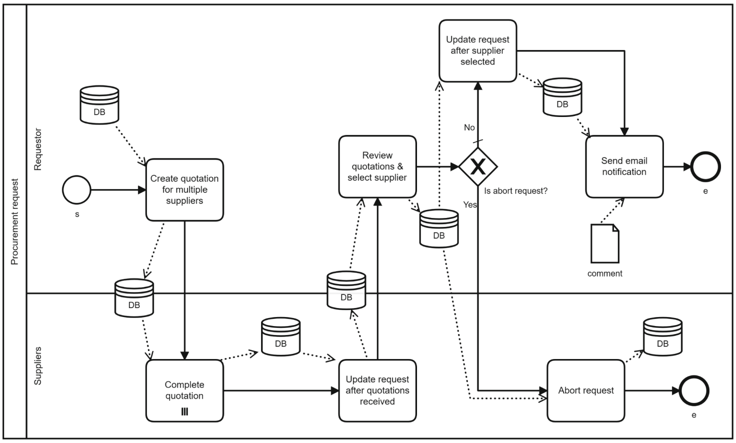


Fig. 2. A procurement request using BPMN 2.0.

This paper focuses on the first step of low-coding, which is language specification. We propose an extension to BPMN called DataFlow BPMN (DF-BPMN) that integrates a data perspective with the control flow perspective. DF-BPMN bridges the gap between process and data diagrams, enabling a clear understanding of the data involved in business processes (see Fig. 1). With DF-BPMN, developers can represent data in a graphical format within BPMN, zooming into the activity for modeling manipulated data and offering different levels of granularity for a better understanding of the interactions between control flow and data perspectives. To evaluate our language, we conducted a quantitative analysis based on real-life business process models and analyzed the results statistically using the Paired t-test [5].

The remainder of this paper is organized as follows. In Sect. 2, we discuss the limitations of BPMN in terms of the data-flow perspective. In Sect. 3, we review existing works related to our contribution. In Sect. 4, we present the zoom-in approach of DF-BPMN. In Sect. 5, we discuss the novel insights of merging data and processes. Then, in Sect. 6, we present the evaluations of DF-BPMN before concluding in Sect. 7.

2 Motivating Example and Limitations of BPMN as a Data-Flow Language

Figure 2 represents a business process model modeled in BPMN 2.0. A user fills a procurement request and identify potential suppliers. This request is sent to those suppliers for quotation. After completed, the quotations are sent back to the requestor for review and selection. To effectively implement the process in Fig. 2, developer must understand how the information is conceptually structured and arranged within each class (suppliers, quotation and request) and how the process interacts with it. *Request* has connected to *Quotation* and *Supplier*. Each *Request* has one or more *Quotation*. Each *Quotation* has one *Supplier* and also the *Supplier* is selected for a *Request*. All these necessary information are concealed in a simple *data store* "DB" in Fig. 2.

We identified conceptual limitations in BPMN **(L1-L3)** that hinder model understandability and quality, requiring a well-structured approach to address these issues and provide clearer guidance for developers.

L1. BPMN Data Stores are Underspecified. Data stores in BPMN process models represent persistent data sources [8], but they lack details on the conceptual structure of a database. For example, in the activity "Create quotation for multiple suppliers" it is unclear whether the input classes are *Supplier*, *Request*, or both, and whether the output class is *Quotation*. This ambiguity complicates the use of BPMN for process models connected to database systems, hinders tracking instance manipulation, and negatively impacts understandability between different developers. For instance, some developers can understand from the data store connection in "Create quotation for multiple suppliers" activity, that the request is already created and they need to select it from the database, while others may understand that they need to create a new instance object of *Request*, which can lead to time loss during implementation.

L2. Interaction Between Data Instances is Not Clear. Data objects represent volatile process data and are connected to activities through associations [8]. For example, in the "create quotation for multiple suppliers" example, the interaction between data instances, such as the "DB" data store used as input and output, is unclear. This ambiguity, coupled with the lack of explicit representation of data cardinality [2], like multi-instance data, complicates the model's interpretation. These limitations can result in development inefficiencies, as developers struggle to determine involved data objects, required information, and data relationships. Understandability of the BPMN model can also vary among developers based on their experiences and perspectives, highlighting the need for a single visual language that conveys all necessary information, including data cardinality.

L3. Data From/To External Environment are Not Supported. Information systems are often connected to external environments, such as user interfaces and services, making it crucial to represent the interaction between process activities and external resources graphically for effective process modeling and development. However, BPMN models lack support for this representation, making it challenging to determine which activities are connected to external data or services. For instance, the activity "Complete quotation" requires the user (a supplier) to provide necessary quotation information through an external user interface, which is not supported in BPMN. Therefore, while implementation, the developers required textual infractions to understand the process model. The graphical representation of external resources is important for effective process implementation, as it helps prevent misunderstandings and errors during implementation.

Since process models and data schema are conceptualized independently, it is necessary to support designers in understanding and capturing the relationship between processes and different data types in order to develop data aware process modeling.

3 Related Work

Low-Code Development Platforms (LCDPs) currently focus on the execution of the process model, with little attention given to extending process modeling languages to make them easier for developers to use [10]. In this section, we not only present LCDPs, but also illustrate some approaches that extend BPMN to link data and processes [4,6,7].

LCDPs are defined as "platforms that enable rapid delivery of business applications with a minimum of hand-coding." Most of these platforms use similar concepts in graphical user interfaces, allowing developers to define and manipulate data specified through tables, forms, reports, and other types of representation [10]. As the author said, "we are comparing BPMN modeling elements to build the case study application with its equivalent modeling constructs using different LCDPs associated with the data handling mechanism and their implementation." which mean that most of platforms used BPMN elements. For example, Bonita[1] is an open-source and extensible platform for business process automation and optimization. They use BPMN diagrams to describe the business processes. The structure of the business data is presented by Business Data Model. Also, they allow developers to define forms, reports, and other types of representation. Another tool called Mendix[2] allows the user to build processes using the available Microflow modeling language, which is based on the BPMN standard and helps to extend or change the default behavior of the developed application. This language defines the same elements as BPMN but with different symbols. For example, an event in Mendix is equivalent to an event in BPMN (details in [10]). However, these representations cannot help developers in their implementation by low-coding. LCDPs in business processes focus on the execution of the process and facilitate the execution of the process to the organization. But developers still spend time on the implementation, and they cannot track their business data using the existing workflow languages.

1 https://bonitasoft.com.
2 https://www.mendix.com.

Various approaches have been developed to extend BPMN and add support for data modeling. These approaches introduce new concepts, elements, and mechanisms for modeling data entities, relationships, and constraints, thus enabling a more comprehensive and integrated representation of both process and data aspects. In [7], the information model of a process is connected to its BPMN process diagram through OCL expressions. The information model is represented by a class diagram that includes a "Artifact" class containing process variables. The process diagram is formalized as a Petri net, and BPMN activities are defined using OCL operation contracts. These contracts are converted into logic derivation rules that can be easily translated into SQL queries. Another framework defines new variables, pre-conditions, and effects on activities by adding new properties and accessing data objects and data stores to modify them [6]. This framework uses a verification model to parametrically verify data-aware processes with respect to read-only relations. However, these approaches neglect the important aspect of process visualization, which is necessary for developers to easily understand and use the model. In [4], "Activity Views" proposes a new extension to bridge the gap between process and data modeling, with a focus on databases. Moreover, [4] only considers the databases aspect of the model. However, The data in Business process does not restricted to database, and there are different data types used in BP, like data object in BPMN.

However, our approach introduces a new visual concept that allows developers to easily model data in BPMN. Furthermore, it enables developers to zoom in on activities to manipulate and interpret data. Additionally, the ability to zoom in and out of activities allows for exploration of different levels of granularity, a feature that is not supported in existing works. Indeed, our approach takes into account most of the data required in business processes.

4 DF-BPMN: DataFlow in Business Process Modeling and Notation

In this section, we will present the graphical elements of our language, followed by an example that addresses all the limitations of BPMN using DF-BPMN, and then proceed with the formal definition of DF-BPMN.

4.1 Graphical Process Modeling Using DF-BPMN

In this section, we present a first step to low-coding approach that aims to establish a conceptual link between BPMN process models and data models, in order to capture the connection between them more effectively, which helps developers better understand the model. Our proposed concept, ***Activity DataFlow***, allows for a more detailed representation of how process activities interact with and manipulate different types of data, such as database, process variable, and external environment data. By specifying the inputs, outputs, and data operations performed by each activity, the method provides a comprehensive and detailed representation of the process and its relationship with data. This allows zooming into the activity for early identification of potential data-related issues and better optimization of the data flow within the process. This helps reduce implementation time and minimize mistakes during execution. Additionally, zooming out of the activity presents the control flow perspective of the process, while developers can zoom in on activities as needed.

Our approach to modeling data in BPMN starts with process activities, which are a common starting point for data modeling. The main objective of the *Activity DataFlow* extension is to visualize how data flows into a specific activity based on various sources, and to capture important details about the data operations that are performed. This allows us to better understand the flow of data throughout an activity.

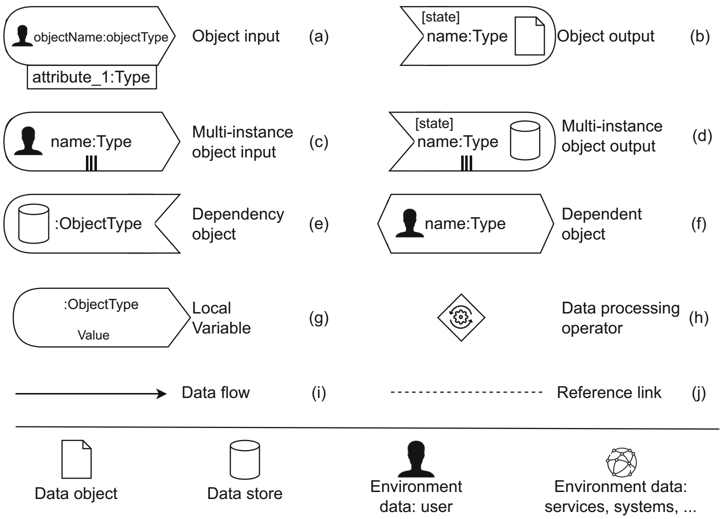


Fig. 3. Symbols used in DF-BPMN.

Figure 3 represents the symbols proposed to allow to build your BPMN models injected by data[3]. The symbols include inputs and outputs that represent the data objects manipulated within the activity. Each object can have a type, and there are various types of objects such as process variables (also known as data objects in BPMN), databases (also known as data stores in BPMN), two types of environment data: (1) related to user operations, such as user forms and websites; and (2) external resources, such as services and systems; as well as local data, which is a static value used within the activity. The icons in the second part of Fig. 3 represent several types of data objects (input or output). Except for local data, which was displayed without an icon.

[3] You can find the detailed documentation here.

The input shape in this language is a semi-circle with an arrow (see Fig. 3 (a)). Each input can have one or more attributes, which are presented as rectangles. The input shape can represent one of several types, as indicated by corresponding icons. Each input also includes the name and type of the object it represents. Similarly, each attribute has a name and type associated with it. The output shape is the complement of the input shape, starting with a left arrow and semi-round shape. It has several types, like the input, and includes a name and type, as well as the option to attach attributes. Additionally, the state of the output object is indicated by "[]" at the top of its name, indicating any operation performed on it during the process (see Fig. 3 (b)).

To facilitate interaction between the input and the output, a dataflow sequence is introduced to represent the transfer of input from the source to the destination, it's a sequence flow presented by an arrow. The dataflow sequence visually connects the input and output, indicating the direction of data transfer. In this way, the flow of information between the input and the output can be easily understood by the process designer (see Fig. 3 (i)). We also use a reference link (as shown in Fig. 3 (j)) to represent equivalent data objects. The reference link is displayed using a dotted line. In addition, the representation of data also needs to be considered. Sometimes, the input may read multiple instances of the same data object, while the output may write or update multiple objects. To address this, the multiple-instance object is represented by adding three bars "|||" to the input or output shape (as shown in Fig. 3 (c-d)).

Moreover, our DF-BPMN language includes three other types of input: dependency, dependent, and local data objects. The dependent and dependency objects are always interdependent, representing situations where the user needs to select certain data from the input in order to modify the output. The dependency object is always represented as a user (f) who needs to read and select data from the dependency object (data object/store) (e), therefore, there are dependent together. The local data object is a variable that can be used as a static value within the activity (g). These different types of inputs expand the range of use cases for our language, providing more flexibility and precision in modeling complex processes.

Finally, in the context of our DF-BPMN language, complex operations such as arithmetic operations, logical operations, and conditions are represented using data processing operators. These operators are denoted by the symbol (h) and can be used in combination with input and output objects to create complex processing logic. For example, if an activity requires the addition of two input objects, it can be represented using a data processing operator denoting addition, with the two input objects as inputs and the output object as the result. These data processing operators add another layer of flexibility to the DF-BPMN language, allowing for the representation of more complex business processes.

4.2 DF-BPMN in Action

Figure 4 depicts several instances of Activities DataFlow for the example shown in Fig. 2. These visualizations provide a way to zoom in on specific activities

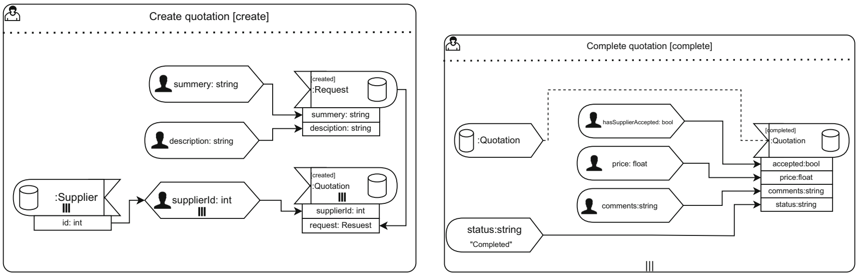


Fig. 4. Activities Modeled with DF-BPMN Language: (a) Create Quotation, (b) Complete Quotation.

and understand how data is manipulated within them without ambiguity[4]. For example, the activity "create quotation" in Fig. 2 has one input data store "DB" and one output data store "DB", making it impossible to determine which data object is in the input and which are in the output. However, the Activity DataFlow "create quotation" (shown in Fig. 4(a)) extends the "create quotation" activity to include all data related to it. It has four inputs: one input from the data store called "supplier" object, which has one attribute (i.e., "id"), and three inputs from a user (environment data) (**L3**). It also has two output objects, one of which is a multiple-instance object (cf. **L1**).

Furthermore, in the same activity, the representation of inputs and outputs addresses **L2** from Sect. 2. The *supplierId* object is a list of integers selected by the user based on the list of objects *supplier* already stored in the data store. Thus, we have two input data objects that are dependent on each other, called the dependency and dependent objects, respectively: *supplier* from the database and *supplierId* list selected by the user. Then, the *summary* and *description* are used to create a *Request* object, which currently has a *[created]* state. Moreover, this created object (*Request*) is used as input for the *Quotation* objects. The *Quotation* is a multiple-instance object based on the list of *supplier* selected by the user, creating a *quotation* object for each *supplierId.* This situation provides a solution for **L2**. This indicates that the goal of the "create quotation" activity is to create two objects with states *[created]*, with respect to the milestone of the activity, which is *[create]*, visualized around the name of the activity. This case resolves the ambiguity of the activity milestone presented in **L4** in Sect. 2.

In addition, the attributes of each object refer to the conceptual model of UML. This helps to resolve **L1** by providing information on the required attributes for each UML class that is being processed in the activity. The association between inputs and outputs represents the interaction between data within the activity (**L2**). The Activity DataFlow "completed quotation" (in Fig. 4(b)) has different input types. It includes a local data input *status*, which takes a value of "completed" to modify the value in the *Quotation* object.

[4] You can find all the process models here.

4.3 The Missing Link: Uniting Process and Data for Clarity

Starting from the BPMN definition in [8], we extend an activity to include a data flow into it. Therefore, Definition 3 indicates the activity's extension into *Activity DataFlow*. Each activity that has data can be enhanced to a *Activity DataFlow*. The latter consists of: (i) a set of inputs defined in Definition 1, which are data instances from various resources, such as a local data, a data object (process variable), a data store (database), or a data environment (other resources: user or other); (ii) a set of outputs defined in Definition 2, which are data instances are written/modified by the activity; (iii) a set of operators, which are the data processing operation from one or multiple inputs to one or multiple outputs, to represent that are operations do in this activity, e.g., condition, arithmetic's operations,... ;(iv) and (v) a collection of dataflow and a set of references that reflect the interaction of data instances into the activity; a dataflow sequence represents the correlation of a data instance; and a reference represents the equivalent of data instances[5]. For example, "Create quotation" of Fig. 2 is connected to data, therefore it can be extended to *Activity DataFlow*. In which, this Activity DataFlow has: (i) Four inputs: one database input (supplier), and three user inputs; (ii) Two databases outputs (request,quotation). The interactions between these data instances are represented through a collection of dataflow and a set of references in the Activity DataFlow.

Definition 1 (Activity Input). *Let* $Input = GI \cup LI$ *as an input of an activity, where:*

- $GI = (objectName, objectType, type, Att_{set}, isMuliple)$ *as a Global Input where:*
  - *objectName is a name to describe the input objectType can be any object type you need (e.g. integer, string or complexType)*
  - $type \in \{data\_object,\ data\_store,\ user,\ systems\}$
  - $Att_{set} = \{att_1, ..., att_k\}$ *where* $att_k = (attributeName, attributeType)$ *such as attributeName is the description of attribute attributeType is any type (e.g. integer, string, complexType, ...)*
  - *isMuliple is a boolean variable to describe if the object is represent multiple-instance object*
- $LI = (objectName, objectType, objectValue, isMuliple)$ *as Local Input such as* $objectValue$ *represents the static value of this input.*

Definition 2 (Activity Output). *Let Output = (objectName, objectType, type, state,* Att_{set}*, isMuliple) as an output of an activity, where objectName, objectType, type,* Att_{set}*, isMuliple are represented in the Definition 1, and state define the state of each object data during the execution time (like created, updated, deleted, ...)*

Definition 3 (Activity DataFlow). *Given an activity ac in a process model. Let Activity DataFlow* $ad_{ac} = (state, I_{set}, O_{set}, Operator_{set}, DF_{set}, R_{set})$ *is a tuple to represents the data flow into the activity, where:*

[5] You can find the implementation details here.

- $I_{set} = \{i_1, \ldots, i_j \mid i_j \in Input\}$, *is a set of inputs accessed by process activity ac.*
- $O_{set} = \{o_1, \ldots, o_l \mid o_l \in Output\}$, *is a set of output produced by process activity ac.*
- $Operator_{set} = \{operator_1, \ldots, operator_m\}$ *is a set of operators in which represent the existing of data processing operations where* $operator_m$ = *operaterName and operaterName is a name to describe the operator.*
- $DF_{set} = \{df_1, \ldots, df_l\}$, *is a set of dataflow sequence to connect different data objects, where* df_l = *(sourceObject, targetObject) such as* {*sourceObject, targetObject*} $\subset \{I_{set} \cup O_{set} \cup Att_{set} \cup Operator_{set}\}$ *where* $Att_{set} \subset \{i_j.Att_{set} \cup o_l.Att_{set} \mid i_j \in I_{set}$ *and* $o_l \in O_{set}\}$ *and sourceObject* $\neq$ *targetObject and vise versa.*
- $R_{set} = \{r_1, \ldots, r_n\}$, *is a set of references to represents the equivalent between objects, where* $r_n = (o_1, o_2)$ *such as* $\{o_1, o_2\} \subset \{I_{set} \cup O_{set} \cup Att_{set}\}$ *where* $Att_{set} \subset \{i_j.Att_{set} \mid i_j \in I_{set}\}$ *and* $o_1 \neq o_2$ *and vise versa.*

Activity DataFlow represents an integration of two established standards, namely BPMN and UML. It combines the concept of activity from BPMN with the class, attribute, and relationship concepts from UML. This feature of the Activity DataFlow method addresses several outstanding limitations (L1-L4)(Sect. 2) related to the representation of data in BPMN processes. Moreover, the definition of inputs(cf. Definition 1) and outputs(cf. Definition 2) resolve the underspecification of data store. Indeed, there are different types of it, included the external environment, which means, the representation of data from/to external environment are supported (cf. L3). Additionally, in Def 3, the dataflow sequence, the reference, and the data processing operator resolve the ambiguity of the interaction between different data instances type (cf. L2). Finally, the state of the activity and the state of each data output instance represent the milestone for each one (cf. L4, resolved).

5 When Processes and Data Meet: Integrating Analysis and Deployment

This section discusses some features of the research design area that resulted DF-BPMN and demonstrates the novel perspectives that may be discovered by using the DF-BPMN during process design, analysis, and deployment.

Understanding the Process Model Using Different Granularity. The concept of granularity is essential in human cognition, as it relates to the production, interpretation, and representation of granules [11]. A granule is a group of objects or points that are connected by either their familiarity, proximity, or utility. This process of granulation results in the formation of granules. When it comes to process modeling, granularity refers to the level of detail at which a process is represented. In the case of DF-BPMN, there are two granularities available: (i) The first granularity represents the entire control-flow of a model using BPMN2.0. For example, a standard BPMN diagram can be used to represent a quotation request. (ii) The second granularity is facilitated by the DF-BPMN model, where each activity can be either expanded or collapsed, allowing the reader to focus on the specific data being manipulated in those activities. Figure 5 is an example of a DF-BPMN that expands one activity and collapses all others in the model to represent the same quotation request example. These different granularities and representations provide valuable insight into the process, making it easier to analyze and communicate with stakeholders.

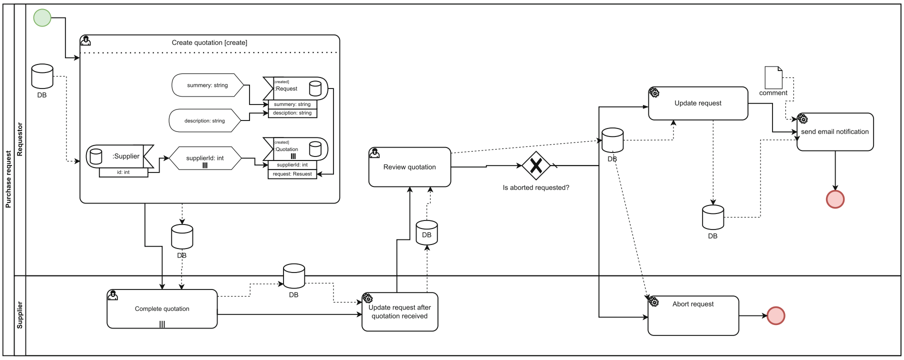


Fig. 5. DF-BPMN example which expand "create quotation" activity and collapse the others activities.

Tracking Data Objects in Business Process. Our extension increases the expressiveness of a BPMN process model with information about process-data-correlation on instance level. As such, it does not interfere with standard BPMN semantics. We defined a state for each data object, thus allow the developer to easily track the operation of each object by extracting its lifecycle based on the state changes. For instance, Fig. 6 represents the lifecycle of the object "Request", including the activity in which it was started and finished. In addition, the object-centric [1] has emerged recently to represent each object instance. This lifecycle can helps in the extraction of Object-Centric Event Data. OCED captures events and activities on specific objects in a system and provides a more detailed view of how objects are processed, making it useful for analyzing complex systems and processes and improving system performance using process mining tools.

6 Evaluation

The aim of this section is to evaluate the understandability of DF-BPMN by process designers and developers, comparing it to BPMN and Activity View in terms of presenting different data types. We will first outline the evaluation steps, and then present the results obtained from the study.

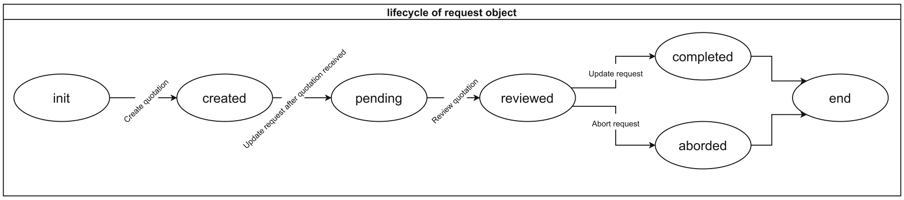


Fig. 6. Request object lifecycle.

6.1 Experiment Description

We conducted an evaluation of our DF-BPMN language in two phases. In Phase 1, we provided a tutorial to introduce DF-BPMN and its usage, with no prerequisite knowledge of UML. Additionally, we presented another work [4] in the tutorial for experimentation and comparison purposes. To ensure comprehension of DF-BPMN and the other work, we conducted a brief quiz. In Phase 2, we aimed to evaluate the understandability of DF-BPMN by presenting three exercises from different domains, with each exercise having different models (BPMN, DF-BPMN, and Activity View) for evaluation. We developed a web application tool[6] available on GitHub at https://github.com/NourEldin-Ali/open-bpmn, using the open-source project Eclipse GLSP[7] to use our language.

In order to evaluate the understandability of DF-BPMN, a human-oriented experiment was conducted, similar to a previous study [4], with a single controlled variable. The goal was to measure the understandability of DF-BPMN by process designers and developers, comparing it with BPMN and other works, particularly with respect to the relationship between a process and data. Two hypotheses were formulated to analyze these improvements quantitatively and qualitatively. The first hypothesis (H1) tested the perceived ease of understanding, suggesting that DF-BPMN improves the visualization of data objects, leading to a better understanding of the data required for activities and how it is utilized in a process, without any textual information or UML. The second hypothesis (H2) tested the perceived ease of understanding, suggesting that DF-BPMN is a better solution for modeling data in BPMN.

The evaluation was conducted by five PhD students who had basic knowledge of BPMN and three professional developers working on a Business Process Management System editor. In Phase 1, all the participants attended a 30-minute tutorial, consisting of a video and a short quiz, on how to use DF-BPMN and Activity View.

In Phase 2 of the evaluation, participants were divided into three groups to validate Hypotheses (H1) and (H2) through three exercises, each featuring a questionnaire with 12 questions. The exercises provided participants with a textual process description, data operations, BPMN, and UML diagrams. Each

[6] Online Demo: https://github.com/NourEldin-Ali/open-bpmn#start-the-online-demo,.

[7] https://www.eclipse.org/glsp/.

group used a different model: BPMN, DF-BPMN, or Activity View, with groups rotating between exercises to avoid learning effect bias. The DF-BPMN group had two iterations, one with text and UML and the other without. The application domains included shipping orders from a website, triage in an emergency room, and loading a book. The goal of this phase was to validate Hypothesis (H1) by comparing DF-BPMN results with and without text and UML, and to validate Hypothesis (H2) using a paired t-test [5], a statistical method to determine significant differences between the means of two related groups while accounting for individual variability.

6.2 Results

The outcome of the study indicates that DF-BPMN has a significant impact on streamlining the design and comprehension of processes and their associated data. This leads to a reduction in the time spent on tasks and improved task accuracy.

During Phase 2, we conducted a quantitative assessment of the effectiveness of DF-BPMN by comparing it to BPMN and Activity View. We measured the time taken to complete each task for each participant and counted the number of accurately answered questions, adhering to strict criteria for both accuracy and completeness of responses. Finally, we employed a paired t-test to analyze the results, comparing DF-BPMN to BPMN and DF-BPMN to Activity View, where the execution times of one exercise using DF-BPMN were contrasted with those of the same exercise using another model (BPMN/Activity View).

The results displayed in Fig. 7 show that the exercises with DF-BPMN took an average of 12 min, and 71% of the answers were evaluated as correct. In contrast, the results with Activity View took an average of 17 min, with only 40% of the answers being correct. With BPMN, the results took 22 min, with only 38% of the answers deemed correct. These results support hypothesis **H2**. In fact, by applying the paired t-test to the measured correct answers between DF-BPMN and BPMN, the p-value $= 8 \times 10^{-5} < 0.05$. Furthermore, between DF-BPMN and Activity View, the p-value $= 0.003 < 0.05$, indicating that the obtained results illustrated in Fig. 7 are statistically significant, and hypothesis **H2** is satisfied. Indeed, the comparison of DF-BPMN results with and without textual descriptions reveals that without text, 74% of answers are correct, while with textual description, 71% are correct. This suggests that the textual description in our approach is not necessary to understand the model, and hypothesis **H1** is satisfied.

Indeed, all the participants stated that completing the first experimental task without the aid of DF-BPMN was more challenging, and 90% of them responded positively when asked if DF-BPMN improved the modeling of the relationship between processes and data. Next, we asked the participants to rate the usability of DF-BPMN on a scale of 1 to 5, where 1 indicated "strongly disagree" and 5 indicated "strongly agree". The average results of this questionnaire-based interview are presented in Fig. 8.

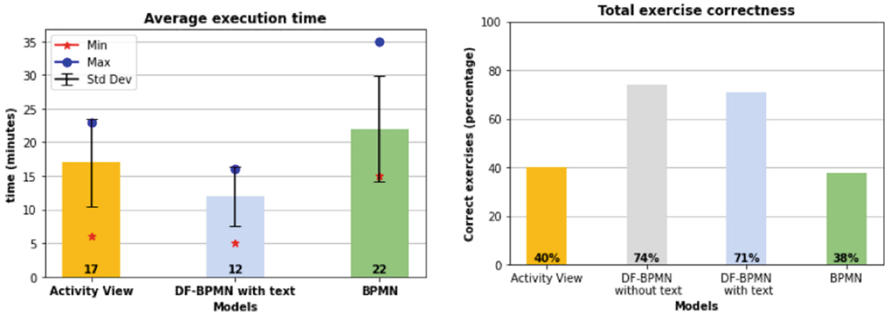


Fig. 7. Average execution time with standard deviation (left) and total percentage of the correct answers (right) for the whole the PHASE 2.

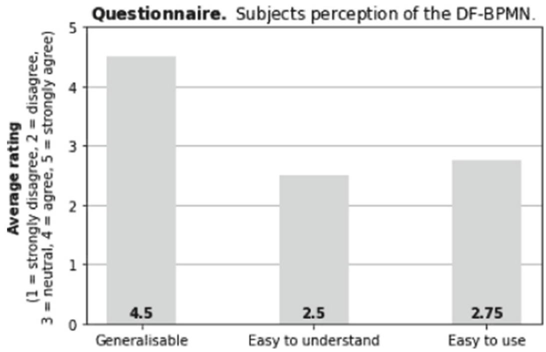


Fig. 8. Average rating of subjects perception of the DF-BPMN.

7 Discussion and Conclusion

Low-code development platforms (LCDPs) aim to simplify software systems' development by providing easy-to-use graphical interfaces. The system behaviors are defined through available data handling and workflow mechanisms enabling the specification of business processes from users that do not have strong programming skills. Moreover, a clear understanding of the data involved in business processes is critical to reduce the mistake in the implementation of the model.

Although LCDPs in business process are most widely used BPMN as business process model, but it has limitations as a data-flow language. Specifically, BPMN underspecifies the data store and does not support the relationship between different data types. Additionally, it does not represent users and external systems, which can lead to misunderstandings in the process model.

To address these issues, we proposed DF-BPMN, a first step in low-code solutions that connects process and data diagrams by using the Activity DataFlow, an extension of the standard BPMN activity. DF-BPMN provides insights into how data flows through a process and identifies areas for data-related improvements, enabling process designers to model the complex relationships between processes and data. DF-BPMN allows developers to represent data in a graphical format, improving collaboration between business and developers.

Based on our evaluation in Sect. 6, DF-BPMN is a promising approach for supporting process designers in modeling the complex relationships between processes and data. DF-BPMN is simple to understand without requiring additional information, and because it is a visual language, we are confident that the information will be understood by everyone. However, humans still require assistance in creating a model using DF-BPMN, which can be resolved using AI assistance. Indeed, we are working on the second step of low-coding by generating an execution code to be used in the engine for the execution of the process.

In conclusion, by integrating low-code and business process modeling, DF-BPMN provides the first step of low-code solution that bridges the gap between process and data diagrams and enables a clear understanding of the data involved in business processes. It has the potential to improve overall efficiency and effectiveness by identifying areas for data-related improvements.

References

1. van der Aalst, W.M.P.: Object-centric process mining: dealing with divergence and convergence in event data. In: Software Engineering and Formal Methods - 17th International Conference, SEFM (2019)
2. van der Aalst, W.M.P., van Hee, K.M.: Workflow management: models, methods, and systems. Cooperative information systems (2002)
3. Aguilar-Savén, R.S.: Business process modelling: review and framework. Int. J. Prod. Econ. **90**, 129–149 (2004)
4. Combi, C., Oliboni, B., Weske, M., Zerbato, F.: Conceptual modeling of processes and data: Connecting different perspectives. In: Conceptual Modeling - 37th International Conference, ER (2018)
5. Field, A.: Discovering statistics using IBM SPSS statistics. SAGE (2013)
6. Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: Delta-BPMN: a concrete language and verifier for data-aware BPMN. In: Business Process Management - 19th International Conference, BPM 2021 (2021)
7. De Giacomo, G., Oriol, X., Estañol, M., Teniente, E.: Linking data and BPMN processes to achieve executable models. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 612–628. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_38
8. OMG: Business Process Model and Notation (BPMN), Version 2.0 (2011). http://www.omg.org/spec/BPMN/2.0
9. OMG: OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1 (2011). http://www.omg.org/spec/UML/2.4.1
10. Sahay, A., Di Ruscio, D., Iovino, L., Pierantonio, A.: Analyzing business process management capabilities of low-code development platforms. Practice and Experience, Software (2022)
11. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy sets and systems (1997)

Management

Towards a Theory on Process Automation Effects

Hoang Vu[1,3], Jennifer Haase[4,5], Henrik Leopold[2,3(✉)], and Jan Mendling[4,5,6]

[1] SAP, Walldorf, Germany
h.vu@sap.com
[2] Kühne Logistics University, Hamburg, Germany
henrik.leopold@the-klu.org
[3] Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
[4] Humboldt-Universität zu Berlin, Berlin, Germany
{jennifer.haase,jan.mendling}@hu-berlin.de
[5] Weizenbaum Institute, Berlin, Germany
[6] WU Vienna, Vienna, Austria

Abstract. Process automation is a crucial strategy for improving business processes, but little attention has been paid to the effects that automation has once it is operational. This paper addresses this research problem by reviewing the literature on human-automation interaction. Although many of the studies in this field have been conducted in different domains, they provide a foundation for developing propositions about process automation effects. Our analysis focuses on how humans perceive automation technology when working within a process, allowing us to propose an effective engagement model between technology, process participants, process managers, and software developers. This paper offers insights and recommendations that can help organizations optimize their use of process automation. We further derive novel research questions for a discourse within the process automation community.

Keywords: Business process management · Process automation · Human Factors · Robotic process automation · Workflow systems · BPMS

1 Introduction

Business process management (BPM) is concerned with the continuous improvement of business processes [8]. Process improvements can be achieved by implementing processes using business process technologies such as workflow systems, robotic process automation, or blockchain technologies [46]. Often, such process automation results in drastic improvements in process performance indicators.

Jan Mendling—The research by Jennifer Haase and Jan Mendling was supported by the Einstein Foundation Berlin under grant EPP-2019-524 and by the German Federal Ministry of Education and Research under grant 16DII133.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 285–301, 2023.
https://doi.org/10.1007/978-3-031-41623-1_17

For instance, major companies report cost savings of 50% thanks to the implementation of robotic process automation [27] and 3,000 saved person-hours per month [24].

So far, most BPM research has been concerned with getting process automation done. While the challenges of doing process automation deserve attention, it is equally important to investigate which short and long-term effects process automation can entail. For instance, robotic process automation seems to be a technology with short-term benefits and long-term problems. Maintenance appears to be difficult due to complicated governance and loss of knowledge [12,32]. Little of these issues are appropriately reflected in BPM research. A theory explaining process automation's different effects is missing so far.

In this paper, we address this research problem from a theoretical angle. Our approach begins with a review of automation effects observed in prior research on human-automation interaction. However, since most of this research focuses on human control of complex technical systems, its findings cannot be directly applied to process automation. Therefore, we examine the extent to which effects described in research on human-automation interaction are relevant to process automation and derive research areas for better understanding in the future.

This paper is structured as follows: Sect. 2 provides background information on our research, including an overview of process automation and a discussion of the general perspectives that have developed in research on human-automation interaction. Section 3 outlines our methodological considerations for reviewing the literature. Section 4 presents our findings based on the review of the literature. Section 5 discusses how our findings can inform research on business process automation. Finally, Sect. 6 concludes the paper with an outlook on future research.

2 Background

In this section, we discuss what process automation is and which general perspectives on automation have been developed in the human factors literature. This discussion serves as a foundation for our subsequent literature review and identification of process automation's effects.

2.1 Business Processes and Process Automation

First, we describe how business processes and process automation are related. To this end, we refer to Fig. 1, a simplification of [8, Ch.1]. A business process receives inputs and is targeted toward achieving some desired results. A business process is typically decomposed into tasks for which different process participants are responsible. The performance of the process is monitored by a process manager, who is also responsible for initiating redesigns of the process if desired results are not achieved. The process manager can commission projects towards the automation of the process, which software developers implement.

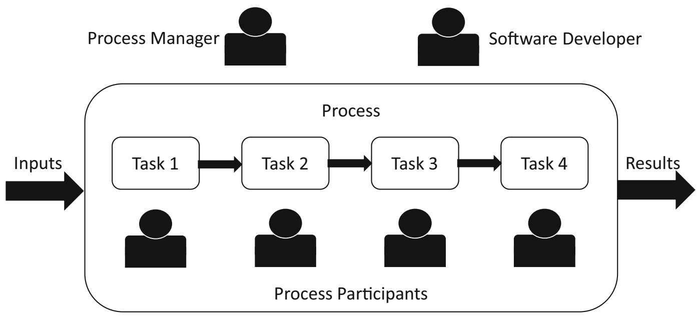


Fig. 1. Process, Tasks, Process Participants, Software Developer, and Process Manager

Automation refers to the machine execution of tasks that humans do not wish to perform or cannot perform as accurately or reliably as machines [36]. In line with this statement, we define *process automation* as assigning at least one task or at least one control flow link between two tasks to a machine. In this paper, we focus on information systems as a specific class of machine. Control flow is often automated using enterprise systems, workflow systems, or robotic process automation systems [8, Ch.9]. Information systems that perform tasks previously performed by humans can range from simple tools, such as calculators, to complex systems, such as artificial intelligence and machine learning.

2.2 General Perspectives on Automation

Research on human-automation interaction has developed several useful concepts for investigating automation. A key observation of this research discipline is that automation comes in various forms and sizes and thus requires a conceptual differentiation to understand various effects on human performances. Parasuraman's process automation model distinguishes four types of tasks and corresponding technology to substitute human performance: inquiring and presenting information, processing and analyzing information, decision and action selection, and (physical) action implementation [34].

Information presentation automation includes generating dashboards or reports to present real-time insights into operational performance or sales figures to users.

Information processing automation involves calculation, data analysis, automatic reasoning, or natural language processing of large data volumes to identify patterns that are difficult to detect manually.

Decision automation can involve using expert systems or decision support tools to provide recommendations based on data analysis and decision rules to assist humans in making complex decisions.

Physical automation includes assembly line robots in manufacturing plants to improve production efficiency and quality or the use of automated vehicles in logistics to transport goods without human intervention.

More complex systems can provide several of these automation functions, just like autonomous cars require to assess information from the vehicle surroundings, analyze these to decide on how to drive properly, and then actually physically enact them.

In addition to the types of tasks, Parasuraman et al. proposed a model that includes ten levels of human interaction with automation [36]. The levels range from no automation (Level 0) to full automation (Level 9), with Level 10 being a hypothetical level that involves no human interaction. While higher levels of automation reduce the individually perceived workload, this comes at a risk of less situational awareness and a higher chance of missing system failures [33, 49]. Overall, each form of automation has the potential to increase efficiency, reduce errors, and enhance safety in various industries. However, it is important to consider the social, psychological, and economic implications of automation, particularly how it affects human labor and job security.

Research on human-automation interaction has discussed automation in various settings. We observe, though, that tasks reflected in this research discipline typically focus on sensor-motoric tasks, often under emergency conditions. Several of the categories established by this literature have face value in the light of process automation, for instance, skill decay. The question begs, however, to which extent research insights on a.o. nuclear power plant disasters and plane crashes can be transferred to an office work area largely supported by information systems and corresponding process automation. We review and structure the human automation interaction literature to address this question and discuss its applicability to process automation.

3 Research Method

This section discusses our methodology for conducting a representative literature review to collect the main effects of automation [3, 5]. This review aims to identify the effects of process automation on humans involved in business processes. To this end, we turn to engineering psychology, a research field investigating automation in general and diverse settings [48]. We deliberately focus on articles published in its flagship journal *Human Factors*, published since 1958, for the paper selection of our literature review. This focus is motivated by the fact that this journal published the most seminal works of engineering psychology. We identify *human-automation interaction* as the key term for our search. Its relevance emerges from process automation applying automation technology for repetitive tasks and coordination. Often, this yields a partial or semi-automated solution. As search string, we therefore use "*human-automation interaction*" as well as the combination of "*human interaction*" and "*automation*".

We were aware that the term *automation* in *Human Factors* is broader than *process automation*. The former includes, for example, areas such as autonomous driving or flight simulation. We believe that the interaction of humans and automation and the effects on humans in these areas need to be investigated if and to what extent they are also relevant to process automation.

The search was conducted in February 2023 and yielded 75 papers. To be included, a paper had to focus on phenomena that could be related to process automation and describe the interaction between humans and technology. We excluded papers describing specific technical mechanisms, solely physical interactions, or focused on highly specialized industries. For example, sensomotirical aspects such as pressing the right button were excluded because they are less relevant for workflows or robotic process automation. For similar reasons, we excluded papers focusing on highly specific domains, such as space science. After selection, 52 papers are included in the analysis.

The first author read all of these papers in their entirety and collected the phenomena and effects of human-automation interaction. To structure the effects, a data-driven qualitative inductive content analysis approach was followed [15]. In several iterations with the author team, all phenomena from the literature were mapped to the basic principles Prerequisites, Phenomena, Consequences, known from the grounded theory approach [6]. Since our main object of investigation is human automation, we further distinguish between human-driven aspects and technology-driven aspects within these three categories. In the following, we explain the overall model with detailed examples from the literature.

4 Findings

The literature on human-technology interaction describes phenomena categorized into interaction preconditions, main interaction phenomena, and resulting consequences (see Fig. 2). Successful interaction requires both humans and technology to meet certain basic requirements. The human must possess knowledge of and trust in the technology, while the technology must fulfill basic functionalities and design principles that cater to human specifics.

During the interaction, effects are observed on both the human and technical sides. These effects are influenced by preconditions and are inherent to the interaction and communication between the two parties. For instance, humans tend to blame technology for failure, particularly when performance expectations are unrealistic. The transparency and communication of the technology affect human reactions.

Long-term effects are visible on both the human and technical side, influenced by previous interactions and preconditions. Successful process automation can lead to humans relying on the system and losing process knowledge due to inactivity. However, it can be challenging for humans to react adequately in cases of technology failure.

In the following, we present the three levels of automation interaction in detail (see Fig. 2) from a human and technical perspective. Note that our list is not exhaustive as we focus on the phenomena discussed in the Human Factors literature. We identified the relationships in a unidirectional manner focusing on the influence of an aspect on another. An example is that the *understanding* of an automation influences the *expectation mismatch* of a human.

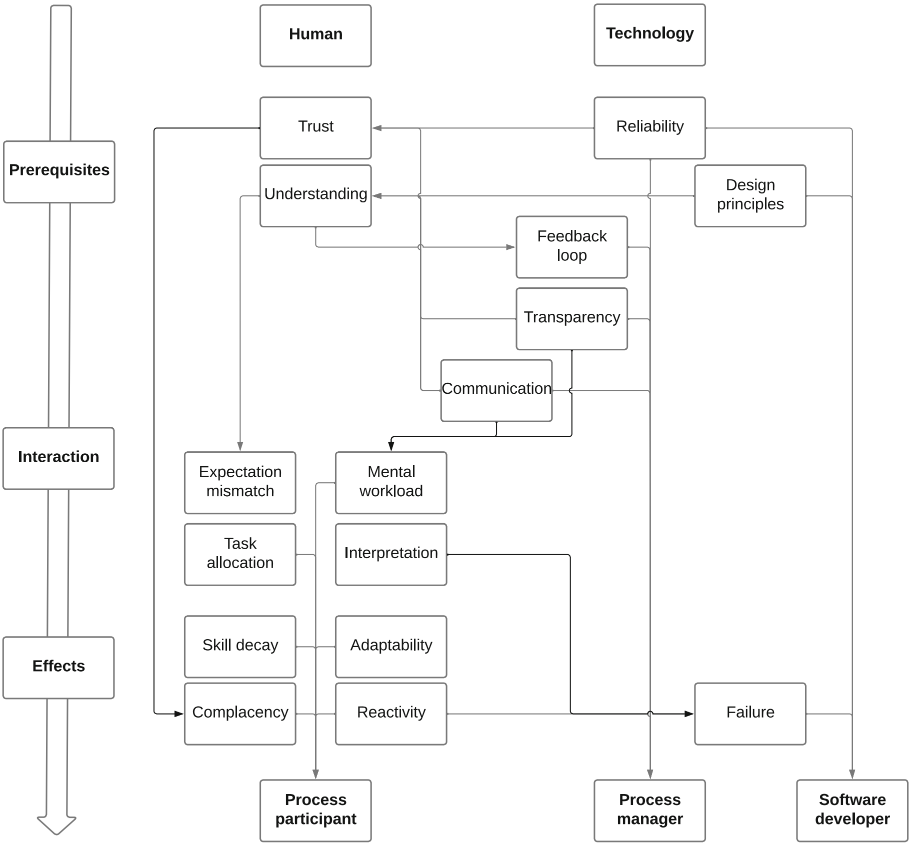


Fig. 2. Overview of all effects found in human-automation interaction literature

4.1 Prerequisites

The first level, prerequisites, pertains to the fundamental steps to facilitate a successful interaction between humans and automation technology. It emphasizes the human prerequisites for interaction, technological design principles, and essential properties. Our analysis has identified *understanding* and *trust* as crucial human requirements, while *transparency*, *reliability*, *design principles*, and the *feedback loop* are significant technological considerations.

Humans Need a Certain Understanding of Technology. To effectively work with automation technology, humans benefit from *understanding* the underlying technological mechanisms. It has been found that many humans are generally under-informed in that regard [41]. Due to the technological advancements in automation, automation technology has become more independent, and humans are no longer as 'in the loop' as before. These advancements can increase the complexity for humans, which can result in gaps and misconceptions. Consequently, automation technology often surprises humans as it behaves in a manner that humans neither anticipate nor understand [45]. Therefore, understanding automation technology is a key concern and refers to the in-depth knowledge a person has about how it works, what it can do, and its limitations. Understanding the technology is considered more important than its reliability and competence in real-world scenarios [1].

Humans Need a Certain Trust in Technology. Another aspect that needs to be accounted for is the *trust* humans have in automation technology. The level of trust determines to what extent humans are willing to accept automation technology and how well they interact with it in interactive settings [22]. A study on the usage of artificial intelligence devices has shown that the trust of humans heavily depends on the technology's transparency, reliability, and compatibility [19]. Humans tend to trust automation technology as they perceive it as an instrument with superior analytical capabilities and can outperform humans [35]. However, in scenarios where expert knowledge is required, the human tends to trust the human expert more. This highlights that human perception can vary based on the complexity of the automation scenario and that establishing trust is particularly important in these scenarios [26].

Technology Needs to be Transparent and Reliable. For an effective interaction between humans and automation technology, both *transparency* and *reliability* are important concerns. Transparency may relate to various features of automation technology, including its capabilities, responsibilities, activities, and goals. Depending on the specific automation scenario, transparency of different features might be important. For example, a study on automation in a nuclear power plant has illustrated that a high level of transparency regarding which automation components are currently running, how they work, and how they interact with other components has led to an increased supervision performance of the operators [41]. Besides transparency, also *reliability* plays an important role. In general, automation technology is considered reliable if it performs consistently and accurately from the user's point of view. Reliability has been found to significantly influence people's perceived trustworthiness of technologies [25].

Technology Needs to Follow Human-Centric Design Principles. Another aspect from the technology perspective is the automation design itself. It is generally difficult to design automation technology to make it clear and understandable to humans [41]. The implications of "bad" design can be severe. In some cases, the effort for using automation technology can outweigh its advantages (in terms of time savings, etc.), meaning that not using the automation technology is perceived as the better option [10]. In other cases, a non-suitable design might lead to human errors, which, at least in critical domains such as aviation, should be prevented at all costs [28].

Against this background, it is important to follow the human-centric *design principles*, which provide recommendations for human-automation interaction [20] as well as the user experience [17]. Among others, relevant aspects include the format of the user interface [2], the position of text and images [42], and the amount of data that is presented to humans [16]. Studies have shown that providing the right visual information to humans can increase their overall performance [29].

Technology has to Provide an Open Framework for Human Input. The *feedback loop* of automation technology, i.e., the possibility to include and consider human inputs, plays an important role in the overall performance of human-automation interaction. In this context, *adaptive automation* can incorporate the human response in the automation design [13]. By including human aspects, such as frustration, satisfaction, motivation, or confidence, to improve automation design, a higher performance, better results, and lower workload for the human can be achieved [37,52].

For complex socio-technical environments, the framework *ecological interface design* captures additional aspects, such as social and physical characteristics as well as the natural relationship in human-automation interaction [44]. This approach aims to enhance usability and safety and has been examined in the context of semantic mapping [43].

Another framework that should be considered when designing human-focused automation is *cognitive engineering* as it aligns the automation design within the cognitive abilities and constraints of a human [51]. This has been used in work domain analysis [31] and in a systemic model of computer response [47] to understand how humans process information, make their decisions and perform their tasks and use this as input to design an effective and human-oriented design.

4.2 Interaction

This section focuses on the various facets of human-automation interaction. It elucidates human behavior during the interaction and the anticipated response of the technology. Our analysis has identified expectation mismatch, interpretation, blame, acceptance, task allocation, and mental workload as the primary human aspects, while communication and justification are significant technological considerations.

Humans Tend to Perceive an Automation Technology Incompletely. The way humans perceive a given automation solution, e.g., in terms of its usefulness, is based on their knowledge of the underlying technology. This can result in an *expectation mismatch* [26], meaning that humans may have too high expectations concerning the automation technology's capabilities. This, in turn, may lead to a lower human-automation interaction performance, e.g., because the human is not expecting the need to deal with automation failure [45]. In these situations, humans tend to blame the automation technology and do not recognize their own responsibility as they perceive the automation technology as superior in terms of capabilities [7]. This effect is even stronger when the degree of automation is higher [14].

Humans Tend not to Interpret Information Thoroughly. Many automation solutions present data to humans. However, humans tend not to thoroughly examine the available data and do not always interpret it correctly [35]. This can lead to the human interacting with automation technology in an unintentional manner, resulting in unintended and incorrect actions by the system [38].

Humans Tend to Manage Their Task Inefficiently. When interacting with automation technology, humans need to focus on their *task allocation*, i.e., the tasks that have been assigned to them. A laboratory experiment has demonstrated that whenever automation greatly exceeds human capabilities, humans feel that they contribute too little to the overall task. As a result, they may intervene more often than required [7]. Especially in multi-tasking scenarios, this has been found to reduce human performance [35].

Another concern related to human-automation interaction is transitioning from automation to human, i.e., takeovers. Here, it has been found that the (perceived) absence of time pressure may lead to longer transition times. Also, multi-tasking leads to a slower response to resume control [11]. In this context, human *mental workload* plays an important role as it determines human cognitive abilities and resources to successfully perform a task. Studies have shown that overall interaction with automation technology can initially result in an extensive mental workload for a human compared to not involving automation at all [4]. This phenomenon, however, decreases over time through a learning process. A way to reduce the mental workload for humans is to transparently present relevant information for the employed automation solution [30].

Technology Should Communicate with Humans. Effective collaboration between humans and automation technology requires a certain level of *communication* from automation technology to humans. This allows humans to develop a high level of trust and understanding, which results in the human being less content to monitor and intervene with an automation unnecessarily [1]. One example is the proper communication of system state uncertainties, as these have a direct effect on the human mental workload, visual attention, and situational awareness [21]. If a human knows that critical situations will be communicated properly, they can focus on their other tasks without spending time monitoring detailed parameters. Another example relates to communicating social intent [25]. If humans know that automation technology considers human well-being explicitly (e.g., by preventing accidents in a production context), this increases the human's trustworthiness.

4.3 Effects

This section details the various effects of human-automation interaction. It describes how humans are influenced during and by the interaction, as well as the implications for the technology itself. Our analysis has identified reactivity, adaptability, and skill decay as the primary human aspects, while failure is a significant technological consideration.

Humans are Affected in Their Readiness to Intervene. As automation technology advances and becomes more robust, humans tend to become less aware of their situation and are less likely to take over manual control when needed [9]. This phenomenon is typically referred to as *skill decay*. Skill decay can lead to *vicious cycles* because humans lose their skills to take over in the course of time. If they, however, need to take over (because the automation technology fails), they might not have the ability to do it, leading to an even higher level of dependency on the automation technology [23]. This means that the ability of humans to intervene in critical processes must be both established and maintained [50].

Humans are Affected by Automation Complacency. An additional effect that occurs with an increasing level of automation is *automation complacency*. This phenomenon refers to a situation where humans become too comfortable and complacent with an automation technology [35]. The consequence of automation complacency is the general human expectation that automation technology will work, without knowing or understanding whether this will be the case. A related effect is *automation bias*, which arises when humans blindly rely on automation technology without actively monitoring and validating its activities. Both effects originate in the human over-trusting automation technology and may pose severe risks for the performance of human-automation interaction [35]. A way to manage complacency is complacency modeling, which can help predict the effects of different types of imperfect automation technology [50].

Humans are Affected by Automation Changes. Over time, the nature of human-automation interaction may change due to technological advancements. Therefore, an effective human-automation performance requires the human to *adapt* to these changes. In many cases, this is a question of additional training [23]. In some cases, however, humans have also been able to adapt without additional training. For example, a study with helicopter pilots has shown an increased human performance when the pilots were presented with additional information [18].

Technology May Fail. The interaction between humans and automation technology may cause *automation failure*. In such a case, the question is whether the human or the automation solution should react to the failure task. A study examining the probability of missed failures (false negatives) and false alarms (false positives) showed that, for time-critical scenarios, an automation technology might fit better to handle the failure, whereas in most other cases a human should take over the automation failure [40]. The *lumberjack analogy* points out that as the level of automation increases, the performance of routine tasks improves, but the monitoring and reactivity of the human to failure scenarios significantly decreases [39].

5 Discussion

Our review of human factors research clarifies the complex interaction between humans and automation technology. We identified three focal areas: interaction prerequisites, main interaction phenomena, and interaction effects. The specific aspects of these focal areas have implications for human-automation interaction in the context of business process automation. The goal of this section is to make these links explicit and highlight how our findings can inform research on business process automation. To structure our discussion, we use the three roles related to process automation, namely process participants, process managers, and software developers.

5.1 Process Participant

The prerequisites of human-automation interaction are *understanding* and *trust*. We argue that these are equally applicable to process participants in the context of process automation. A process participant benefits from having foundational knowledge about the automation solution in semi- and fully-automated scenarios to effectively work with it. During the interaction itself, the process participant can have an *incomplete understanding* and may *misinterpret information* regarding an automation solution. This aspect is accelerated because the division of labor hinders process participants in their understanding of the whole process [8], even if no automation is in place. These aspects can be addressed through training and education initiatives that teach the process participant to work efficiently with automation technology. In addition, the effects on *mental workload* and *task allocation* are worsened, given that the process participant might take over tasks in multiple processes. The exposure to *automation change* is as relevant for the process participant. The process changes over time, defining new requirements on the technology, and corresponding changes likely impact participants and their role in the process. Therefore, relevant research questions from the perspective of the process participants include:

PP1: What are the gaps in foundational knowledge with respect to automation solutions? To address the problem of insufficient foundational knowledge, it is important to identify which aspects process participants typically struggle with and which of these aspects may lead to lower performance.

PP2: How can effective training and education initiatives be developed to support process participants who work with automation technologies? A key concern in this context will be mechanisms to develop understanding and trust, with a specific focus on process change.

PP3: What is the impact of automation change on the role and responsibilities of process participants? Change in this context may have a variety of implications. Among others, it might be necessary to reestablish trust as well as the human understanding of the automation solution. It might be interesting to also connect these aspects to typical challenges of change management, such as resistance to change.

PP4: Which factors lead to incomplete understanding and misinterpreting automation information on process performance? It is important to understand which factors may cause these issues to effectively prevent them. Possible causes may relate to human understanding, the automation design, but also cognitive factors of the human, such as mental workload.

PP5: What is the impact of task allocation on mental workload, and what are ways to optimize it in the context of process automation? The key concern in this context is not to overwhelm humans. Especially when automation technology is introduced or changed, there is a need for careful consideration of the human mental workload, such that the benefits of automation technology are not outweighed by humans struggling with an effective human-automation interaction.

5.2 Process Manager

From the standpoint of the process manager, the aspects of *transparency* and *reliability* are just as important in process automation. As automation technology should work as designed, the process manager must ensure it works reliably and communicates its results transparently. In this context, the interplay with technology requires a *human-centric design* that meets the profile and characteristics of the people involved. Additionally, as the technology in process automation might also fail, the process manager needs to manage these failure scenarios with an error-handling mechanism. This includes whether error handling should be performed by a human, such as a process manager or process participant, or by a technology that monitors automation and performs a task if a dedicated condition is encountered. Therefore, research questions from the perspective of the process manager should include:

PM1: How can process managers select a human-centric design incorporating human trademarks and needs? It is important that the process manager chooses the most suitable technology that fits the needs of their users and can work efficiently in different process contexts.

PM2: How can process managers establish a bi-directional communication channel for efficient collaboration and feedback? While the feedback loop has been emphasized as an important aspect, how can process managers establish a communication channel that allows humans to provide feedback to an automation solution and the other way around?

PM3: How can process managers handle and manage failure scenarios? Automation failure is (almost) inevitable. Therefore, it is relevant to identify and classify typical failure scenarios. So-called black swan events, i.e., rare but severe errors, might deserve particular attention in this context.

PM4: How can process managers measure the effectiveness of different error-handling mechanisms and choose the optimal approach for different types of process automation? Once error-handling mechanisms have been selected (see PM3), it is critical to understand their effectiveness. Again, the criticality of the domain might be an important contextual factor that needs to be considered.

PM5: What is the impact of automation on the process manager's role and responsibilities, and what are ways to optimize their performance in the context of process automation? Many of the questions above ultimately focus on the interaction between the user and automation technology. However, it is also relevant to understand how the roles and responsibilities of the process manager are evolving. Among others, process managers might need to monitor how *drift* affects the performance of automation technology or establish governance mechanisms for the different automation solutions employed.

5.3 Software Developers

From the perspective of a software developer, the goal is to design automation technology that meets basic design principles and is reliable, robust, and free of failures. The automation technology should provide the necessary features from the human point of view and enable individualization of the interaction. Further research from the perspective of the software developer should include the following research questions:

SD1: How can software developers implement basic design principles to ensure the usability and effectiveness of automation technology? The importance of design principles is well known, but it is necessary to understand how existing design principles affect the usability and effectiveness of automation technology.

SD2: How can software developers integrate human feedback into their development process to continuously improve it? Automation technology must evolve over time, e.g., due to drift, technological advances, or changing requirements. Finding ways to effectively integrate user feedback is necessary to continuously improve the automation solutions used.

SD3: How must software developers design automation technologies to communicate with humans efficiently and as free of interpretation as possible? Efficient communication positively affects user perception, acceptance, and comprehensibility.

SD4: What are mechanisms to implement error handling to manage failure scenarios in process automation? Effective approaches for error handling should be developed, whether performed by users or the automation technology itself. By building on the identified failure scenarios (see PM3), a link between failure scenarios and the error-handling mechanism can be established.

SD5: How can software developers choose the optimal balance between automation and human involvement for different types of processes? One of the key questions with ongoing technological advancements is how much humans still need to be involved and whether the benefits from the shift towards automation technology outweigh the cost of its implementation. A comprehensive investigation of this question for various task types and processes is imperative for an outcome-driven usage of automation technology.

The identified research directions require an empirical research agenda. Research on human-automation interaction has developed specific research strategies, such as experimental designs, simulation, observational studies, and case studies [48], which can be adopted and reused to this end. Corresponding results will provide the foundation for a theory on process automation.

6 Conclusion

This paper focused on the human aspects of process automation, recognizing that humans are critical to the success of all process automation scenarios, whether as process participants in semi-automated processes or as process managers overlooking interaction between users and automation technology. While existing research on process automation elaborates on the technological options for implementation, such as robotic process automation or workflow capabilities, we shifted the focus towards the human as a crucial factor for process execution success. To this end, we examined the journal *Human Factors* to identify relevant aspects and effects of human-automation interaction applicable to the field of process automation.

Our analysis highlighted multiple aspects of both humans and technology, which we classified into the categories prerequisites, interaction, and effects. These aspects illustrate the complexity of the relationship between technology and humans and the diverse factors that can influence their performance. We concluded that it is essential to incorporate human characteristics and trademarks into automation design, establish efficient means of communication between humans and technology, and carefully evaluate the appropriate degree of automation based on its impact on overall process performance and its ability to support human decision-making based on their cognitive abilities.

References

1. Balfe, N., Sharples, S., Wilson, J.R.: Understanding is key: an analysis of factors pertaining to trust in a real-world automation system. Hum. Factors **60**(4), 477–495 (2018)
2. Bartlett, M.L., McCarley, J.S.: No effect of cue format on automation dependence in an aided signal detection task. Hum. Factors **61**(2), 169–190 (2019)
3. Brocke, J.v., et al.: Reconstructing the giant: On the importance of rigour in documenting the literature search process (2009)

4. Collet, C., Petit, C., Champely, S., Dittmar, A.: Assessing workload through physiological measurements in bus drivers using an automated system during docking. Hum. Factors **45**(4), 539–548 (2003)
5. Cooper, H.M.: Organizing knowledge syntheses: a taxonomy of literature reviews. Knowl. soc. **1**(1), 104 (1988)
6. Corbin, J.M., Strauss, A.: Grounded theory research: procedures, canons, and evaluative criteria. Qual. Sociol. **13**(1), 3–21 (1990)
7. Douer, N., Meyer, J.: Judging one's own or another person's responsibility in interactions with automation. Hum. Factors **64**(2), 359–371 (2022)
8. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, vol. 2. Springer (2018). https://doi.org/10.1007/978-3-662-56509-4
9. Endsley, M.R.: From here to autonomy: lessons learned from human-automation research. Hum. Factors **59**(1), 5–27 (2017)
10. Eriksson, A., Stanton, N.A.: Driving performance after self-regulated control transitions in highly automated vehicles. Hum. Factors **59**(8), 1233–1248 (2017)
11. Eriksson, A., Stanton, N.A.: Takeover time in highly automated vehicles: noncritical transitions to and from manual control. Hum. Factors **59**(4), 689–705 (2017)
12. Eulerich, M., Waddoups, N., Wagener, M., Wood, D.A.: The Dark Side of Robotic Process Automation (2022). Available at SSRN 4026996
13. Feigh, K.M., Dorneich, M.C., Hayes, C.C.: Toward a characterization of adaptive systems: a framework for researchers and system designers. Hum. Factors **54**(6), 1008–1024 (2012)
14. Furlough, C., Stokes, T., Gillan, D.J.: Attributing blame to robots: I. the influence of robot autonomy. Hum. Factors **63**(4), 592–602 (2021)
15. Gibbs, G.R.: Thematic coding and categorizing. Analyzing Qual. Data **703**, 38–56 (2007)
16. Gillan, D.J., Wickens, C.D., Hollands, J.G., Carswell, C.M.: Guidelines for presenting quantitative data in HFES publications. Hum. Factors **40**(1), 28–41 (1998)
17. Heymann, M., Degani, A.: Formal analysis and automatic generation of user interfaces: approach, methodology, and an algorithm. Hum. Factors **49**(2), 311–330 (2007)
18. Innes, R.J., Howard, Z.L., Thorpe, A., Eidels, A., Brown, S.D.: The effects of increased visual information on cognitive workload in a helicopter simulator. Hum. Factors **63**(5), 788–803 (2021)
19. Ismatullaev, U.V.U., Kim, S.H.: Review of the factors affecting acceptance of AI-infused systems. Human Factors 00187208211064707 (2022)
20. Jamieson, G.A., Vicente, K.J.: Designing effective human-automation-plant interfaces: a control-theoretic perspective. Hum. Factors **47**(1), 12–34 (2005)
21. Johnson, A.W., Duda, K.R., Sheridan, T.B., Oman, C.M.: A closed-loop model of operator visual attention, situation awareness, and performance across automation mode transitions. Hum. Factors **59**(2), 229–241 (2017)
22. Kaplan, A.D., Kessler, T.T., Brill, J.C., Hancock, P.: Trust in artificial intelligence: meta-analytic findings. Hum. factors **65**(2), 337–359 (2021)
23. Lee, J.D.: Review of a pivotal human factors article:"humans and automation: use, misuse, disuse, abuse". Hum. Factors **50**(3), 404–410 (2008)
24. Ludacka, F., Duell, J., Waibel, P.: Digital transformation of global accounting at Deutsche Bahn Group: the case of the TIM BPM suite. In: vom Brocke, J., Mendling, J., Rosemann, M. (eds.) Business Process Management Cases Vol. 2, pp. 57–68. Springer, Heidelberg (2021). https://doi.org/10.1007/978-3-662-63047-1_5

25. Lyons, J.B., Vo, T., Wynne, K.T., Mahoney, S., Nam, C.S., Gallimore, D.: Trusting autonomous security robots: the role of reliability and stated social intent. Hum. Factors **63**(4), 603–618 (2021)
26. Madhavan, P., Wiegmann, D.A.: Effects of information source, pedigree, and reliability on operator interaction with decision support systems. Hum. Factors **49**(5), 773–785 (2007)
27. Marek, J., Blümlein, K., Wehking, C.: Process automation at generali CEE holding: a journey to digitalization. In: vom Brocke, J., Mendling, J., Rosemann, M. (eds.) Business Process Management Cases Vol. 2, pp. 19–28. Springer, Heidelberg (2021). https://doi.org/10.1007/978-3-662-63047-1_2
28. Marquez, J.J., Gore, B.F.: Measuring safety and performance in human-automation systems: special issue commentary. Hum. Factors **59**(2), 169–171 (2017)
29. McGuirl, J.M., Sarter, N.B.: Supporting trust calibration and the effective use of decision aids by presenting dynamic system confidence information. Hum. Factors **48**(4), 656–665 (2006)
30. van de Merwe, K., Mallam, S., Nazir, S.: Agent transparency, situation awareness, mental workload, and operator performance: A systematic literature review. Human Factors (2022). https://doi.org/10.1177/00187208221077804
31. Naikar, N., Sanderson, P.M.: Evaluating design proposals for complex systems with work domain analysis. Hum. Factors **43**(4), 529–542 (2001)
32. Noppen, P., Beerepoot, I., van de Weerd, I., Jonker, M., Reijers, H.A.: How to keep RPA maintainable? In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 453–470. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_26
33. Onnasch, L., Wickens, C.D., Li, H., Manzey, D.: Human performance consequences of stages and levels of automation: an integrated meta-analysis. Hum. Factors **56**(3), 476–488 (2014)
34. Parasuraman, R.: Designing automation for human use: empirical studies and quantitative models. Ergonomics **43**(7), 931–951 (2000)
35. Parasuraman, R., Manzey, D.H.: Complacency and bias in human use of automation: an attentional integration. Hum. Factors **52**(3), 381–410 (2010)
36. Parasuraman, R., Sheridan, T.B., Wickens, C.D.: A model for types and levels of human interaction with automation. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **30**(3), 286–297 (2000)
37. Prinzel, L.J., III., Freeman, F.G., Scerbo, M.W., Mikulka, P.J., Pope, A.T.: Effects of a psychophysiological system for adaptive automation on performance, workload, and the event-related potential p300 component. Hum. Factors **45**(4), 601–614 (2003)
38. Sarter, N.B., Woods, D.D.: How in the world did we ever get into that mode? mode error and awareness in supervisory control. Hum. Factors **37**(1), 5–19 (1995)
39. Sebok, A., Wickens, C.D.: Implementing lumberjacks and black swans into model-based tools to support human-automation interaction. Hum. Factors **59**(2), 189–203 (2017)
40. Sheridan, T.B., Parasuraman, R.: Human vs. automation in responding to failures: an expected-value analysis. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 44, pp. 1–4. SAGE Publications Sage CA: Los Angeles (2000)
41. Skraaning, G., Jamieson, G.A.: Human performance benefits of the automation transparency design principle: validation and variation. Hum. Factors **63**(3), 379–401 (2021)

42. Sylla, C., Drury, C.G., Babu, A.: A human factors design investigation of a computerized layout system of text-graphic technical materials. Hum. Factors **30**(3), 347–358 (1988)
43. Vernon, D., Reising, C., Sanderson, P.M.: Ecological interface design for pasteurizer ii: a process description of semantic mapping. Hum. Factors **44**(2), 222–247 (2002)
44. Vicente, K.J.: Ecological interface design: progress and challenges. Hum. Factors **44**(1), 62–78 (2002)
45. Victor, T.W., Tivesten, E., Gustavsson, P., Johansson, J., Sangberg, F., Ljung Aust, M.: Automation expectation mismatch: incorrect prediction despite eyes on threat and hands on wheel. Hum. Factors **60**(8), 1095–1116 (2018)
46. vom Brocke, J., Mendling, J. (eds.): Business Process Management Cases. MP, Springer, Cham (2018). https://doi.org/10.1007/978-3-319-58307-5
47. Whang, M.C., Lim, J.S., Boucsein, W.: Preparing computers for affective communication: a psychophysiological concept and preliminary results. Hum. Factors **45**(4), 623–634 (2003)
48. Wickens, C.D., Hollands, J.G., Banbury, S., Parasuraman, R.: Engineering Psychology and Human Performance. Psychology Press (2015)
49. Wickens, C.D., Li, H., Santamaria, A., Sebok, A., Sarter, N.B.: Stages and levels of automation: an integrated meta-analysis. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 54, pp. 389–393. Sage Publications Sage CA: Los Angeles (2010)
50. Wickens, C.D., Sebok, A., Li, H., Sarter, N., Gacy, A.M.: Using modeling and simulation to predict operator performance and automation-induced complacency with robotic automation: a case study and empirical validation. Hum. Factors **57**(6), 959–975 (2015)
51. Woods, D.D., Roth, E.M.: Cognitive engineering: human problem solving with tools. Hum. Factors **30**(4), 415–430 (1988)
52. Yang, E., Dorneich, M.C.: Affect-aware adaptive tutoring based on human-automation etiquette strategies. Hum. Factors **60**(4), 510–526 (2018)

Process Mining and the Transformation of Management Accounting: A Maturity Model for a Holistic Process Performance Measurement System

Simon Wahrstötter(✉)

Technical University Dortmund, 44221 Dortmund, Germany
simon.wahrstoetter@tu-dortmund.de

Abstract. Traditionally, Performance Management (PM) is considered one of the core functions of management accounting, focused on the results of business units and primarily based on financial measures. However, with the growing emphasis on process orientation and the implementation of Business Process Management (BPM), traditional PM needs to be adapted to measure what is managed, i. e. business processes. To achieve this, process-oriented organizations rely on a Process Performance Measurement System (PPMS), with Process Mining as the state-of-the-art tool for monitoring and improving processes.

In theory, the Process Mining-supported PPMS should be well integrated into the PM System (PMS), and process performance should be measured holistically, i.e. by both quantitative and qualitative figures. However, in practice it remains unclear whether these criteria are being met and whether management accounting is involved in the utilization of Process Mining and the development of a holistic PPMS. To address this research gap, a multiple case study within the German energy industry was conducted. Drawing on data from 33 semi-structured interviews, this paper presents a five-stage maturity model for the implementation of a holistic, Process Mining-supported PPMS and examines how management accounting can promote progression along this path. Due to its interdisciplinary nature, this study further contributes to research by demonstrating that the involvement of management accounting is not only beneficial to the success of Process Mining and BPM, but also crucial to the management accounting profession itself.

Keywords: Management Accounting · Process Mining · Process Performance Measurement · Business Process Management · Maturity Model

1 Introduction

In comparison to traditional companies that emphasize functional structures and hierarchy, process-oriented organizations focus on horizontal business operations ranging from customer to customer [20] and apply Business Process Management (BPM) as suitable management practice [17, 22]. Consequently, executives and process owners are reliant on a Process Performance Measurement System (PPMS) in order to analyze,

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 302–318, 2023.
https://doi.org/10.1007/978-3-031-41623-1_18

control, and optimize processes and thereby organizational performance [17, 18]. Over the last few years, Process Mining has become the state-of-the-art tool to support Process Performance Measurement (PPM) [11] and provide managers or process owners with the process intelligence they need for informed decision-making.

In traditional function-oriented organizations, Performance Management (PM) is a critical element of management accounting's "business model" that involves serving as a business partner to management, providing assistance in informed decision-making and corporate governance [16].[1] Due to the organization's general emphasis on business units, management accounting and its tools, such as the PM System (PMS), are tailored to fit this function-oriented structure. Furthermore, business performance is mainly assessed through financial measures, with qualitative aspects of performance often neglected [5, 7]. This is why Kueng [18] argues that process orientation necessitates not only a shift in organizational structure and management practices but also a transformation in traditional management accounting towards a process-oriented PMS that assesses performance holistically, i.e. measuring both quantitative and qualitative performance indicators. As Process Mining is the state-of-the-art tool to implement this kind of PPMS, it can be viewed as a potentially disruptive innovation to traditional management accounting.

When traditional companies adopt process orientation and evolve into matrix organizations, it raises an important question about the extent to which accountants embrace Process Mining as an innovative tool to transform the PMS and incorporate it into their "business model". If management accounting does not embrace the technology, it is crucial to identify who actually is responsible for providing the Process Mining-supported PPMS and how responsibilities are shared with accountants. This investigation includes examining the level of integration between the traditional PMS and the PPMS in Process Mining-using companies and the extent to which organizations measure performance holistically. The exploration of these parameters constitutes the first research question (RQ1).

The second research question (RQ2) of this paper aims to conceptualize a maturity model for a well-designed Process Mining-supported PPMS. Based on an interview-based case study, this study will identify the key challenges associated with each maturity stage, as well as the strategies required to overcome them. As a result, the maturity model will provide practical guidance for companies to facilitate progression along this path and develop an effective Process Mining-supported PPMS.

To address these research questions, the subsequent sections of this paper are structured as follows. Section 2 briefly discusses related work and highlights existing research gaps. In Sect. 3, the methodological approach is outlined, while Sect. 4 presents the findings of the multiple-case study. Section 5 concludes the paper by summarizing the results and proposing avenues for future research.

[1] This applies in particular to the profession of "Controlling", which represents a form of management accounting special to German-speaking regions.

2 Related Work and Research Gaps

Process Mining is utilized to extract event logs from information systems to discover, monitor, and improve processes [25, 26]. Due to the exponential growth in the number of companies employing the technology in recent years [13], it has become an integral element of BPM [28]. This is largely attributed to its effectiveness in supporting PPM, which is widely acknowledged as a key determinant of BPM success [4, 17, 21, 24]. The Process Mining-supported PPMS collects and disseminates key performance indicators (KPIs) for different dimensions, such as time, quality, and costs, to provide evidence-based insights into process performance [10, 18]. However, studies on PPM are relatively scarce, particularly compared to the considerable research conducted on PM in general business context [1, 7]. Moreover, research on Process Mining has primarily focused on its technical aspects [13, 33], resulting in a significant gap in understanding how organizations adopt and integrate this technology into their existing systems and the effects of such integration [28].

This lack of understanding presents a challenge, as BPM requires adapting the existing PMS to measure performance holistically and serve process-oriented companies [1, 2, 7, 18, 31]. While the widely recognized Balanced Scorecard incorporates process performance as one of its fundamental pillars, its primary focus remains on goal setting and strategic alignment. In contrast, Process Mining enables companies to measure and achieve predefined target values operationally, making its integration into the PMS essential. Unfortunately, there is a "missing link" between management accounting, which is responsible for the traditional PMS and accounting system, and BPM [23, 30]. A thorough literature review on the relationship between accounting and BPM discovered only 39 papers, with none of them related to financial accounting and none of them published in accounting-specific journals [2]. This lack of attention means that proposals for process-aware accounting systems primarily emerge from BPM research and are limited [23, 29]. Vice versa, studies in the BPM field often fail to identify accountants as an impacted party of Process Mining adoption [28].

The present study seeks to address this research gap by collecting empirical data on how companies actually use Process Mining and integrate the Process Mining-supported PPMS into their existing PMS. By doing so, it responds to central research calls formulated by prominent scientists in the BPM field [2, 13, 27, 28]. Furthermore, this study goes beyond previous work, which has largely been limited to theoretical conceptions or literature reviews. Ultimately, it offers practical guidance for companies seeking to integrate Process Mining technology into their operations. As more organizations embrace process orientation and Process Mining, the issue of proper integration becomes increasingly important.

3 Methodology

3.1 Research Design

Given the lack of understanding of how organizations integrate Process Mining into their existing PMS and why such integration can be challenging, this study employed a qualitative case study approach to explore this complex and emerging issue [32]. Building upon the theoretical foundations of empirical social research, the utilization of qualitative research is deemed appropriate as it primarily focusses on capturing individual perspectives and experiences of the involved parties, providing valuable insights for gaining a thorough understanding of social phenomena [6]. To provide a comprehensive analysis, a sample of 15 German companies was compiled, comprising seven private utilities, five municipal utilities, and three sector-specific consultancies. The energy industry was selected for various reasons: The liberalization of energy markets and decentralization of electricity generation have necessitated utilities to become more customer-centric and therefore process-centric. However, empirical studies investigating the implementation of BPM and Process Mining within utilities are scarce, highlighting the need for a more detailed examination of this industry. Lastly, the researcher's affiliation with one of the companies enabled an in-depth case study and facilitated access to other utilities. The cases were identified and chosen through a selective sampling and snowballing approach, that leveraged the principles of similarity and contrast to yield profound insights into the research questions [12]. Specifically, the sample was composed of organizations linked to the energy industry and familiar with the implementation of Process Mining. However, the organizations varied in business area, ownership status, company size, and duration of technology usage. Appendix 1 lists the selected companies and their specific characteristics.

3.2 Data Collection

The data collection for this study involved conducting 33 in-depth expert interviews across the 15 selected companies.[2] To ensure consistency, a semi-structured interview guideline organized the interviews into five themes: the company and its process orientation, BPM, the PMS and the PPMS, Process Mining, and roles and capabilities. Consistent with the methodology of qualitative research, the questions were predominantly open-ended, aiming to elicit detailed and nuanced responses. For example, the interviewees were queried about the responsible unit for implementing Process Mining, the influence of Process Mining on the (P)PMS and management accounting, as well as the key challenges observed during the usage of the technology.

The length of one interview was 83 min on average, resulting in approximately 46 h of interview material, which was recorded and transcribed verbatim with the consent of the interviewees. The extensive case study in one of the selected utilities, comprised of 19 interviews and supplementary archival data, incorporated a "variety of voices" [19] by involving individuals in different positions related to the use of Process Mining, e. g. the CFO, executives from accounting, line managers, process owners and experts, process analysts, and data scientists. Interviews with representatives from other utilities primarily engaged executives or members of the process analyst team, such as those affiliated with a Center of Excellence (CoE) for Process Mining. An overview of the conducted interviews and respective interviewees is presented in Appendix B.

[2] Since the interviews were conducted in German, the quotes used in Sect. 4 have been translated into English. To maintain transparency and avoid any loss of meaning, the original German quotes are included in Appendix C.

3.3 Data Analysis

To analyze the data, the transcripts were subjected to a qualitative content analysis, which is a widely accepted scientific approach for deriving systematic and valid inferences from text [6]. MAXQDA was utilized to code the material in a hybrid approach, combining deductive and inductive coding. Initially, nine broad categories were derived from related work, the research questions, and the interview guide, resulting in an a-priori coding system. Following this, ten interviews were used to revise and specify these categories by building subcategories through inductive coding. Based on this second round of coding, a formalized coding rulebook was developed, including examples to ensure consistency in the analyses. Finally, the rulebook was used to code the entire interview material, resulting in the identification of 5108 codes, organized into seven main and 150 sub-categories.

4 Findings

4.1 Process Mining as Enabler of Effective PPM

The empirical data highlights the importance of Process Mining in supporting PPM and the entire BPM concept. In many companies, the introduction of a PPMS coincides with the adoption of Process Mining technology, causing interviewees to use the terms "Process Mining" and "PPM" interchangeably. Only a few of the larger companies, which started their BPM journey in the midst of the last decade, used some non-digital PPM tools like the Balanced Scorecard and collected process-relevant KPIs beforehand. However, these were deemed inadequate for operational control: "*And all these metrics that we collect now, regarding process adherence, lead times, automation levels, previously it was not possible to collect them. […] For example, we can now continuously and promptly monitor our performance levels, identify processing deficits, assess where we are failing to achieve our processing rates, and control our resources. Process mining enables us to do all of this, which was not possible before.*" [PO2] In consequence, the emergence of Process Mining revolutionizes PPM entirely, also triggering a thorough revision of the existing BPM concept. Some utilities are even using Process Mining to leapfrog, with process documentation and the installation of process owners only beginning with the use of the technology or being omitted completely. In summary, Process Mining is seen as the very enabler of an effective PPMS and a state-of-the-art tool for BPM.

The study further reveals a typical path for implementing Process Mining technology. After performing a proof of concept (PoC) on a specific process, often supported by a consultancy, a project-based roll-out-phase follows. As the potential of the technology is quickly recognized, the number of requesters and demands rises sharply. To account for this, the deployment of the technology is professionalized, with almost all companies in the sample assigning the operation and further development of the tool to permanent positions, teams, or even organizational units, e.g. CoEs. The employees entrusted with the Process Mining-supported PPMS, often called process analysts, consider Process Mining to be their daily business, fully utilizing their capacity: "*How often do we use it? Every day, throughout.*" [U10].

The study identifies three basic types of organization for how BPM is realized in practice and how the collaboration of process owners and process analysts is structured. The predominant approach is to centralize process analysts in a dedicated team, who provide their PM service to fixed process teams, agile teams or respond to ad-hoc requests from process owners or business units. U1 represents the most formalized example of this type, where the entire business is mapped by twelve core processes, each divided into a different number of sub-processes and borne by a process owner. To control, operate, and optimize the processes, process owners are supported by a cross-functional team of different roles, including a process analyst, a process manager, and various process experts from the business units involved in the process. The second type of organization is used by only three utilities, where the process analysts are decentralized and part of the operational unit or segment responsible for the analyzed processes. This approach is utilized when the deployment of the technology is limited to only one segment or unit, without its use across the entire organization. As a third type, two of the companies implement a hybrid approach, involving a centralized Process Mining team complemented by a human resource deployed as a satellite or citizen developer in the operational units: "*We have started to develop the concept of Citizen Developers, which means that in the departments receiving the analysis, at least one colleague is designated to maintain the dashboards. This approach has proven successful.*" [U4] The finding that most companies organize the Process Mining-supported PPMS in a centralized team raises the important question of where exactly this unit of process analysts is being installed and how they cooperate with the PM unit in charge of the traditional PMS.

4.2 Organizational and Functional Fragmentation

Consistent with theory [5], interviewees across all companies in the study acknowledge a functional overlap between the Process Mining-supported PPMS and the PMS. This overlap is grounded in the shared objective of gathering information and KPIs to enhance transparency, measure and increase performance, and effectively support managerial decision-making: "*Looking at it from a management accounting perspective, one approaches it from a performance standpoint: What is the quality of the processes and, of course, what efficiency do certain processes deliver*?" [MA] However, only U1 considers this common goal as an opportunity to integrate the team of process analysts into the management accounting department. Both the CFO and the head of PM argue, that Process Mining is a powerful tool for performing the business partner role and optimizing corporate management comprehensively. Hence, its integration into management accounting is viewed as consistent and empowering.

In contrast, all other utilities surveyed do not assign process analysts to the management accounting department. Instead, they are centralized in a process management unit, in the business development department, in an IT or digitization team, or assigned to the operational units themselves. In light of these findings, it can be inferred that organizational fragmentation of the traditional PMS and the Process Mining-supported PPMS is prevalent across most companies.

Admittedly, a cohesive (P)PMS[3] can still be achieved despite organizational fragmentation, given there is effective cooperation between accountants and process analysts. However, even in U1, where process analysts are assigned to management accounting, the functional integration of the PMS and the PPMS is deemed insufficient. For one thing, this is evident from process owners and experts not recognizing process analysts as part of management accounting. Secondly, some interviewees envision a future where accountants and process analysts merge to become a full-service provider, offering executives a comprehensive (P)PM solution, but currently view the two roles as "entirely decoupled": "*Actually, I pursue a holistic approach, where we not only focus on PPM but on PM as a whole. Currently, we have this in only a few units. [...] Especially in technical units, the analysts are entirely decoupled from the operational accountant*." [PA4] Hence, it is apparent that organizational integration alone does not guarantee effective collaboration and the emergence of a holistic (P)PMS. Conversely, being organizationally separated, as seen in the other utilities surveyed, makes effective collaboration even more challenging. This can be inferred from interviewees reporting a lack of interest from traditional management accounting in process orientation, Process Mining technology, or supporting process owners with relevant information. In consequence, in many companies surveyed, there is no collaboration at all.

These findings make important to understand how the PMS and PPMS are functionally separated or how management accountants and process analysts share tasks. This question can be answered consistently across all companies in the survey. The line of fragmentation is drawn between financial measures, including costs in particular, and all other (process) performance indicators. Traditional accountants are seen as the "keeper of the numbers", responsible for managing budgets of functional units to ensure financial targets are met. Process-oriented PM, on the other hand, lacks this perspective and fails to provide sufficiently precise quantification of value contributions and costs of individual activities and processes. As evidenced by other studies before [23, 29], there is a lack of suitable process-aware accounting systems, with only one private utility (U6) having a dedicated activity-based costing (ABC) system ready for operational use. This deficiency also prevents the aggregation of process-related measures into common overall corporate financial indicators, a balance sheet or a profit and loss statement: "*That is indeed a stage we have not yet reached. We have indicators for the performance of processes. But at some points – and this is the step that is needed – I clearly have to elaborate which real value levers are involved. How do they contribute to my overall results, overall return, overall cash flow, and the financial success of the company?*" [MA] Instead, process analysts focus on more qualitative measures that accountants are not typically interested in, such as throughput time, process quality, customer satisfaction and the rate of processing and automation. As pointed out by some interviewees, improving these qualitative measures has the potential to positively improve costs and other financial measures, making them useful to management accounting. However, in order to fully capitalize on this potential, a stronger cooperation is needed between process analysts and accountants.

[3] This notation highlights the intrinsic relationship between the PMS and PPMS, emphasizing that any fragmentation or integration has a simultaneous impact on both.

In response to RQ1, the empirical findings yield several notable key takeaways: Process Mining is regarded as the cutting-edge tool that enables effective PPM and enhances BPM. However, despite the potential for a more holistic approach to measuring performance, this study reveals a significant lack of integration between the process analyst team and the traditional management accounting department, resulting in both organizational and functional fragmentation of the (P)PMS. The parting line between these two teams runs along the collected key figures, with management accounting responsible for financial measures and process analysts focused on more qualitative, process-related indicators. Instead of measuring performance holistically, there are two separate systems that measure performance independently, ultimately failing to provide a complete picture of overall performance. Consequently, instead of one comprehensive business partner providing full-service to management, there are two separate entities, one advising on financial matters, and the other supporting with process-related issues.

4.3 A Five-Stage Maturity Model for a Holistic and Fully Integrated Process Mining-Supported PPMS

After examining the current situation of the Process Mining-supported PPMS in the surveyed companies, this chapter proposes actionable steps towards utilizing the full potential of Process Mining, promoting closer collaboration between accountants and process analysts, and achieving a truly holistic view of performance. To this end, a five-stage maturity model is presented. By identifying key challenges at each stage and presenting strategies for overcoming them, this section demonstrates a roadmap for the development of a holistic and fully-integrated (P)PMS. Gaining an understanding of these stages, enables other companies to assess their current maturity level and take strategic steps towards advancing their PPMS to the next tier. An illustration of the five stages of the maturity model and the corresponding classification of the surveyed companies is presented in Fig. 1.

Stage 1: (Process Mining-Supported) PPMS Missing. Companies in this stage either do not measure process performance at all or are not utilizing Process Mining to support their PPMS yet, including organizations currently conducting a PoC. Although there might be some sort of process orientation and BPM, process performance is not assessed based on data. The most significant challenge for implementing the technology is convincing different stakeholders, first and foremost management. As organizations might be new to process orientation and BPM in general, it is crucial to present the advantages of these concepts, such as increased operational efficiency and reduced costs, enhanced customer orientation and experience, or facilitated agility and innovation [4]. Once convinced, it should be argued that successful BPM necessitates an effective PPMS, as companies have to measure what is managed. Other studies have already shown that the introduction of process owners has no positive effect on operational performance if a PPMS is missing [17]. Since non-digitized methods of process analytics, unlike Process Mining, suffer from severe deficiencies, an effective PPMS requires the introduction of Process Mining. All in all a Process Mining-supported PPMS can be seen as prerequisite for effective BPM.

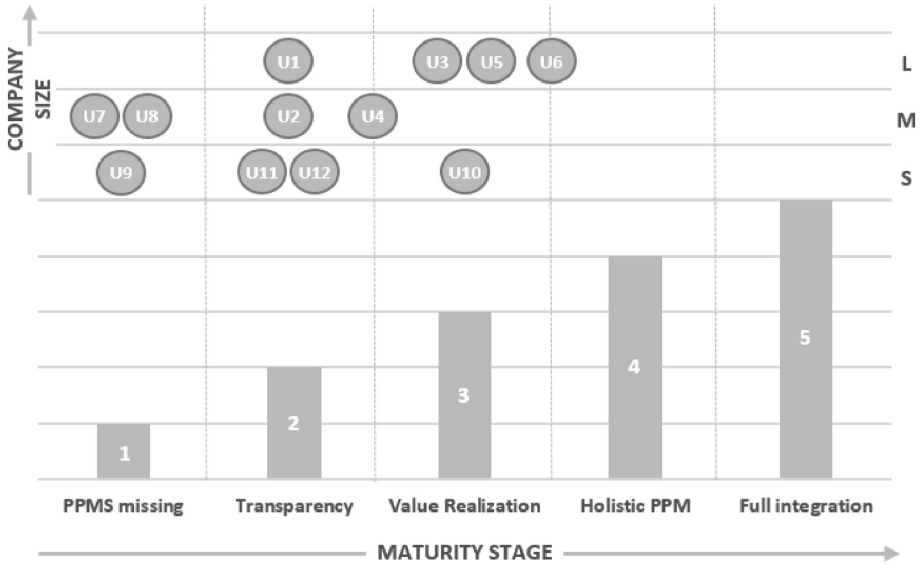


Fig. 1. Five-stage maturity model and classification of the examined utilities. (S*mall*: < 1.500 employees | M*edium*: 1.500–3.000 employees | L*arge*: > 3.000 employees)

Convincing management includes justifying the costs induced by implementing Process Mining. Therefore, obtaining the approval of needed financial resources is the second significant issue of this stage. To demonstrate the benefits of the technology to management and other stakeholders, interviewees recommend to conduct a PoC using a digitized mass process that is already documented. Comparing the documented process with the process variants identified by the technology is a powerful tool for persuading management. However, as translation of qualitative process measures into financial measures is not easy, valuing the investment in Process Mining upfront is difficult. Yet, according to a number of interviewees, the adoption of Process Mining is a matter of corporate philosophy, not a financial issue. Specifically, they argue that process-oriented companies committed to BPM are dependent on the deployment of Process Mining as a fundamental tool for performance analysis and optimization. Therefore, the technology should be regarded as an essential business enabler, much like Excel, rather than a discretionary expense: "*You might have to think of it as an instrument that has more of an SAP CO kind of character, where you say: 'This is the tool or equipment of a process analyst to help with steering the company.' And would you really do a cost-benefit analysis for CO*?" [PM].

Stage 2: Transparency. In the second stage of the maturity model, companies have implemented Process Mining technology for process transparency and PM. However, they are not utilizing the KPIs to optimize or automate their processes, hence not fully realizing the potential benefits of the tool. One of the primary reasons for this is that companies, particularly smaller ones with limited expertise and IT resources, can become overwhelmed by the technical complexity of using Process Mining. Gathering data from different IT systems, producing data models, and checking data quality can be a complex process, and poor data quality can lead to false information and KPIs, causing mistrust and refusal to use the technology. To overcome these challenges, close collaboration between process analysts, process experts who know the operational IT systems, and data scientists is necessary. Therefore, some companies integrate data scientists in the process analyst team, while others train process analysts to adopt the data scientist role on top. Either way, the assigned resources should be free to fully focus on Process Mining.

Producing acceptance of the technology represents another significant challenge in this stage. According to interviewees, employees must learn to manage the transparency provided by Process Mining, as they may fear performance audits and ramifications due to the tool's potential to provide visibility into task execution and duration. Additionally, the technology's objective to increase operational efficiency and minimize costs, eventually leading to process automation, contributes to employee anxiety about job security. These concerns emphasize the importance of early engagement with the working council to jointly develop regulations for the use of Process Mining, particularly in German-speaking countries. To further increase acceptance, it is crucial to establish a relationship of trust not only between the process analyst team and the operational units but also among different operational divisions. Transparency in cross-functional processes represents a new and challenging concept compared to measuring performance within one's own business unit. Therefore, it is of utmost importance to avoid using the tool for monitoring employees or for finger pointing, as interviewees unanimously explain that these practices can quickly erode trust and create resistance to the technology: "*If I use Process Mining to engage in finger pointing, then it's just a flash in the pan. [...] If it's used for that purpose, it's already over.*" [PA1] Instead, an advanced feedback and error culture needs to be established. This entails the development of effective communication channels with the process analysts demonstrating high social skills and effective collaboration. The representative of U10 puts it short and simple: "*Who would let someone else manage his own processes if that person was an asshole? Nobody!*".

In this early stage of usage, it is not advisable to overwhelm employees with the full potential of the technology by overemphasizing buzzwords such as machine learning, AI, and process automation, as they may trigger fear and hinder the acceptance of the technology. Instead, it is crucial for employees to become familiar with the technology's capabilities first and develop a basis of trust along with a positive feedback culture. Furthermore, following a comprehensive review of potential data quality issues, employees must gain proficiency in understanding the provided process-related information and KPIs before utilizing the technology for optimization and automation purposes. Therefore, to mitigate the risk of long-term acceptance issues, it is advisable to build a strong foundation in stage 2 instead of leaping to stage 3.

In light of the aforementioned challenges, interviewees emphasize the need for companies to recognize that Process Mining is not a turnkey solution, but rather necessitates a consistent effort, along with significant human resources. This entails the necessity of having process analysts who possess a profound understanding of the operational processes. The surveyed companies utilize strategies such as assigning fixed process analysts to process teams, establishing satellites in the operational units or conducting job rotations to facilitate this. Many interviewees acknowledge underestimating the effort required from operational units and process experts, who need to contribute their specialized expertise.

Stage 3: Value Realization. The third stage of the maturity model marks the point where companies move beyond using Process Mining for transparency and KPI gathering, and begin to utilize it to set performance goals and improve real processes. This stage offers companies the opportunity to realize value by increasing efficiency, reducing costs, and enhancing automation and processing rates. However, interviewees report several challenges regarding value realization**.** One significant issue is that managers and process owners may find satisfaction with the transparency provided by Process Mining and solely associate value with the technology as a business intelligence tool, offering faster and easier access to information. This tendency is linked to a lack of accountability and responsibility for process performance, which is particularly prominent in matrix organizations, where process owners have limited control over business units and their employees, requiring the consent of line managers to allocate resources and embrace optimization projects. In U1, where process owners also function as line managers, process owners tend to focus more on the process part that can be influenced through their business unit and their role as a line manager, as opposed to the overall process: "*By leaving the processes somewhat out of the target agreement processes, it's not really transparent what is being achieved. […] As long as this is the case, everyone will continue to think only in terms of their functional units and try to optimize them.*" [PM] This limitation of control makes it difficult for process owners to assume responsibility for process performance and drive optimization, given their dependence on other executives who may block their efforts. To address this issue, companies can employ strategies such as empowering process owners, setting process-related goals, and evaluating the performance of process owners and managers based on process performance. Therefore, management commitment is essential in this stage, particularly in resolving issues and conflicts between process owners and unit executives. To solve problems and foster process thinking, some of the surveyed companies install bodies such as a process board comprising management, unit executives, and process owners.

Having assertive process owners who feel genuinely responsible for process performance is crucial for yet another reason. With the assistance of the Process Mining-supported PPMS, process analysts can only assess performance, identify deficiencies, and propose countermeasures to improve the process. However, despite the process analysts' skills and the tool's capabilities, most of the potential optimization will not materialize without modifications in the operational units instructed by a process owner. This emphasizes the significance of laying a strong foundation for effective collaboration in the second stage and underscores the maturity model's sequential nature. Nonetheless, for process analysts to generate meaningful analyses and effective proposals for optimization, they require active involvement, sufficient time, excellent process knowledge and thinking, and a thorough understanding of the operational business units. Given these rigorous demands, interviewees suggest strategies such as job rotations and placing citizen developers within operational units.

Since it is true that optimization of quantitatively measured process performance is feasible, interviewees explain that many ideas to optimize processes do not require quantitative figures for implementation. However, the absence of a sophisticated ABC system and a process-aware accounting system represents another challenge in this third maturity stage. The lack of quantitative measures makes it difficult to determine the value of process optimizations, and convince process owners, unit executives and management to prioritize proposed process-related projects and allocate resources [23]. This is especially problematic since resources are often scarce and can be utilized for other optimization efforts. For instance, management tends to allocate resources where value can be measured quantitatively, such as positive net present value (NPV) projects, rather than projects with unquantified benefits, such as reducing throughput time by two days. When faced with multiple optimization options, companies cannot be sure that they are investing in the best option without quantitative measures. To enhance process owner accountability, reinforce process optimization projects with facts, and make informed investment decisions, process-oriented companies need an advanced ABC system and a process-aware accounting system.

Stage 4: Holistic PPM. Companies in the fourth stage of the maturity model measure process performance holistically using both qualitative and quantitative KPIs. With the assistance of a sophisticated ABC system, companies on this level can convert qualitative process performance measures into financial metrics such as calculating the cost savings resulting from reductions in throughput time or increases in automation rates. Assigning costs and profits to individual activities and cases running through processes allows them to evaluate the NPV of investments into process optimization and automation projects, and justify the costs for Process Mining by calculating the value it generates.

An important challenge for realizing holistic PPM is measuring the actual time employees spend on specific tasks, known as handling time. This differs from measuring throughput time, as tasks may not be fulfilled immediately, resulting in wait times. However, collecting data on handling time can be difficult due to technical limitations in IT systems that may not capture this information at a granular level, thereby constraining measurements to throughput time. Moreover, there may be resistance from executives and employees in sharing this level of transparency if IT systems of others are not equally transparent. However, employee rights and works committees pose an even bigger challenge to measuring handling time, with interviewees indicating that it results in severe resistance and leads to failure of the technology. Companies can instead estimate handling time and assign personnel costs based on this estimation to identify the process variants, activities, or suppliers that induce the highest costs, allowing for optimization actions to be evaluated quantitatively. While this approach may be less precise, starting with this alternative can improve estimations and enhance organizational learning. The surveyed companies that have implemented this strategy report undergoing a trial-and-error phase, testing and revising the granularity and useful applications of their ABC system. Consequently, it is recommended for companies to enter a similar trial-and-error phase, develop a minimum viable product, test it in real life, and exchange experiences with other companies facing comparable problems.

Integrating costs and other financial measures into processes and developing a process-aware accounting system is a critical challenge for realizing a holistic PPM. This challenge typically falls within the realm of management accounting, with ABC being an accounting tool since its inception. While companies can achieve stage 3 of the maturity model without the assistance of accountants, their expert knowledge would be invaluable in advancing to stage 4 and 5. Interviewees even report that process analysts expect accountants to develop a sophisticated ABC system and therefore do not engage in it themselves. However, the empirical study reveals a significant organizational and functional fragmentation between the PPMS and PMS, with a severe lack of collaboration between process analysts and accountants. Many interviewees attribute this lack of collaboration to the low interest of management accounting in process orientation in general and process-related or qualitative measures in particular, leading to accountants who still view PPM with suspicion: "*Our management accounting is not very process-oriented, but rather focused on budget and investment. [...] In terms of processes, we have no exchange with management accounting, no. Exchange would only make sense, after all, if they also considered the process perspective.*" [U10] This empirical evidence reflects the disinterest of accounting that has been identified in research (cf. Sect. 2).

A practical strategy to promote collaboration between process analysts and accountants is the integration of process analysts into the management accounting team. However, as demonstrated by the study, organizational integration alone does not guarantee close cooperation. Instead, both roles must work together as a team, sharing knowledge and expertise. Additionally, the integration of quantitative measures and the implementation of a process-aware accounting system necessitate training for both process analysts and accountants. Consequently, as companies progress to stage 4 of the maturity model, the role of process analysts transforms into that of an "analytical process accountant", which imposes even greater demands on their skills and expertise.

From a socio-technical perspective, it is important to comprehend the underlying reasons for accountants' resistance towards Process Mining. The technology and the development it triggers result in the creation of a second PMS that operates in concurrence to their own. The presence of a parallel PMS may be perceived as a threat to their established "business model", and supporting the PPMS would mean undermining their function-oriented PMS. Management accountants have traditionally been the sole business partners to management and the sole measurers of performance, and thus desire to maintain their expertise and status quo. Therefore, the development of a sophisticated ABC-system and process-aware accounting system signifies a transformation of traditional management accounting and generates fears, leading to acceptance problems.

Stage 5: Full Integration. At this stage of the maturity model, companies do not only measure performance holistically, but through a fully integrated (P)PMS. This integration means that there is no longer a function-oriented PMS separate from a Process Mining-supported PPMS. By analyzing the monetary outcomes resulting from process optimization, organizations can effectively incorporate the insights obtained from Process Mining into the short- and medium-term financial plans of traditional accounting practices. Companies at this level of maturity further use their sophisticated ABC system to give budgets to processes and process owners, rather than to business units and line executives. This shift in budgeting allows for a higher level of responsibility and accountability within BPM. Some organizations may have developed their process-aware accounting system to the point where they can draw general financial indicators, a balance sheet, and a profit and loss statement from the system. While not every company may need to go this far or fully replace its function-oriented organizational structure, a process-oriented transformation of the accounting system is essential to realizing a fully process-oriented company, as envisioned by Hammer [14, 15] and Davenport [8, 9] over thirty years ago. Without this transformation, managers have no option but to rely on the traditional accounting system, which uses business units as cost centers for creating balance sheets and financial reports.

With none of the surveyed companies having advanced to this level of maturity (cf. Fig. 1), the challenges and corresponding strategies to overcome them cannot be directly inferred from the interview data. However, progressing to this level demands a more extensive transformation than that required in stage 4, thus presenting a magnified version of the corresponding acceptance issues. Yet, it is advisable that accountants overcome these issues and participate in the transformation of the accounting system. Otherwise, process analysts may develop a sophisticated ABC system and create a process-aware accounting system on their own, gradually rendering accountants and traditional PM obsolete. Therefore, it is in management accounting's best interest to adopt Process Mining technology and transform their PMS and accounting system accordingly. The integration of the Process Mining-supported PPMS is crucial to the management accounting profession itself.

5 Conclusion and Future Work

As one of the first, this paper aimed to bridge the research gap on the relationship between management accounting and BPM in the context of process-oriented organizations that use Process Mining to support PPM. By conducting 33 interviews across 15 companies from the German energy industry, this study examined how companies integrate the technology into their existing PMS and identified the effects and challenges that arise from this adoption. Regarding the level of integration as well as the extent of collaboration between accountants and process analyst, the findings revealed a significant organizational and functional fragmentation of the (P)PMS, impeding the development of a sophisticated ABC system, a process-aware accounting system, and holistic PPM. In light of these results, this paper presented a five-stage maturity model that provides companies with evidence-based guidance on proper integration, emphasizing that progression along this path is not only beneficial to Process Mining success and BPM, but also crucial to the management accounting profession. By doing so, this study contributes to the previously overlooked yet emerging research stream focused on the organizational and managerial aspects of Process Mining with the aim of maximizing the business value obtained from this technology. Notable examples of papers within this research stream include references [3, 11, 13, 33].

Notwithstanding the valuable insights derived from this study, it is important to acknowledge its limitations. First, the sample and scope were confined to a specific industry and country, thereby compromising the generalizability of the findings. Additionally, the research did not employ any particular theoretical frameworks. Consequently, the findings and limitations of this study offer considerable potential for future work. One possible research inquiry is to evaluate the degree of integration between the PMS and the Process Mining-supported PPMS across diverse industries and geographical regions, which could lead to the refinement and enhancement of the maturity model if necessary. The identification of companies with higher levels of maturity could provide opportunities for gaining insights into the process of developing a sophisticated ABC and a process-aware accounting system. To augment the understanding, the incorporation of conceptual papers is recommended, specifically from accounting research. Furthermore, the application of theories can lead to a more thorough analysis of each stage to better comprehend the challenges and strategies required to overcome them. For instance, to investigate the acceptance issue in stage 1, theories such as the technology acceptance model or the model of task technology fit can be utilized. Given that Process Mining is a disruptive technology to traditional management accounting, innovation theories can be leveraged to search for compensatory strategies in stage 4.

Appendix

Supplementary material for this article is available online at https://bit.ly/448dPAR. It contains three documents: Appendix 1 presents the composition of the case study sample. Appendix 2 provides an overview of the conducted interviews and respective interviewees, and Appendix 3 lists the used quotes together with the German originals.

References

1. Abeygunasekera, A.W.J.C., Bandara, W., Wynn, M.T., Yigitbasioglu, O.: Process performance measurement system characteristics: an empirically validated framework. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNBIP, vol. 360, pp. 211–227. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26643-1_13
2. Abeygunasekera, A.W.J.C., et al.: Nexus between business process management (BPM) and accounting. Bus. Process. Manag. J. **24**(3), 745–770 (2018). https://doi.org/10.1108/BPMJ-12-2016-0235
3. Badakhshan, P., et al.: Creating business value with process mining. J. Strateg. Inf. Syst. **31**(4), 101745 (2022). https://doi.org/10.1016/j.jsis.2022.101745
4. Bai, C., Sarkis, J.: A grey-based DEMATEL model for evaluating business process management critical success factors. Int. J. Prod. Econ. **146**(1), 281–292 (2013). https://doi.org/10.1016/j.ijpe.2013.07.011
5. Blasini, J., Leist, S.: Success factors in process performance management. Bus. Process. Manag. J. **19**(3), 477–495 (2013). https://doi.org/10.1108/14637151311319914
6. Bortz, J., Döring, N.: Forschungsmethoden und Evaluation. Für Human- und Sozialwissenschaftler, 5th edn. Springer-Lehrbuch. Springer, Heidelberg (2016)
7. Choong, K.K.: Are PMS meeting the measurement needs of BPM? A literature review. Bus. Process. Manag. J. **19**(3), 535–574 (2013). https://doi.org/10.1108/14637151311319941
8. Davenport, T.H.: Process Innovation. Reengineering Work Through Information Technology. Harvard Business School Press, Boston, Mass (1993)
9. Davenport, T.H., Short, J.E.: The new industrial engineering: information technology and business process redesign. Sloan Manage. Rev. **31**(4), 1–31 (1990)
10. Dumas, M., et al.: Fundamentals of Business Process Management. Springer, Heidelberg (2018)
11. Eggers, J., Hein, A., Böhm, M., Krcmar, H.: No longer out of sight, no longer out of mind? how organizations engage with process mining-induced transparency to achieve increased process awareness. Bus. Inf. Syst. Eng. **63**(5), 491–510 (2021). https://doi.org/10.1007/s12599-021-00715-x

12. Eisenhardt, K.M.: Building theories from case study research. Acad. Manag. Rev. **14**(4), 532 (1989). https://doi.org/10.2307/258557
13. Grisold, T., et al.: Adoption, use and management of process mining in practice. Bus. Process. Manag. J. **27**(2), 369–387 (2021). https://doi.org/10.1108/BPMJ-03-2020-0112
14. Hammer, M.: Reengineering work - don't automate, Obliterate. Harvard Bus. Rev. **68**(4), 104–112 (1990)
15. Hammer, M., Champy, J.: Reengineering the Corporation. A Manifesto for Business Revolution, 1st edn. HarperBusiness, New York, NY (1993)
16. Horváth, P., Gleich, R., Seiter, M.: Controlling, 14th edn. Verlag Franz Vahlen, München (2020)
17. Kohlbacher, M., Gruenwald, S.: Process ownership, process performance measurement and firm performance. Int. J. Product. Perf. Mgmt. **60**(7), 709–720 (2011). https://doi.org/10.1108/17410401111167799
18. Kueng, P.: Process performance measurement system: a tool to support process-based organizations. Total Qual. Manag. **11**(1), 67–85 (2000). https://doi.org/10.1080/0954412007035
19. Myers, M.D., Newman, M.: The qualitative interview in IS research: examining the craft. Inf. Organ. **17**(1), 2–26 (2007). https://doi.org/10.1016/j.infoandorg.2006.11.001
20. Reijers, H.A.: Implementing BPM systems: the role of process orientation. Bus. Process. Manag. J. **12**(4), 389–409 (2006). https://doi.org/10.1108/14637150610678041
21. Rosemann, M., vom Brocke, J.: The six core elements of business process management. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1. IHIS, pp. 105–122. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-642-45100-3_5
22. Škrinjar, R., Trkman, P.: Increasing process orientation with business process management: critical practices'. Int. J. Inf. Manage. **33**(1), 48–60 (2013). https://doi.org/10.1016/j.ijinfomgt.2012.05.011
23. Sonnenberg, C., vom Brocke, J.: The missing link between BPM and accounting. Bus. Process. Manag. J. **20**(2), 213–246 (2014). https://doi.org/10.1108/BPMJ-12-2012-0136
24. Trkman, P.: The critical success factors of business process management. Int. J. Inf. Manage. **30**(2), 125–134 (2010). https://doi.org/10.1016/j.ijinfomgt.2009.07.003
25. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_19
26. van der Aalst, W.: Process Mining. Data Science in Action, 2nd edn. Springer, Heidelberg (2016)
27. vom Brocke, J., Jans, M., Mendling, J., Reijers, H.A.: Call for papers, issue 5/2021. Bus. Inf. Syst. Eng. **62**(2), 185–187 (2020). https://doi.org/10.1007/s12599-020-00630-7
28. vom Brocke, J., Jans, M., Mendling, J., Reijers, H.A.: A five-level framework for research on process mining. Bus. Inf. Syst. Eng. **63**(5), 483–490 (2021). https://doi.org/10.1007/s12599-021-00718-8
29. vom Brocke, J., Recker, J., Mendling, J.: Value-oriented process modeling: integrating financial perspectives into business process re-design. Bus. Process. Manag. J. **16**(2), 333–356 (2010). https://doi.org/10.1108/14637151011035633
30. Vukšić, V.B., Bach, M.P., Popovič, A.: Supporting performance management with business process management and business intelligence: a case analysis of integration and orchestration. Int. J. Inf. Manage. **33**(4), 613–619 (2013). https://doi.org/10.1016/j.ijinfomgt.2013.03.008

31. Wieland, U., et al.: Process performance measurement system – towards a customer-oriented solution. Bus. Process. Manag. J. **21**(2), 312–331 (2015). https://doi.org/10.1108/BPMJ-04-2014-0032
32. Yin, R.K.: Case Study Research and Applications. Design and Methods. Supplementary Textbook. SAGE Publications, Inc, [Washington, D.C.] (2018)
33. Zerbino, P., Stefanini, A., Aloini, D.: Process science in action: a literature review on process mining in business management. Technol. Forecast. Soc. Chang. **172**, 121021 (2021). https://doi.org/10.1016/j.techfore.2021.121021

Conversational Process Modelling: State of the Art, Applications, and Implications in Practice

Nataliia Klievtsova[1], Janik-Vasily Benzin[2], Timotheus Kampik[3], Juergen Mangler[2], and Stefanie Rinderle-Ma[2(✉)]

[1] Austrian Center for Digital Production, Vienna, Austria
nataliia.klievtsova@acdp.at
[2] TUM School of Computation, Information and Technology, Technical University of Munich, Garching, Germany
{janik.benzin,juergen.mangler,stefanie.rinderle-ma}@tum.de
[3] SAP Signavio, Berlin, Germany
timotheus.kampik@sap.com

Abstract. Chatbots such as ChatGPT have caused tremendous hype lately. For BPM applications, it is often not clear how to apply chatbots to generate business value. Hence, this work aims at the systematic analysis of existing chatbots for their support of conversational process modelling as a process-oriented capability. Application scenarios are identified along the process life cycle. Then a systematic literature review on conversational process modelling is performed. The resulting taxonomy serves as input for the identification of application scenarios for conversational process modelling, including paraphrasing and improvement of process descriptions. The application scenarios are evaluated for existing chatbots based on a real-world test set from the higher education domain. It contains process descriptions as well as corresponding process models, together with an assessment of the model quality. Based on the literature and application scenario analyses, recommendations for the usage (practical implications) and further development (research directions) of conversational process modelling are derived.

Keywords: Conversational process modelling · Chatbots · Process Descriptions · Process Models

1 Introduction

AI-powered chatbots *"have a considerable impact in many domains directly related to the design, operation, and application of information systems"* and at the same time need to be handled with care [70], as models provide you with information without considering their own technology's limitations. Business process management as an information systems discipline seems a viable candidate to benefit from chatbots and hence from the recent advances in large

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 319–336, 2023.
https://doi.org/10.1007/978-3-031-41623-1_19

language models, in particular, when supporting users in creating and improving process-related content, most prominently process models and process descriptions. Process models enable participants to understand the processes in which they are involved [17] and to improve business performance [6]. However, errors in the process models may have adverse business consequences [24], and may lead to problems during process execution and quality issues [15].

Currently the creation of process models is often based on the interaction between domain experts having the knowledge of the process and process modellers/analysts capable of process modelling and analysis techniques. Hence, the acquisition of as-is models can consume up to 60% of the time spent on process management projects [29]. The overarching question of this work is thus how and to which degree chatbots can replace the process modeller/analyst when creating process models through **conversational modelling (CM)** with the domain expert.

CM means conversation flow modelling where the chatbot can receive and interpret inputs from the user (i.e., follow-up questions, unexpected inputs, or changes of topic) and provide appropriate responses that keep the conversation coherent [49].

This question can be broken down into the following research questions:
RQ1 How can CM methods/tools be employed for process modelling?
RQ2 Which CM methods/tools exist for process modelling?
RQ3 How can we evaluate CM methods/tools with respect to process modelling?
RQ4 Which implications do Chatbots have for BPM modelling practice/research?

RQ1 – RQ4 are tackled as follows: Based on the concept of conversational process modelling, initial application scenarios are posed based on the process life cycle (cf. Sect. 2). These initial application scenarios provide the keywords for the subsequent literature review (cf. Sect. 3) which aims at refining the scenarios along a taxonomy of existing approaches. For evaluating existing chatbots, a test set of process descriptions, process models, and quality assessment is collected and prepared (cf. Sect. 4.1). The systematic analysis of the chatbots (cf. Sect. 4.2) along with the refined application scenarios are conducted based on key performance indicators and provide the basis for deriving practical implications and research directions in conversational process modelling (cf. Sect. 5).

2 Conversational Process Modelling

Only few papers address conversational modelling, mostly by focusing on the design of virtual human agents (aka chatbots), e.g., [49,61]. However, there is no common understanding of conversational **process** modelling yet and we hence provide informal Concept 1 which takes up characteristics of conversational modelling regarding the participants in the conversation, i.e., the domain expert and the chatbot, and the iterative nature of the conversation.

Concept 1. (Conversational process modelling) *describes the process of creating and improving process models and process descriptions based on the iterative exchange of questions/answers between domain experts and chatbots.*

Concept 1 reflects the overarching goal of conversational process modelling, i.e., to enable process modelling and improvement based on interaction between the domain expert and the chatbot, instead of interaction between the domain expert and the process analyst/modeller. This goal constitutes the first pillar to analyze the BPM life cycle w.r.t the process modelling scenarios where conversational process modelling can be applied. The second pillar reflects the assumption that conversational process modelling is exclusively based on domain expert/chatbot interaction and does not employ any other tool. In the conclusion, we will sketch how conversational process modelling can be extended if the chatbot usage is augmented by other tools such as process simulation tools.

In the following, Concept 1 is fleshed out for application scenarios along the BPM life cycle as provided in [27]. The BPM life cycle is chosen as it provides a systematic structuring of the different process-oriented tasks and capabilities towards creating business value.

Process discovery subsumes a range of methods to create process models (and is not to be confused with process discovery as the process mining task is necessarily based on event logs). The typical input in a process discovery project consists of textual process descriptions gathered based on interviews or workshops. Based on the process descriptions, process models are created by process modellers/analysts. We identified the following steps as suitable for being supported by chatbots: (1) gathering the process descriptions for creating the process model. This also includes the preparation of the process descriptions, i.e., to increase the quality of the process description in terms of, for example, being precise, e.g. through automatic paraphrasing. (2) taking a process description as an input and producing a process model (accompanied by the process description). Here, the chatbot can be employed for analyzing the text and extracting process model relevant information such as activities and their relations as well as actors [12]. Finally (3) assessing a process model (with the accompanying process description), regarding model quality based on quality metrics such as cohesion [72] and guidelines such as number of elements or label style [8].

The **process analysis** phase builds the bridge between the as-is process model created in the process discovery phase and the to-be model created in the process redesign phase. It is concerned with the qualitative and quantitative assessment of a process models. A qualitative analysis comprises, for example, an assessment whether or not certain activities can be automated; this can then be analogously reflected by an action recommendation, e.g., if the automation potential is not fully exploited, yet. The chatbot can support this assessment based on the extracted activities in the process discovery phase. The results of the qualitative assessment can then be used in the process redesign phase for corresponding redesign actions. Quantitative process analysis comprises, for example, detecting bottlenecks based on process simulations. As mentioned before, for this work, we assume that the chatbot is used without invoking further tools and systems such as a process simulator. Hence, quantitative process analysis does not include tasks for conversational process modelling at this stage, but for future work as discussed in Sect. 4.3.

Process redesign comprises the definition of the redesign goal which again is considered a managerial task. The chatbot can support the domain expert by proposing existing redesign methods such as Lean Six Sigma, as well as in querying models (cf. [56]) or applying the redesign instructions. Especially important is refactoring of process descriptions, based on existing guidelines on process model refactoring or catalogues of process smells such as [73].

The phases of **process implementation** and **process monitoring** are considered as a part of future work of conversational process modelling as they will require the invocation of additional tools and systems such as a process engine or process-aware information system.

Table 1 summarizes the initial application scenarios for conversational process modelling along the process life cycle phases and steps which constitute the input for the subsequent literature and test set based analyses.

Table 1. Application Scenarios and Chatbot Tasks along Process Life Cycle

# application	input	output	chatbot task
1. gather information	process description	process description	paraphrase
2. process modelling	process description	process model, process description	extract
3. assure model quality	process model, process description, process modelling guidelines and metrics	quality issues, refined process model, refined process description	compare and assess
4. select redesign method	collection of process models and process descriptions	redesign method, selection of process models and process descriptions	select method, query models
5. apply redesign method	collection of process models and process descriptions, redesign method	collection of process models and process descriptions	query and refactor models

The BPMN model depicted in Fig. 1 assembles and refines the application scenarios, together with their input, outputs, and related chatbot tasks as summarized in Table 1 into a generic process model for conversational process modelling, reflecting its interactive and iterative characteristics: at first, the domain expert provides a process description which is refined (→ paraphrase) and the results are displayed (→ extract). Then an assessment of the result quality is conducted (→ compare and assess). If the quality is insufficient, the process models/descriptions are refined (→ query, refactor), possibly based on a specific method (→ select method), until the quality reaches a sufficient level.

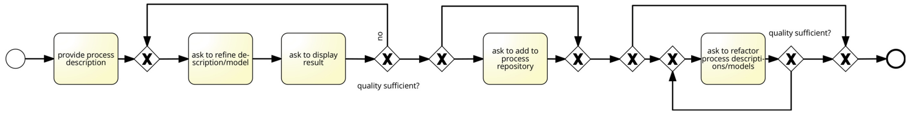


Fig. 1. The Process of Conversational Process Modeling (modeled in BPMN using SAP Signavio)

3 State of the Art

The literature analysis consists of two steps, i.e., i) a pre-review based on the initial application scenarios and life cycle phases summarized in Table 1 and based on the outcome of the pre-review, ii) a more generalized review including, for example, NLP-based methods for the extraction of model information from process descriptions. i) and ii) follow the guiding principles of [37].

i) Pre-review: The pre-review is conducted based on the keywords resulting from building the cross product of the application scenarios and keyword "chatbot" summarized in Table 1, e.g., `''process modelling'' chatbot`. These keywords are then used in the title search (allintitle) on google.scholar.com[1]. Next, we use the keywords resulting from the cross product of application scenario and chatbot task, e.g., `''process modelling'' paraphrase` and the keywords resulting from the cross product of keyword "conversational" and the application scenarios (allintitle), e.g., `conversational ''process modelling''`. In order to broaden the pre-review, we repeated the search for application and chatbot, but without keyword "process". Most of these searches result in 0 or a couple of hits, which were rejected due to quality issues or domain irrelevance.

The pre-review did not yield deeper insights into techniques, opportunities, and limitations of conversational process modelling. The results rather point towards generalizing the keywords used for the search, particularly covering NLP-based methods. Hence, for the **ii) second search**, we used https://scholar.google.com to produce Table 2. It shows the list of 52 papers relevant for a wide variety of relevant topics. Selection of the papers for the list was done based on the existence of the enumerated keywords (Selection Criteria) in the abstract or the title (for the first 20 hits).

In the following, we will discuss the literature collected in Table 2 regarding five fundamental questions that partly correspond to the research questions and partly to the pointers derived from the pre-review.

How do chatbots work, and what are important areas of application? A chatbot is a type of human-computer interaction, used to simulate conversations to solve particular user problems [3]. Chatbots work by processing language input from humans (furthermore referred to as natural language processing (NLP) [21, 50]), and reacting to it. The interpretation of human input is achieved through a set of rules [20, 26, 40], or by utilizing large language models (LLMs) [42], which

[1] last accessed 2023-03-23 and 2023-03-26 respectively.

are trained to understand the meaning/intent/context [18,44] and generate new content based on different statistical and probabilistic techniques. According to [51] the main areas of chatbot application are human resources, e-commerce, learning management systems, customer service, and sales.

How are responses generated? After receiving user input, the chatbot processes it into a machine-readable form and based on that input generates a natural language output utilizing different types of response generation methods [77]. Chatbot systems can be divided into six categories, based on the type of response generator [44]. (1) template-based: response is selected from the list of predefined pairs of query patterns; (2) corpus-based: converts user query to a structured query language (SQL) query and passes it to utilized techniques of professional knowledge management (i.e., database, ontology); (3) intent-based: task-oriented system, which based on user query tries to recognise user intent with the help of advanced NLU techniques; (4) RNN-based: RNN-based (Recurrent Neural Network) chatbot generates response query directly from the user query with the help of the model, trained on dialogue data set; (5) RL-based: RL-based (Reinforcement Learning) chatbots use rewarding and punishing functions to achieve the desired behaviour; (6) hybrid-based: a combination of approaches listed above to achieve better performance or to overcome limitations, faced by using one approach only.

How can response generation be implemented? All of the above types utilize some type of knowledge graph to formalize the configuration [7,76] and the intended output format of the conversation [4,55]. The knowledge graph is either accessed by simple querying languages such as AIML or SPARQL, or it is encoded as part of a neural network through training. So responses are either queried explicitly or generated implicitly as part of a neural network. Both approaches have different strengths and weaknesses. For conversation-related applications such as entertainment, neural networks work well, but for other applications with special output, other approaches are still valid solutions. Low-code solutions to control explicit responses [25] as well as BPMN-based solutions to encode potential progressions of a conversation [60] have been proposed. One example of such a system is PACA [41]. Automatically learning from user interactions can be achieved not only for neural networks (e.g., reinforcement learning) but also by encoding interactions automatically into rules, such as in [5,36].

Can chatbots deal with business processes? According to the survey of chatbot integration [9], 2 out of 347 chatbot systems support the business process interface pattern, i.e., [34,43] that convert BPMN process models into dialog models/chatbots. Currently, there are no chatbots that are able to generate BPMN models themselves. However, interest in the generation of models from various types of document sources has recently increased [29,31,64]. Referring to [32] as an input for business process model generation use case diagrams, business rules, standard operating procedures, and plain unstructured text are considered. Based on the approaches mentioned above, the following 3 steps for creating BPMN can be summarized [12,66]: (1) Sentence Level Analysis:

Table 2. Literature Queries, Hits, and Selections

Query (allintitle:)	Hits	Selection Criteria	#	List
chatbot technology overview	1		1	[3]
Natural language processing	10400	automated NLP	2	[21,50]
nlp Chatbot Development	7	deep learning	1	[59]
chatbots business processes	2	capability to learn	1	[36]
Chatbot integration	32	chatbot integration	1	[9]
quark chatbot	1		1	[34]
((Chatbots) OR (chatbot)) Process Models	2	process model	1	[43]
reasoning processes descriptions	3		1	[67]
”process model generation”	15	text	1	[29]
generating BPMN diagram	2	text	1	[64]
business process (model) OR (models) generating	34	Natural Language,document sources	2	[31,32]
extracting business process language models	2	NLP, language model	2	[12,66]
AI based language models	2	NLP, LMs	1	[42]
large language models	628	NLP, BPMN	3	[52,75] , [38]
BOMN generation	22	NLP, LMs	1	[48]
“process extraction” from text	6	text, textual information	1	[10,11,13]
“knowledge graphs” chatbots	5	NLP, LMs	1	[4,7,54,55,76]
chatbots BPMN modelling	0	—	—	—
chatbots graph generation	0	—	—	—
((model based) OR (model-based))	12	NLP, BPMN, UML	1	[28]
generate graphs chatbots	0	—	—	—
generate graphs plain text	0	—	—	—
BPMN modelling chatbots	0	—	—	—
low-code chatbot development	1		1	[25]
generating texts models	2	process model	1	[39]
declarative process model generation	0	—	—	—
process models chatbot	1	—	1	[5]
process conversational agents	7	BPMN	2	[41,60]
rule based chatbots	5	natural language, AIML	3	[20,26,40]
chatbot designs	4	natural language	2	[18,44]
Process Models Chatbots	1	—	1	[43]
mining models from text	11	process model	1	[45]
automatic generation bpmn	5	from BPMN, process model	3	[14,23,62]
text information extraction	539	unstructured text, semi-structured text	7	[22,33,53,57,58,68, 69]
text data augmentation methods	8	methodology	1	[79]
data augmentation approaches nlp	1		1	[2]
easy data augmentation techniques	4	data augmentation	3	[30,63,74]
automatic machine translation paraphrasing	3	paraphrasing	2	[71,78]
paraphrasing automatic evaluation	7	bleu, english	2	[16,35,78]

extraction of basic BPMN artefacts such as tasks, events, and actors; (2) Text Level Analysis: exploration of relationships between basic items, e.g., gateways. (3) Process Model Generation: create a syntactically correct model, that captures the semantics of the input. [67] proposes a machine-readable intermediate format generated out of natural language (either through automatic or manual annotation). The result is then easy to interpret by computers.

How can we evaluate chatbots with respect to BPM modelling? Currently there are no gold standard data sets that can be used to evaluate and compare the efficiency of process extraction from unstructured text [10]. In [29] a set of 47 text-model pairs from industry and textbooks are introduced, which could be converted with an accuracy of 77% (up to 96% of similarity for some cases) from text to model. In [39], 53 model-text pairs were used to evaluate performance of a novel model-to-text transformation method. To avoid the necessity of constant creation of new datasets by hand, data augmentation techniques (increase of the training set size with the help of the modified copies of already existing data set items) can be used [2,79]. Another important tool is paraphrasing [35], which is about generating similar texts from a source. Such texts are generally recognized as lexically and syntactically different while remaining semantically equal.

4 Performance of Current Generation LLMs for Conversational Process Modelling

In order to assess the performance of conversational process modelling tools and answer RQ3, it is necessary to come up with a data set, an evaluation method, and a set of KPIs. Extending the three steps, which are required to create a BPMN model (see Sect. 3), a fully integrated conversational process modelling toolchain would contain: (a) extraction of tasks from textual descriptions, (b) extraction of logic such as decisions or parallel branchings from textual descriptions, (c) creation and the layout of a BPMN model, and (d) the application of modifications for refinement of BPMN models. As a fully integrated conversational process modelling tool does not exist yet, in this paper we concentrate on how well current LLMs, namely GPT models text-davinci-001 (GPT1), text-davinci-002 (GPT2), text-davinci-003 (GPT3) from openai.org playground[2], as well gpt 3.5 turbo (GPT3.5) from writesonic.com[3], perform for extracting tasks for textual description (see (a) above). Task extraction is a starting point of the conversational process modelling toolchain, as the task is an atomic element of the process flow, which represents a unit of work that should be performed [65].

4.1 Test Set Generation

The test set [46] utilized in this paper contains 21 textual process descriptions from 6 topics or domains. For each process description between 8 and 11 BPMN

[2] last access: 2023-03-29.
[3] last access: 2023-03-29.

process models have been created by modelling novices. These models represent different possible ways of interpreting the textual process description. Each model has at least one start and end event, 3 exclusive gateways, one parallel gateway, and an average of 14 tasks. Some models also contain sub-processes, pools, and lanes. Each model was evaluated by a modelling expert using a quality value from 0 to 5, to reflect, on how well the textual description has been transformed into a BPMN model, i.e., all tasks and decisions from the textual description are in the BPMN, tasks which can run in parallel have been correctly identified, and the BPMN model is well-formed.

An example of a textual description and an associated interpretation, i.e., the BPMN model, can be seen in Fig. 2.

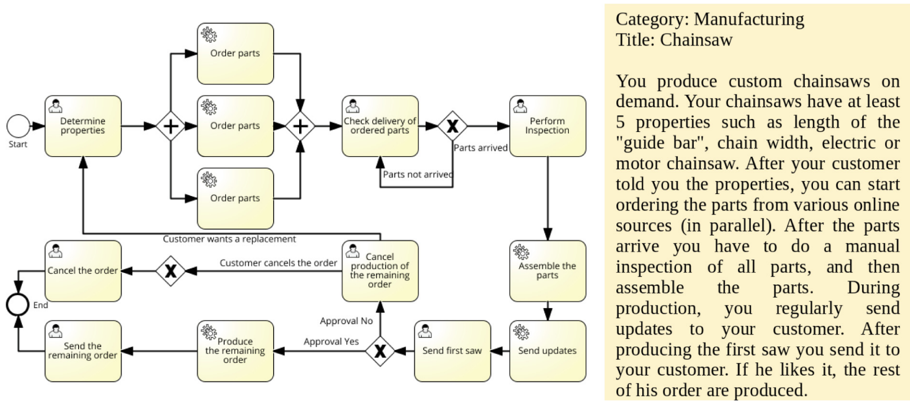


Fig. 2. Textual Description And BPMN Model From the Evaluation Dataset

4.2 Evaluation

In this section, we will use the following KPIs and discuss their impact on conversational process modelling approaches: **KPI1** - Text Similarity; **KPI2** - Set Similarity; **KPI3** - Set Overlap; **KPI4** - Restricted Text Similarity; **KPI5** - Restricted Set Similarity; **KPI6** - Restricted Set Overlap; **KPI7** - Average Augmented Task Extraction Prevalence and Similarity (GPT3 only). All results, including non-averaged data, are also available in [47].

Prompt engineering and KPIs: KPIs 1–3 are used to assess task extraction from original process descriptions. This is realized by passing the following prompt "Considering following $< process\_description >$ return the list of main tasks in it" to the LLMs. For assessment based on KPIs 4–6, the original prompt is changed to the "Considering following $< process\_description >$ return the list of main tasks (each 3–5 words) in it" to improve the granularity of extracted tasks and to refine the quality of obtained tasks' labeling. KPI7 is used to evaluate how stable task extraction is, performed by LLMs by extending the set of original process descriptions by utilizing different paraphrasing algorithms.

Task extraction from associated models is realised by parsing .XML documents and extracting relevant BPMN activities, keeping their sequence in the process flow.

As the basis for each similarity measurement we utilize contextual (BERT) and non-contextual (TD-IDF) vectorisers with a cosine similarity metric [19]. The contextual and non-contextual approaches will be denoted as C and NC.

For **KPI1**, each LLM (GPT1, GPT2, GPT3, GPT3.5) is instructed to extract the tasks from the original process descriptions. The answer is then compared to the original text to assess the completeness of the extraction. The results are depicted in Table 3. For this KPI, GPT3.5 is the most successful LLM.

Table 3. Text Similarity (**KPI1**): Comparison of tasks extracted by LLM and original text using contextual (BERT) and non-contextual (TD-IDF) vectorisers

Method	gpt1	gpt2	gpt3	gpt3.5
non-contextual	0.46	0.65	0.60	0.63
contextual	0.76	0.80	0.78	0.84

Table 4 shows the results for **KPI2**. The four LLMs are instructed to extract tasks from each textual description. This set of tasks is then compared to the set of tasks, extracted from each BPMN model mentioned above (see Sect. 4.1). As for every textual description multiple BPMN models exist, the results are averaged per textual description. The averages are then again averaged for all textual descriptions. GPT3 is successful for this KPI with 74% extraction rate.

Table 4. Set Similarity (**KPI2**): Comparison of tasks extracted by LLM with tasks extracted from BPMN Models. For each text a set of n tasks is extracted. Each text has 8–11 associated models from which again a set m of tasks can be extracted. Each set n is compared with all sets m, yielding a set of similarities which is averaged for similarity methods contextual (C) and non-contextual (NC)

LLM	C	NC	avg. # of tasks extracted from texts	avg. # of tasks extracted from models
gpt 1	0.72	0.32	7.6	12
gpt 2	0.71	0.32	6.7	12
gpt 3	0.74	0.35	7.7	12
gpt 3.5	0.73	0.36	8.5	12

For **KPI3**, the goal is to quantify the overlap between extracted tasks from the original text and associated to its models: (1) how similar are individual tasks, and (2) how many tasks exist only in one of the two extractions. The results are shown in Table 5 and show that between 6 and 7 tasks extracted from the models are also found in the text, while about 6 tasks could not be found in the extracted text. When looking at it from the point of view of the

Table 5. Set Overlap (**KPI3**): Each task extracted from the text is compared (for each associated model) with task extracted from the model. If the similarity is bigger than a threshold, a task is deemed common, else it is deemed to only occur in either the model or the text.

LLM	similarity	common model	common chat	only in model	only in chat
gpt 1	C	6.6	4.5	5.2	3.2
gpt 1	NC	5.9	4	5.9	3.6
gpt 2	C	6.2	4.1	5.6	2.6
gpt 2	NC	5.6	3.6	6.2	3
gpt 3	C	6.7	4.6	5.1	3
gpt 3	NC	6.7	4.6	5.1	3
gpt 3.5	C	7	4.7	4.9	3.8
gpt 3.5	NC	6.5	4.4	5.4	4.1

tasks extracted from the text, the ratio becomes 4:3. So almost 50% of the tasks are not similar between the model and text (see discussion for details).

KPI4 focuses on restricting the number of words per extracted task, to coax the bot into extracting more tasks, as generally, the number of extracted tasks from the text is lower than the number of tasks contained in the models (see discussion for more details). Table 6 shows that this decreases the similarity when comparing text (due to stronger paraphrasing), but **KPI5** (cf. Table 7) and **KPI6** (cf. Table 8) show an increase in the number of tasks by one while not decreasing similarity when compared to the tasks from the model.

Table 6. Restricted Text Similarity (**KPI4**): Task names are allowed to only have 3–5 words, cmp. Table 3.

method	gpt1	gpt2	gpt3	gpt3.5
non-contextual	0.24	0.47	0.38	0.27
contextual	0.70	0.77	0.73	0.73

Table 7. Restricted Set Similarity (**KPI5**): Task names are allowed to only have 3–5 words, cmp. Table 4.

LLM	C	NC	avg. # of tasks extracted from texts	avg. # of tasks extracted from models
gpt 1	0.73	0.32	7.6	12
gpt 2	0.74	0.33	7.7	12
gpt 3	0.73	0.32	8.3	12
gpt 3.5	0.75	0.30	8.5	12

Finally, for **KPI7**, we assessed the effects of paraphrasing on prevalence and similarity, i.e., how stable LLMs are for task extraction with similar input. We use nine different algorithms for paraphrasing text [2] (rewriting sentences using synonyms), which is, for example, useful to clean up textual descriptions from humans. The results are displayed in Table 9, and show that especially the contextual similarity does not decrease significantly, while the number of extracted tasks even improves in comparison to the original text.

Table 8. Restricted Set Overlap (**KPI6**): Task names are allowed to only have 3-5 words, cmp. Table 5.

LLM	similarity	common model	common chat	only in model	only in chat
gpt 1	NC	6	4	5.7	3.5
gpt 2	NC	6.4	4.2	5.4	3.5
gpt 3	NC	7	4.7	4.8	3.5
gpt 3.5	NC	6.8	4.6	5	3.8

Table 9. Average Augmented Task Extraction Prevalence and Similarity GPT3 (**KPI7**): for nine different paraphrasing methods, the average number of tasks, and similarity measures are calculated. The second row holds the value of the original text from Table 6

	Original	SR	DL	SW	IN	NLPaug	TDE	TRU	TES	EMB
avg. # tasks	8.25	8.10	8.43	7.48	8.19	8.10	7.57	7.86	8.62	8.29
C similarity	0.73	0.69	0.69	0.68	0.70	0.70	0.70	0.67	0.70	0.70
NC similarity	0.38	0.20	0.22	0.25	0.21	0.21	0.21	0.19	0.21	0.22

4.3 Discussion

Tables 3, 4, 5, 6, 7, 8 and 9 clearly show that GTP3 currently supports task extraction the best, beating GPT3.5. The potential reason for GPT3 success could be the size of the model (175 billion parameters over 1,3 billion for GPT3.5). GPT3.5 model is optimized for a chat and may not be as effective for more complex language tasks [1].

Another important insight is that manually designed and refined models contain additional tasks that cannot be directly extracted from the original text but exist due to a humans ability to "read between the lines" or reason about task granularity. GPT extracts tasks exactly as written in text but does not have the capability to reason when it makes more sense to have multiple small tasks instead of a big one. We tried to coax GPT3 into extracting more tasks by restricting the number of words describing a task (i.e., its label), which increased the average number of extracted tasks slightly by 1, as can be seen in Table 7.

On average, GPT extracted a third less tasks than existed in the model. When strictly looking at the capability of extracting tasks from the original text, GPT3, on average, achieves a text similarity of 80%. The interpretation of this value is difficult. It could mean that the LLM missed about 20% of the tasks or, alternatively, that 20% of the text are just the filler words that have been ignored by the LLM. Together with the observation that the LLM does not like to split up tasks, the 30% less tasks extracted from the text in comparison to the model, hint at a possible explanation.

5 Conclusion: Practical Implications and Research Directions

From the state-of-the-art discussion in Sect. 3 and the results of the evaluation presented in Sect. 4.3, the following two main managerial implications can be derived:

1. For the chatbot application scenarios “gather information” and “process modelling” (cf. Table 1), chatbots are in principle ready to be applied in practice as-is, yet the results have to be taken with a grain of salt, i.e., the domain expert should always check the results. However, the lack of an appropriate, human-readable output format, e.g., a BPMN process model, limits the space of early adopters in a company significantly to experts at the intersection of their domain and computer science. This limitation is particularly unfortunate, as it counteracts the goal of conversational process modelling to minimize the necessary technical skills of the domain expert.
2. For the chatbot application scenarios “compare and assess”, “select method, query models”, and “query and refactor models”, off-the-shelf chatbots are not yet ready to be applied due to their inability to output process models and to understand process model semantics.

As business process modelling has become an important tool for managing organizational change and for capturing requirements of software, the first managerial implication is that conversational process modelling can already have a significant business impact. Considering that the central problem in this area – the acquisition of as-is models – consumes up to 60% of the time spent on process management projects [29], chatbot-based partial automation can be sufficiently impactful, even if substantial human refinement is required.

The second managerial implication is that future research should focus on integrating the strong language capabilities of chatbots with the specialized capabilities of existing knowledge-based tools. The integrative research direction is more promising than chatbot training with specialized process modeling training sets featuring native process models, e.g., process models in BPMN format and a number of semantic targets, such as information on the existence of deadlocks in a process model. First, training of the chatbot with respect to business process models ignores the vast existing modeling knowledge encoded into existing tools. Second, semantics are clearly defined and encoded in existing tools such that training chatbots with the aim of understanding formal semantics is futile unless it serves as an intermediate step that unlocks further value.

To conclude, while advanced tasks such as model querying, refinement, and analysis presumably require domain-specific solutions, the interplay of traditional, knowledge based approaches to business process modeling can relatively straight-forwardly be augmented by machine learning-based chatbots to facilitate tedious tasks such as information gathering and basic model creation.

References

1. Openai documentation: models overview. https://platform.openai.com/docs/models/
2. Data augmentation approaches in natural language processing: a survey. AI Open **3**, 71–90 (2022). https://doi.org/10.1016/j.aiopen.2022.03.001
3. Adamopoulou, E., Moussiades, L.: An overview of chatbot technology. In: Artificial Intelligence Applications and Innovations, pp. 373–383 (2020)
4. Ait-Mlouk, A., Jiang, L.: KBot: a knowledge graph based chatbot for natural language understanding over linked data. IEEE Access **8**, 149220–149230 (2020). https://doi.org/10.1109/ACCESS.2020.3016142
5. Alman, A., Balder, K.J., Maggi, F.M., van der Aa, H.: Declo: a chatbot for user-friendly specification of declarative process models. In: Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2020, vol. 2673, pp. 122–126. CEUR-WS.org (2020). https://ceur-ws.org/Vol-2673/paperDR12.pdf
6. Alotaibi, Y.: Business process modelling challenges and solutions: a literature review. J. Intell. Manuf. **27**(4), 701–723 (August 2016). https://doi.org/10.1007/s10845-014-0917-4,https://ideas.repec.org/a/spr/joinma/v27y2016i4d10.1007_s10845-014-0917-4.html
7. Avila, C.V.S., Franco, W., Maia, J.G.R., Vidal, V.M.P.: CONQUEST: a framework for building template-based IQA chatbots for enterprise knowledge graphs. In: Métais, E., Meziane, F., Horacek, H., Cimiano, P. (eds.) NLDB 2020. LNCS, vol. 12089, pp. 60–72. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51310-8_6
8. Avila, D.T., dos Santos, R.I., Mendling, J., Thom, L.H.: A systematic literature review of process modeling guidelines and their empirical support. Bus. Process. Manag. J. **27**(1), 1–23 (2021). https://doi.org/10.1108/BPMJ-10-2019-0407
9. Baez, M., Daniel, F., Casati, F., Benatallah, B.: Chatbot integration in few patterns. IEEE Internet Comput. **25**(03), 52–59 (2021). https://doi.org/10.1109/MIC.2020.3024605
10. 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)
11. Bellan, P., Dragoni, M., Ghidini, C.: Process extraction from text: state of the art and challenges for the future. arXiv preprint arXiv:2110.03754 (2021)
12. Bellan, P., Dragoni, M., Ghidini, C.: Extracting business process entities and relations from text using pre-trained language models and in-context learning, pp. 182–199 (09 2022). https://doi.org/10.1007/978-3-031-17604-3_11
13. Bellan, P., Ghidini, C., Dragoni, M., Ponzetto, S.P., van der Aa, H.: Process extraction from natural language text: the pet dataset and annotation guidelines. In: Workshop on Natural Language for Artificial Intelligence (2022)
14. Belo, O., Gomes, C., Oliveira, B., Marques, R., Santos, V.: Automatic generation of ETL physical systems from BPMN conceptual models. In: Model and Data Engineering, pp. 239–247 (2015)
15. de Brito Dias, C.L., Dani, V.S., Mendling, J., Thom, L.H.: Anti-patterns for process modeling problems: an analysis of BPMN 2.0-based tools behavior. In: Business Process Management Workshops (2019)
16. Callison-Burch, C., Cohn, T., Lapata, M.: Parametric: an automatic evaluation metric for paraphrasing. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 97–104 (2008)

17. de Camargo, J.V., et al.: A complementary analysis of the behavior of BPMN tools regarding process modeling problems. In: Augusto, A., Gill, A., Bork, D., Nurcan, S., Reinhartz-Berger, I., Schmidt, R. (eds.) Enterprise, Business-Process and Information Systems Modeling, pp. 43–59. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-07475-2_4
18. Cañizares, P.C., Pérez-Soler, S., Guerra, E., de Lara, J.: Automating the measurement of heterogeneous chatbot designs. Appl. Comput., 1491–1498 (2022)
19. Chandrasekaran, D., Mago, V.: Evolution of semantic similarity-a survey. ACM Comput. Surv. **54**(2), 41:1–41:37 (2021). https://doi.org/10.1145/3440755
20. Choa, N., Limb, Y., Limc, J.: Research design to compare the impacts of two different types of chatbots on mobile shopping behavior: rule-based and natural language processing-based. Editorial Board, p. 43
21. Chowdhary, K.: Fundamentals of Artificial Intelligence. Springer India (2020). https://doi.org/10.1007/978-81-322-3972-7, https://books.google.at/books?id=8SfbDwAAQBAJ
22. Ciravegna, D., et al.: Adaptive information extraction from text by rule induction and generalisation (2001)
23. Cossentino, M., Lopes, S., Sabatucci, L.: A tool for the automatic generation of MOISE organisations from BPMN. In: WOA, vol. 1613, p. 69 (2020)
24. Dani, V.S., Freitas, C.M.D.S., Thom, L.H.: Recommendations for visual feedback about problems within BPMN process models. Softw. Syst. Model. **21**(5), 2039–2065 (2022). https://doi.org/10.1007/s10270-021-00972-0
25. Daniel, G., Cabot, J., Deruelle, L., Derras, M.: Xatkit: a multimodal low-code chatbot development framework. IEEE Access **8**, 15332–15346 (2020). https://doi.org/10.1109/ACCESS.2020.2966919
26. Dihyat, M.M.H., Hough, J.: Can rule-based chatbots outperform neural models without pre-training in small data situations: a preliminary comparison of AIML and Seq2Seq. In: Workshop Semantics Pragmatics Dialogue, pp. 1–3 (2021)
27. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-33143-5
28. Ed-douibi, H., Cánovas Izquierdo, J.L., Daniel, G., Cabot, J.: A model-based chatbot generation approach to converse with open data sources. In: Web Engineering, pp. 440–455 (2021)
29. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: Advanced Information Systems Engineering. pp. 482–496 (2011)
30. Fu, K., Lin, J., Ke, D., Xie, Y., Zhang, J., Lin, B.: A full text-dependent end to end mispronunciation detection and diagnosis with easy data augmentation techniques. arXiv preprint arXiv:2104.08428 (2021)
31. 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
32. Indahyanti, U., Djunaidy, A., Siahaan, D.: Auto-generating business process model from heterogeneous documents: a comprehensive literature survey. In: Electrical Engineering, Computer Science and Informatics, pp. 239–243 (2022). https://doi.org/10.23919/EECSI56542.2022.9946460
33. Jiang, J.: Information extraction from text. Mining Text Data, pp. 11–41 (2012)
34. Kalia, A.K., Telang, P.R., Xiao, J., Vukovic, M.: Quark: a methodology to transform people-driven processes to chatbot services. In: Service-Oriented Computing, pp. 53–61 (2017)

35. Kauchak, D., Barzilay, R.: Paraphrasing for automatic evaluation. In: Human Language Technology Conference of the NAACL, pp. 455–462 (2006)
36. Kecht, C., Egger, A., Kratsch, W., Röglinger, M.: Quantifying chatbots' ability to learn business processes. Inf. Syst. **113**, 102176 (2023). https://doi.org/10.1016/j.is.2023.102176
37. Kitchenham, B.: Procedures for Performing Systematic Reviews. Keele University Technical Report TR/SE-0401, Keele University (2004)
38. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. arXiv preprint arXiv:2205.11916 (2022)
39. Leopold, H., Mendling, J., Polyvyanyy, A.: Generating natural language texts from business process models. In: Advanced Information Systems Engineering, pp. 64–79 (2012)
40. Lim, Y., Lim, J., Cho, N.: An experimental comparison of the usability of rule-based and natural language processing-based chatbots. Asia Pacific J. Inf. Syst. **30**(4), 832–846 (2020)
41. Lins, L.F., Melo, G., Oliveira, T., Alencar, P., Cowan, D.: PACAs: process-aware conversational agents. In: Business Process Management Workshops, pp. 312–318 (2022)
42. Liu, Z., Roberts, R.A., Lal-Nag, M., Chen, X., Huang, R., Tong, W.: AI-based language models powering drug discovery and development. Drug Discovery Today **26**(11), 2593–2607 (2021)
43. López, A., Sànchez-Ferreres, J., Carmona, J., Padró, L.: From process models to chatbots. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 383–398. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_24
44. Luo, B., Lau, R.Y.K., Li, C., Si, Y.W.: A critical review of state-of-the-art chatbot designs and applications. WIREs Data Min. Knowl. Discov. **12**(1), e1434 (2022). https://doi.org/10.1002/widm.1434
45. M. Riefer, S. Ternis, T.T.: Mining process models from natural language text: a state-of-the-art analysis. In: Wirtschaftsinformatik, pp. 9–11 (2016)
46. Mangler, J., Klievtsova, N.: Dataset: textual process descriptions and corresponding BPMN models (2023). https://doi.org/10.5281/zenodo.7783492
47. Mangler, J., Klievtsova, N.: Evaluation details: GPT capabilites for extracting tasks from textual process descriptions (2023). https://doi.org/10.5281/zenodo.8063211
48. Maqbool, B., et al.: A comprehensive investigation of BPMN models generation from textual requirements-techniques, tools and trends. In: Information Science and Applications, pp. 543–557 (2019)
49. McTear, M.F.: Conversational modelling for chatbots: current approaches and future directions (2018)
50. Meurers, D.: Natural language processing and language learning. Encyclopedia Appl. Linguist, 4193–4205 (2012)
51. Miklosik, A., Evans, N., Qureshi, A.M.A.: The use of chatbots in digital business transformation: A systematic literature review. IEEE Access **9**, 106530–106539 (2021). https://doi.org/10.1109/ACCESS.2021.3100885
52. Min, B., et al.: Recent advances in natural language processing via large pre-trained language models: a survey. arXiv preprint arXiv:2111.01243 (2021)
53. Mooney, R.J., Bunescu, R.: Mining knowledge from text using information extraction. ACM SIGKDD Explor. Newsl. **7**(1), 3–10 (2005)
54. Omar, R., Mangukiya, O., Kalnis, P., Mansour, E.: ChatGPT versus traditional question answering for knowledge graphs: current status and future directions towards knowledge graph chatbots. arXiv preprint arXiv:2302.06466 (2023)

55. Patsoulis, G., Promikyridis, R., Tambouris, E.: Integration of chatbots with Knowledge Graphs in eGovernment: the case of getting a passport. In: 25th Pan-Hellenic Conference on Informatics, pp. 425–429 (2021)
56. Polyvyanyy, A. (ed.): Process Querying Methods. Springer (2022). https://doi.org/10.1007/978-3-030-92875-9
57. Rahman, S., Kandogan, E.: Characterizing practices, limitations, and opportunities related to text information extraction workflows: a human-in-the-loop perspective. In: Human Factors in Computing Systems, pp. 1–15 (2022)
58. Rau, L.F., Jacobs, P.S., Zernik, U.: Information extraction and text summarization using linguistic knowledge acquisition. Inf. Process. Manage. **25**(4), 419–428 (1989)
59. Rawat, B., Bist, A.S., Rahardja, U., Aini, Q., Ayu Sanjaya, Y.P.: Recent deep learning based NLP techniques for chatbot development: an exhaustive survey. In: Cyber and IT Service Management, pp. 1–4 (2022). https://doi.org/10.1109/CITSM56380.2022.9935858
60. Rooein, D., Bianchini, D., Leotta, F., Mecella, M., Paolini, P., Pernici, B.: aCHAT-WF: generating conversational agents for teaching business process models. Softw. Syst. Model. **21**(3), 891–914 (2022). https://doi.org/10.1007/s10270-021-00925-7
61. Rossen, B., Lind, S., Lok, B.: Human-centered distributed conversational modeling: efficient modeling of robust virtual human conversations. In: Ruttkay, Z., Kipp, M., Nijholt, A., Vilhjálmsson, H.H. (eds.) IVA 2009. LNCS (LNAI), vol. 5773, pp. 474–481. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04380-2_52
62. Ryniak, C., Burgert, O.: Automatic generation of checklists from business process model and notation (BPMN) models for surgical assist systems. Curr. Dir. Biomed. Eng. **6**(1), 20200005 (2020). https://doi.org/10.1515/cdbme-2020-0005
63. Santing, L.: Easy data augmentation techniques for traditional machine learning models on text classification tasks. B.S. thesis (2021)
64. Sholiq, S., Sarno, R., Astuti, E.S.: Generating BPMN diagram from textual requirements. J. King Saud University - Comput. Inf. Sci. **34**(10), 10079–10093 (2022). https://doi.org/10.1016/j.jksuci.2022.10.007
65. Silver, B.: BPMN Method and Style: With BPMN Implementer's Guide. Cody-Cassidy Press (2011). https://books.google.at/books?id=mLDYygAACAAJ
66. Sintoris, K., Vergidis, K.: Extracting business process models using natural language processing (NLP) techniques. In: Business Informatics, vol. 01, pp. 135–139 (2017). https://doi.org/10.1109/CBI.2017.41
67. Sànchez-Ferreres, J., Burattin, A., Carmona, J., Montali, M., Padró, L.: Formal reasoning on natural language descriptions of processes. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 86–101. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_8
68. Soderland, S.: Learning information extraction rules for semi-structured and free text. Mach. Learn. **34**, 233–272 (1999)
69. Tang, B., Wu, Y., Jiang, M., Chen, Y., Denny, J.C., Xu, H.: A hybrid system for temporal information extraction from clinical text. J. Am. Med. Inform. Assoc. **20**(5), 828–835 (2013)
70. Teubner, T., Flath, C., Weinhardt, C.: Welcome to the era of chatGPT. Bus. Inf. Syst. Eng. (2023). https://doi.org/10.1007/s12599-023-00795-x
71. Thompson, B., Post, M.: Automatic machine translation evaluation in many languages via zero-shot paraphrasing. arXiv preprint arXiv:2004.14564 (2020)
72. Vanderfeesten, I.T.P., Reijers, H.A., van der Aalst, W.M.P.: Evaluating workflow process designs using cohesion and coupling metrics. Comput. Ind. **59**(5), 420–437 (2008). https://doi.org/10.1016/j.compind.2007.12.007

73. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring large process model repositories. Comput. Ind. **62**(5), 467–486 (2011). https://doi.org/10.1016/j.compind.2010.12.012
74. Wei, J., Zou, K.: EDA: easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
75. Witteveen, S., Andrews, M.: Paraphrasing with large language models. arXiv preprint arXiv:1911.09661 (2019)
76. Wittig, A., Perevalov, A., Both, A.: Towards bridging the gap between knowledge graphs and chatbots. In: Web Engineering, pp. 315–322 (2022)
77. Meyer von Wolff, R., Nörtemann, J., Hobert, S., Schumann, M.: Chatbots for the information acquisition at universities – a student's view on the application area. In: Følstad, A., et al. (eds.) CONVERSATIONS 2019. LNCS, vol. 11970, pp. 231–244. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-39540-7_16
78. Yoshimura, R., Shimanaka, H., Matsumura, Y., Yamagishi, H., Komachi, M.: Filtering pseudo-references by paraphrasing for automatic evaluation of machine translation. In: Machine Translation, pp. 521–525 (2019)
79. Yu, J., Choi, J., Lee, Y.: Mixing approach for text data augmentation based on an ensemble of explainable artificial intelligence methods. Neural Process. Lett. 1–17 (2022). https://doi.org/10.1007/s11063-022-10961-z

Detecting Weasels at Work: A Theory-Driven Behavioural Process Mining Approach

Michael Leyer[1,2](✉), Arthur H.M. ter Hofstede[2], and Rehan Syed[2]

[1] Philipps-University of Marburg, Marburg, Germany
[2] Queensland University of Technology, Brisbane, Australia
michael.leyer@wiwi.uni-marburg.de

Abstract. Stresses and temptations in the workplace foster employee behaviour that is less than desirable. The term *weasel* has been used to describe employees that exhibit a variety of undesirable behaviour at work, including taking undeserved credit, performing below expectation, shirking work, and making co-workers look bad. While this behaviour has traditionally been hard to detect, contemporary systems record many of our work-related actions and the resulting event logs can be subjected to process mining analysis. In this paper we focus on detecting weasels through the evidence they leave behind in such event logs. We capture a variety of weasel behaviours in the form of patterns and suggest how process mining can be used to unearth this behaviour. The patterns are validated through a survey with relevant stakeholders.

Keywords: Principal-Agent theory · Process mining · Patterns · Behavioural deviation

1 Introduction

A major challenge for contemporary organisations is to be able to determine whether their staff are conducting work in line with their contract, which captures what they should contribute to the delivery of goods and services to customers [9]. How employees should perform their work to achieve the related organisational goals is (mostly) documented in the processes of an organization. But this does not mean that these processes are actually followed. Organizational reality may be very different. Human behaviour may deviate from how to conduct work, when to conduct it, and even what work to perform for a variety of reasons. Some popular examples are: different opinions on how to do a job, social interaction leading to favourite and non-favourite colleagues, individual differences regarding abilities, as well as opportunistic behaviour. Hence, it is a key task of managers to control and ensure that employees do what they are supposed to do according to the framework set by the official processes, especially to avoid that individual bad behaviour becomes an accepted norm [10]. This task

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 337–354, 2023.
https://doi.org/10.1007/978-3-031-41623-1_20

is difficult as it requires regular information on *how* employees conduct their work and recent developments such as working remotely have not made it easier as they lead to a reduction in workplace observations. One potential solution to this problem is the use of process mining [1] as it enables the analysis of process execution data to answer business related questions [4]. But while such data contains footprints of employee behaviour, a behavioural background is needed to guide algorithm development to unearth behavioural patterns.

Process mining research focusing on the resource perspective has not leveraged behavioural theories in order to uncover deviating behaviour. In order to be able to use such theories the field needs to move beyond conformance checking and descriptive performance measures in order to identify and understand specific types of behaviour. These theories can help us understand why certain behaviours occur and help capture patterns that can be used to unearth such behaviours in event logs.

This paper uses Principal-Agent Theory, a fundamental theory used to explain the relationship between principals and agents in organizational settings [15]. At the core of this theory is the assumption of opportunistic behaviours that agents will show in order to hide their true intentions and performance. Individuals showing such behaviour are sometimes referred to as *weasels* and their behaviour has been defined as "the propensity of employees to engage in unsanctioned, non-work related activities during work time" [8]. Using the underlying notion of opportunistic behaviour, we refer to any work-related activity that is not desirable from a performance perspective as weasel behaviour. On the basis of the theory, we develop a collection of patterns capturing a range of weasel behaviour and ways to identify such behaviour through the footprints that it leaves behind in event logs. The use of patterns to characterise problems and their solutions is well established in the field of business process management. Patterns can provide a first grasp on a fuzzy topic. They establish terminology and structure, and can serve as the basis for algorithm and tool development.

The main contribution of this paper is the conceptualisation of weasel behaviour and its manifestation in event logs through a collection of patterns based on Principal-Agent Theory. Each pattern describes a type of behaviour, one or more examples, why it occurs, how it can be detected, how it can be remedied, potential side-effects of remedies, and so-called *false flags* – behaviour that looks like the behaviour of the pattern but is not an actual manifestation.

The paper is organised as follows. In Sect. 2 Principal-Agent Theory is introduced, its relation to event logs is explicated, related work is discussed, and the derivation of the patterns from the theory is outlined. Sections 3, 4, and 5 discuss the various patterns. Empirical validation of the patterns is addressed in Sect. 6. Section 8 briefly discusses and summarises our work and identifies opportunities for future work.

2 Theoretical Background

In this section we first outline Principal-Agent Theory (Sect. 2.1), which forms our theoretical basis, then we make a link between this theory and process mining by describing how events logs can potentially be used to unearth weasel behaviour (Sect. 2.2), we discuss related work (Sect. 2.3), and finally we relate the theory to the patterns proposed (Sect. 2.4).

2.1 Principal-Agent Theory

Principal-Agent Theory considers the relationship between two actors. Specifically, it describes the relationship between an actor (e.g. organization or individual) giving an order (the principal) and an actor executing this order (the agent). According to the theory, both actors strive to maximise their advantage, whereby an asymmetrical distribution of information exists between the principal and the agent [15]. The principal has incomplete information about the agent's behaviour and, therefore, must trust him to some degree [22]. The agent has the advantage that they have more information available on the execution of the order. This asymmetry leads to an imbalance of transaction power which is typically exploited by agents as they behave opportunistically. Opportunistic behaviour describes actions from an agent that occur due to the maximisation of the agent's own interests which are often different or even contrary to the principal's interest. It causes different challenges in agency relationships: *ex-ante*, the misrepresentation of ability (adverse selection due to hidden characteristics) as well as *ex-post*, the lack of effort (moral hazard due to hidden information or actions) and breach of word (hold-up due to hidden intentions) problems occur [11]. Agency cost under Principal-Agent Theory can be defined as the total of the principal's monitoring cost, the agent's bonding expenditures, and residual loss [15]. Managers must consider a trade-off between the cost of information acquisition and the cost of outcome and transfer risks [11].

2.2 Event Logs

Event logs contain a sequence of time-stamped events that are a reflection of the execution of tasks by employees guided by processes supported by process-aware information systems (PAIS). Apart from the task executed, the time at which this occurred and the case to which the task belongs, these systems may also record the resource(s) involved and values of various data attributes. Therefore, these logs can be used to monitor the actions of agents in processes. In the context of Principal-Agent Theory, process mining has the potential to uncover hidden actions, i.e., detecting how employees behave differently from the way dictated by official processes. Depending on the availability of data, this can cover activities, processing times, as well as originators. In this setting, principal refers to a person who is responsible for the process but not performing tasks in that process. All employees working in a process are considered to be actors, independent of the nature of their relationship. In some cases, there are hierarchical relationships between actors in a process (e.g., a manager and their subordinates can perform work in the same process).

2.3 Related Work

One area of focus in prior work is the identification of opportunistic behaviour in organizations from a principal-agent perspective independent of the use of event logs. These studies identify opportunistic behaviour related to the way how work is performed (routing), the performance of agents, and social relations between agents [7,12,14,19]. Kellogg et al. [16] highlight the role of using algorithms to control employees through recording and rating their actions but not through the explicit use of event logs. The authors refer to a counteraction of employees termed "algoactivism" where they manipulate the recording of their data. The usefulness of identifying patterns to reduce co-destructive behaviour in the context of crowdsourcing was highlighted by [6]. Haag et al. [13] address deviant IT usage behaviour using devices outside of official workflow software and thus address undesired consequences for organizations. Another area of focus in prior work manifests in studies in the domain of process mining which concern the detection of behaviour in a descriptive way, i.e., identifying patterns with regard to performance, interactions, and so on [3,18,20]. With regard to workflow patterns, [23] identify the dimensions of control flow, data, resource and time as being important. Although the paper is not based on theory and focuses on workarounds as a positive expression of employee behaviour, it is informing about the importance and nature of unusual deviations to be detected in event logs. The work of [24] constitutes initial work on the use of behavioural theory, applied to identify organizational groupings of employees. Overall, however, the use of behavioural theory to unearth human behaviour by examining the footprints such behaviour leaves behind in event logs has not yet been explored.

2.4 Conceptualisation of Patterns Based on Principal-Agent Theory

The starting point of conceptualising the patterns is the nature of the data being used. Process mining includes a focus, through event log analysis, on the order of activities, the performance of processes, activities, and employees as well as the social interactions between employees [1]. Given that Principal-Agent Theory highlights opportunistic behaviour it is expected to observe reflections of such behaviour in such logs. Hence, the categories in which we identify patterns include those that are rerouting-related (i.e., having personal interests to deviate from the officially defined process flow), performance-related (i.e., trying to hide true performance) and social-related (i.e., using social relationships for personal advantage). On a more specific level within these categories, we can again combine the nature of event log data with the theoretical underpinning of Principal-Agent Theory. First, within the re-routing category of patterns, the main parameters are the work to be done (activities) and who is performing the work (originators). This can then be extended to the connection of activities, i.e., on the one hand the control flow (Re-Ordering) and on the other hand the process instances (Preferential Work Selection). These patterns all follow the idea that agents have a personal interest to increase their personal gain by hiding information and actions. Second, within the performance category of patterns, actors may want to hide their actions in performing activities which either try to manipulate measures (Performance Masking) or hide the real activities. In the latter case this can be either hiding low performance because of inability (Overwork Hiding) or lack of interest (Performance Blow-out). Finally, there also can be intention to perform a hold-up with customers by performing more and charging more (Gold Plating). Third, within the social-related category of patterns, agents can cooperate to mask their performance or work against each other. For cooperation, there can be a drive to simply enjoy socialising (Idling). When working against each other, agents try to get other actors to do the work for them, driven by moral hazard (Social Borrowing). There can also be mixtures of cooperation and mutual obstruction when an individual agent wants to obtain a free-ride at the expense of others (Social Loafing) or when a group has the intention to increase their performance at the expense of others (Peer Mobbing). Such cooperation can also be against the boss (Boss Mobbing).

Table 1 provides an overview of the various categories and their patterns. We will describe the patterns in detail, including their potential detection through the application of process mining, in the following three sections.

Table 1. Weasel Behavioural Patterns

Rerouting-related	Performance-related	Social-related
1. Activity Deviation	5. Performance Masking	9. Idling
2. Originator Deviation	6. Performance Blow-out	10. Social Loafing
3. Re-Ordering	7. Overwork Hiding	11. Peer Mobbing
4. Preferential Work Selection	8. Gold Plating	12. Boss Mobbing
		13. Social Borrowing

3 Rerouting-Related Patterns

Pattern 1: Activity Deviation

Description. Activities are observed in the log that are not a part of the process model. This can happen only if the process management system is flexible enough to allow this or in case the process is manually performed (but the actions are recorded automatically). Consider for example management of unforeseen exceptions or systems that are so flexible that they allow arbitrary workarounds. While some activity deviations can represent learning experiences [2], others – and these are the focus of the pattern – may be a manifestation of working around rules or of otherwise creating actor-related benefits (e.g., making their life easier).

Examples. Customers may be requested to sign papers even though it is not necessary anymore (e.g., due to digital signatures or changed approval procedures) or a clerk in a bank always checks with the head of the branch to avoid being caught doing something wrong even though this runs counter to instructions from the CEO. Other examples would be employees in a production environment unnecessarily checking objects into machines or employees in a hospital performing extra diagnosis steps where this is not in the interest of the principal.

Rationale. It may be that agents believe that performing a certain action is necessary or desirable (e.g., saving time in their own interest) while their principal may not think so and it isn't part of the official protocol.

Manifestation and Detection. The pattern can be detected by observing activities recorded in the log which are not part of the official process model (conformance checking).

Remedy. Enforcing execution of the process through the application of a process-aware information system may help alleviate or eliminate the problem. If such a system is not used, spot checks and more stringent monitoring may be fruitful.

Potential Side-Effects. Flexibility in processes has advantages and drawbacks. In some cases a step not or no longer part of the process model may be beneficial for the organization and this may be picked up by an experienced actor. Reducing flexibility in that case may lead to undesirable outcomes. It may also take away initiative and motivation of agents with positive intentions.

Potential False Flags. This pattern can occur if the original process model was not updated. Then its occurrence is not due to intentional behaviour.

Pattern 2: Originator Deviation

Description. Activities are undertaken by resources that are not (officially) allowed to work on them. While this may be beneficial on occasion, here focus is on those occasions where this is not desirable. For such deviations to be able to happen, a process management system needs to be flexible enough or the process is performed manually (but the actions are recorded automatically).

Examples. An experienced employee may wish to perform a certain action instead of a more junior and inexperienced employee though this runs against the instructions of the higher echelon. It may be the case that performing a given activity has certain benefits (e.g., financial or kudos) and taking over the activity is seen as beneficial for one's career. Another example is a principal is checking on an agent because he does not trust them sufficiently.

Rationale. From a behavioural perspective, it may be that an agent thinks that they can do the task better than another agent in order to present themselves as the better agent (e.g., taking credit) or to reduce the risk of something going wrong (e.g., hiding mistakes).

Manifestation and Detection. In the event log, there are activities performed by originators that are not assigned to that originator in the official process model (again, conformance checking can detect this).

Remedy. If the process is controlled through a PAIS, stricter allocation rules can be put in place. If the process is not supported by such a system, disincentives can be put in place for conducting work outside expected tasks.

Potential Side-Effects. In certain instances, the pattern is desirable, such as when it is evident that the outcome will be superior. In such instances, there should be sufficient flexibility to permit transfer of work.

Potential False Flags. If the official process model has not been updated what seems to be a transfer of work may in fact not be.

Pattern 3: Re-Ordering

Description. Activities are performed by agents in a sequence that runs contrary to regulations, best practice, or to what is prescribed in the process model. This is only possible in case the process is not strictly controlled by the process management system or through manual oversight.

Examples. Employees in a production environment checking objects into machines for processing in an order that is wasting resources is an example of the manifestation of the pattern. Another example occurs when employees in a hospital perform steps for diagnosis in a different order than prescribed, which is not in the interest of the principal.

Rationale. From a behavioural perspective, it may be that subordinates believe that processing an instance requires a different order of activities (e.g., getting a better outcome in their personal interest) while their superior may not think so and it isn't part of the official protocol.

Manifestation and Detection. Activity sequences in the log, as determined by timestamps, that do not correspond to activity sequences in the official process model can be detected through conformance checking.

Remedy. The use of a PAIS can help alleviate the issue as can educating the workforce around proper procedure and the rationale behind it.

Potential Side-Effects. The use of a PAIS may enforce a way of working that is too strict and no longer allows meaningful workarounds.

Potential False Flags. This pattern can occur if the original process model was not updated. Then its occurrence is not due to intentional behaviour.

Pattern 4: Preferential Work Selection

Description. Agents specifically choose process instances or work items with particular characteristics. This may be because they find that work relatively easy or because of the credit or pay-off that they expect to get. The system has to allow for individual selection of cases or work items by agents.

Examples. Agents may choose to work on cases that involve high value customers in order to obtain their favour, receive other inducements, or more easily achieve their KPIs. Consider clerks choosing insurance claims with fewer attachments, expecting the work to be of lower complexity.

Rationale. Agents aim to minimise their effort required to achieve certain performance criteria.

Manifestation and Detection. Cases of a certain type are chosen more often than expected by an agent or the order cases are worked on after their initiation is not what is expected.

Remedy. One can introduce a random element in the assignment of process instances to reduce occurrences of this pattern. Another possibility is to control, either through a PAIS or manual oversight, who can perform what type of work when. A portfolio approach allows a predetermined distribution of work among agents and would also prevent the pattern from occurring.

Potential Side-Effects. Preventing the pattern may prevent specialisation from occurring which may be sub-optimal in terms of overall performance. Some agents may simply be better suited for certain types of work.

Potential False Flags. An agent or a group of agents has specialised in a certain type of work or task and there is agreement that they perform the work more often.

4 Performance-Related Patterns

Pattern 5: Performance Masking

Description. This pattern occurs when agents act in such a way, through their dealing with or knowledge of IT systems, that their true performance cannot (easily) be established.

Examples. When time on task is an important key performance indicator and this is measured by the times the system thinks someone has opened and then closed a work item, an actor can fake the actual working time by opening a work item, transferring its content online (e.g., by taking a photograph), closing the work item, finalising the work item offline, and then transferring the work back into the system. In such cases the actual time on task will seem substantially less than what happened in reality. Another example in the context of insurance claims can be the regular contacting of customers so that it seems that the claim handlers involved are genuinely concerned about the customer in order to achieve higher customer satisfaction. A better outcome may however not necessarily be achieved.

Rationale. Agents wish to hide that they are underperforming or would like to optimise their actions to maximise a bonus.

Manifestation and Detection. Unusual action sequences, performed in relatively short periods of time, may be an indication of the occurrence of this pattern. It could for example be unusual to see many open-close actions performed in short time frames or regular but brief customer interactions.

Remedy. When detecting this pattern repeatedly for an agent, one could choose to closely monitor their work. Another option is to rethink the bonus structure to avoid wrong incentives.

Potential Side-Effects. Underperforming agents or agents that try and take advantage of specific situations undermine group morale if the issue is not addressed.

Potential False Flags. An agent is always interrupted by phone or colleagues when performing their work.

Pattern 6: Performance Blow-out

Description. Working times for activities may be determined by service level agreements (SLAs). This pattern occurs when agents maximise their time when working on these activities by stretching their work, acting as if they are doing the work, or by falsely recording end times.

Examples. From a behavioural perspective, an agent can perform the job of a particular case quickly but chooses to use the remaining time until the SLA is met to play on the phone and finish the job by clicking on time. Another example is that parts at a machine are checked out when the time is met and until then an agent might go for a cigarette break.

Rationale. An agent wants to hide their real performance and gain extra free time while being at the workplace but not actually working.

Manifestation and Detection. The pattern can be detected when different agents take very different times for work of a comparable nature. It may also manifest itself when work performance of an actor decreases over time.

Remedy. Incentives can be provided for completing work faster than required by service-level agreements (SLAs). If the issue is suspected to arise from boredom with particular work, tasks could be rotated more often, particularly tasks of varying degrees of complexity, so that agents experience their work as sufficiently challenging. Real-time monitoring of work can also be deployed to pick up early on that no or little work is being performed.

Potential Side-Effects. More intense monitoring of work can lead to a reduction in morale and trust.

Potential False Flags. The work to be done is more complicated than expected/calculated and an agent really continually needs the full time span.

Pattern 7: Overwork Hiding

Description. Work is performed by agents outside their working hours, but this is not due to them taking on more work than they should have, but rather their inability to do their allocated work in the time frame expected for employees with their stated level of skills and experience.

Examples. For example, an employee was promoting, and in fact overstating, their abilities during the hiring process and as they want to keep their job after being hired they want to mask the time they require to perform work. Another example is an employee wanting to be perceived as high performing by others, e.g., their supervisor, and thus not wishing to be seen as relatively slow in performing work.

Rationale. An agent wants to conceal poor performance as they claimed a high performance ability. Hence work takes longer than it should and the agent does not want to have this revealed.

Manifestation and Detection. Timestamps of activities performed by particular agents show that these activities were performed outside the official working times of these agents.

Remedy. The type of work allocated to agents that try to hide their overwork can be modified to better suit their skills and experience. Also better screening of agents to ensure a match between abilities and requirements can be helpful.

Potential Side-Effects. Actual working times for certain types of work may be estimated wrongly. This may influence work allocation and how other agents are evaluated.

Potential False Flags. An agent may have been overloaded and due to their commitment to work felt obliged to finish it outside work hours. Also, overwork and psychological exhaustion may occur after a certain period of time.

Pattern 8: Gold Plating

Description. Gold plating refers to the phenomenon of introducing higher levels of regulation or conducting more work than what is required or prescribed, or to the addition of features or services that are not required by the original SLA with the customer.

Example. A car may require a government-approved certificate every year for the car to be considered road-worthy. A garage involved could choose to perform tasks that are not strictly required. This is possible as customers often cannot assess what was really necessary to be done.

Rationale. Agents have an interest to perform extra yet unnecessary work that increases their personal goal achievement/bonus.

Manifestation and Detection. Cases with certain characteristics (e.g., related to certain types of customer) are taking longer or cost more than expected, or they contain unexpected tasks.

Remedy. Transparency in terms of comparative performance may help mitigate the occurrence of the pattern.

Potential Side-Effects. Gold plating may lead to customers feeling deceived and thus to reduced trust on their part.

Potential False Flags. The specifics of a case may require a higher level of service than expected.

5 Social-Related Patterns

Pattern 9: Idling

Description. This pattern occurs when agents do not perform work during work time, but rather focus on non-work related activities or spend time socialising with their colleagues.

Examples. Employees smoking cigarettes alone or with colleagues, playing solitaire, having a casual chat etc. during working time are all examples of idling.

Rationale. Agents want to use (part of their) official working times for something else (e.g., social interaction) as they either perceive the work as boring, overwhelming, or unfair given their current salary, but do not want to be perceived as bad performers at the end of the day.

Manifestation and Detection. If agents do not make any specific effort to mask their idling, this pattern can be detected in an event log by observing that the idling agents take substantially more time for certain tasks than others. In case their working time is also explicitly recorded, then long periods will be observed where the agent is not performing any task.

Remedy. Assigning more stimulating and challenging work to employees in case they are bored may be beneficial in reducing idling. In order to detect longer idle times, systems that check whether an agent is actually performing work can be used.

Potential Side-Effects. Close monitoring of agents may lead to a micromanagement work culture with low trust.

Potential False Flags. Agents can often be interrupted with e.g. questions and thus stop working on their activities repeatedly.

Pattern 10: Social Loafing

Description. This pattern emerges in the context of group work, where a group member shirks work and does not pull their weight in achieving a common goal, e.g. as the group member hopes for a free (or cheap) ride. It is a cause of group productivity being less than expected based on the typical individual performance of the members involved.

Examples. Group members guilty of social loafing may skip meetings or contribute minimally to them. They may also not perform their allocated tasks, or perform them in a minimal way. Taking long coffee breaks, lengthy personal discussions, missed appointments, increased absences, all constitute examples of loafing behaviour.

Rationale. An agent uses the group to hide their non-performance (e.g., as they are lazy or incompetent). Group dynamics (e.g., an agent considering themselves to be superior) may be an alternative cause for underperformance witnessed.

Manifestation and Detection. The pattern can be detected in logs when analysing the relative contribution of individuals being assigned to a group (and performance is measured on this level) compared to working alone (and performance is measured individually) towards achieving process goals (e.g., low cycle time). For this purpose, algorithms from organizational mining can be used [24].

Remedy. Individuals can be assigned to different groups over time, the relative performance determined and group assignments can then be made for the best joint performance that can be achieved. Group cohesiveness has been recognised as a key factor helping to reduce social loafing [17]. It can encourage a higher feeling of responsibility. Group cohesion can be measured with questionnaires and group spirit exercises can be assigned where necessary.

Potential Side-Effects. While group assignments may lead to high performance, they may not achieve a sustainable level of satisfaction among individuals due to the creation of a highly competitive environment. Also, group stability may be affected by frequent changes in group composition.

Potential False Flags. An individual is simply not performing well in a social context due to group dynamics or they are not performing so well in terms of individual KPIs but make the overall team perform better (sometimes referred to as the "Shane Battier effect" [21]).

Pattern 11: Peer Mobbing

Description. Peer mobbing occurs when an agent, or a group of agents, chooses to degrade the performance of a colleague or a group of colleagues, operating at peer level, by taking over their work without their approval or even their knowledge (at least initially).

Example. A group of colleagues takes away cases from one of their own because these cases are easy and makes their own performance look good. This is possible as work is picked from a shared pool (e.g., work items on a shared work list).

Rationale. The performance of a group can be enhanced by taking over work from others that improves the group's performance, has a high pay-off, or is relatively easy for the effort required.

Manifestation and Detection. The pattern can be detected by observing that work of a certain nature is performed relatively more often than would be expected by a group of agents compared to other individual agents.

Remedy. One can put limits on how much work of a certain nature can be performed by certain agents within a determined time frame. Another option is to shuffle groups from time to time to avoid collusion between agents.

Potential Side-Effects. Preventing the pattern may prevent specialisation from occurring which may be suboptimal in terms of overall performance. Some agents may simply be better suited for certain types of work.

Potential False Flags. An agent or a group has specialised in a certain task and there is agreement that they perform those tasks more often.

Pattern 12: Boss Mobbing

Description. Boss mobbing occurs when agents want to mob their boss by repeatedly performing poorly as a team in order to make their boss look bad. At some stage, a higher echelon may conclude that the poor performance of the team is due to the poor performance of the boss.

Example. A group of agents feels stressed by their boss and thus decides to start performing poorly with the hope of the boss getting fired. For example, they could collectively choose to not conduct work in time or perform the work below average standard.

Rationale. Agents may choose to mob their boss, if the latter is too demanding, has high expectations, or is too controlling (e.g., acts like a micromanager). Achieving removal of the boss is expected to lead to a more relaxed work environment for the agents. Note that this requires collusion between (almost) all agents involved as otherwise decreased team performance may be attributed to the select group of agents involved. It should also be pointed out that this happens when the work involved is rather standard and agents are not highly motivated knowledge workers. Job security is also a factor as agents may not be keen to risk their work through conscious underperformance; hence the pattern is more likely to occur when agents feel more secure in their employment.

Manifestation and Detection. Every agent of a team is showing bad performance in terms of goals set. This becomes more striking in case these workers inhibited very different levels of performance in the past or in case their qualifications and levels of experience are very different. Homogeneous performance would not be expected in such cases.

Remedy. Rotating team members regularly to avoid groups of agents to become too comfortable and to consider colluding, and rotating principals regularly are ways to prevent the pattern from occurring. Incentives (e.g., awards, bonus payments) for increased performance can also help mitigate the pattern's occurrence.

Potential Side-Effects. By not giving a team or a boss enough time to adjust to each other, team performance may suffer and there is no chance of recovering in the future.

Potential False Flags. A team really does not work together well and thus does not perform well.

Pattern 13: Social Borrowing

Description. Social borrowing occurs when agents manage to get other agents to do (some of) their work for them without them being credited or acknowledged for it and without it being part of the work that these agents are expected to contribute to.

Examples. An actor can ask a colleague to perform a statistical task for them as they are not confident in performing it themselves. Another actor may simply want to go home early so asks a colleague to help finish one of their tasks.

Rationale. Social borrowing may occur due to agents not being able to perform tasks assigned to them, it may happen in order to reduce their workload, or it may be a means to increase recognition by peers and superiors (other than the assisting agents).

Manifestation and Detection. A specific agent being present at the same time at work according to the log, leads to a decrease of performance of another agent compared to when the specific agent is not present. This could be reinforced if the actor has a marked increase in performance in the conduct of certain tasks when this other agent is present.

Remedy. Identify the specific agent and make sure that agents that can be used to do his work are not present at the same time.

Potential Side-Effects. An unhealthy social atmosphere may be created when talking to each other or collaborating with each other is regarded as suspicious in terms of social borrowing.

Potential False Flags. Idling with a specific agent could be happening due to a social relationship between agents and the agent with a decrease in performance is not that good in masking their idling compared to the other agent.

6 Empirical Evidence

To get an impression of the usefulness and importance of the patterns identified in terms of validation, we conducted a quantitative empirical study. The target group are managers on different levels in organizations (e.g., banks, insurance companies) who focus on routine work and information processing who have identified the patterns in their work life – not necessarily in an event log. The questionnaire is built on the patterns, i.e., each pattern is presented in a randomised order, and we employ single-item measures referring to whether a respondent has identified the respective pattern in their work life and whether they find the ability to identify the pattern useful. As such we cover occurrence as well as relevance. We choose single-item measures as we want to get a quick and first impression before digging deeper into pattern specifics and for this, single-item measures are considered reliable [5]. The scale for each question ranged from 1 – Do not agree at all – to 5 – Fully agree. In addition we ask for main characteristics of the respondents such as age, gender, work experience in the profession, current position, and industry in which they work. We also allow the participants to make a general comment on the patterns at the end of the questionnaire. The questionnaire and dataset can be found here: https://tinyurl.com/Detectingweasels.

We acquired 21 respondents (61 impressions) with a convenience sample via LinkedIn and management forums. Of the total number of respondents, 19 fully answered the questionnaire; the other two provided feedback for some, not all, patterns. The participants are 36.9 years old on average, have a profound average work experience of 10.4 years with 3.1 years in their current position on average; 57.1 per cent are female, 38.1 per cent are male, and 4.8 per cent are others. The quantitative results in Table 2 show that there are indeed different assessments with regard to the patterns, but they also show a quite high occurrence and relevance overall.

Table 2. Quantitative Results

	Mean Occurrence	Mean Relevance
1. Activity Deviation	3.55	4.15
2. Originator Deviation	3.50	3.95
3. Re-Ordering	3.60	4.20
4. Preferential Work Selection	3.60	4.10
5. Performance Masking	3.55	3.90
6. Performance Blow-out	3.70	4.20
7. Overwork Hiding	3.05	3.58
8. Gold Plating	3.84	4.11
9. Idling	4.00	4.21
10. Social	4.05	4.30
11. Peer Mobbing	3.10	3.60
12. Boss Mobbing	2.05	3.15
13. Social Borrowing	3.42	3.90

The qualitative results (eight respondents) support the practical relevance (four respondents), but also highlight issues with some patterns. One respondent refers to a more positive view on some weasel behaviour pointing out that individuals may have a skill deficit and learn from processing. A second respondent suggests that sometimes detected weasel behaviour can even have benefits for the organization. A third respondent mentions that using weasel-driven KPIs will lead to a quite mechanic work environment. While we addressed the first two concerns in our patterns in the potential false flag part, the third one is more fundamental. We agree that this aspect is relevant, but is a general issue associated with using KPIs (and these are used in many work environments). Finally, it should be noted that one respondent could not distinguish between social loafing and idling.

7 Discussion

The empirical results show the relevance of the patterns which are varying but nevertheless all values are above a value of 3. While individual patterns are described in distinction to each other, they might not necessarily point to an employee showing weasel behaviour. This aspect is reflected in the potential false flags, but to avoid misunderstandings a combination of different patterns should be considered for assigning a weasel status. The combination does not necessarily has to be in the different categories, but a typical threshold should be defined for a sum of individual patterns that is identified for a person. The detection of the respective patterns can occur with standard event logs and respective algorithms should be developed based on the pattern descriptions. The descriptions can be used to develop such algorithms. The descriptions indicate which parameters should be considered and provide measurements and provide guidelines for assignments of agents if implemented as algorithms. While this detection can be standardised and automated, the interpretation is up to experts. They are responsible for further interpretation and to analyse important sequences. This combination of automation and human activities allows to delegating the vast scanning of logs and focusing human time on the more complicated interpretation task. It is important to have the human in the loop to avoid employees being wrongly judged and legal issues occurring.

8 Conclusion

Our work has several theoretical implications. First, we are pioneers in describing how behavioural theory can be used to identify general patterns of individual behaviour at the workplace. The advantage is that understanding why things happen is a necessary precursor for support for business-decision making grounded in cause-and-effect. Second, we contribute to the BPM literature by describing specific patterns for event logs. With this, we provide a blueprint that can also be used for other theoretical lenses and a basis for the development of process mining algorithms for detecting weasel behaviour. Third, our empirical results show the relevance of the proposed patterns for business-decision making.

Practical implications are that companies have quite specific patterns at hand that they can use to apply to their event logs to identify weasels. It helps them to be faster in detecting such individuals without interfering with the workforce. For a practical application, a company should identify which patterns in combination are relevant or how many flags according to the patterns have to be visible in this regard to identify an employee as a weasel. This can avoid unnecessary discussions as well as false accusations.

Our work has several limitations. First, we provide limited empirical evidence from the perception of managers. While it shows a first idea of the relevance of the patterns, more representative results are necessary. Second, we focus on Principal-Agent Theory only. The application of other behavioural theories may lead to supplementary patterns or different ways of analysing and understanding event logs. Thirdly, the use of patterns is always a reduction of the heterogeneity of reality which results in a loss of information. But they also allow for the identification of certain types of behaviour that are otherwise vague. Consequently, these patterns can be subject to discussion and (re-)used across different organizations. Fourthly, the patterns are not covering every possible type of weasel behaviour, but cover the range of what has been identified as conceptually relevant and deducted from theory. In this regard, it has to be kept in mind that the theory is providing the reasoning and some conceptual direction, but will not allow to define a complete set of patterns. Fifthly, there is an ethical discussion around employees having to take measures in order to resist unfair and unreachable goals and work orders. While we agree that this is an important topic, our focus here has been on how to detect opportunistic behaviour from an agent perspective, independent of a higher dimension of workload definition and fairness.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
2. Adams, M., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Facilitating flexibility and dynamic exception handling in workflows through worklets. In: Belo, O., Eder, J., Falcão e Cunha, J., Pastor, O. (eds.) The 17th Conference on Advanced Information Systems Engineering (CAiSE 2005), Porto, Portugal, 13–17 June, 2005, CAiSE Forum, Short Paper Proceedings. CEUR Workshop Proceedings, vol. 161. CEUR-WS.org (2005)
3. Aysolmaz, B., Nemeth, M., Iren, D.: A method for objective performance benchmarking of teams with process mining and DEA. In: Rowe, F., et al. (eds.) 29th European Conference on Information Systems - Human Values Crisis in a Digitizing World, ECIS 2021, Marrakech (2021)
4. Badakhshan, P., Wurm, B., Grisold, T., Geyer-Klingeberg, J., Mendling, J., vom Brocke, J.: Creating business value with process mining. J. Strateg. Inf. Syst. **31**(4), 101745 (2022)
5. Bergkvist, L.I.: Appropriate use of single-item measures is here to stay. Mark. Lett. **26**(3), 245–255 (2015)
6. Bidar, R., ter Hofstede, A.H.M., Sindhgatta, R.: Co-destruction patterns in crowdsourcing. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) CAiSE 2020. LNCS, vol. 12127, pp. 54–69. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49435-3_4
7. Biener, C., Eling, M., Landmann, A., Pradhan, S.: Can group incentives alleviate moral hazard? the role of pro-social preferences. Eur. Econ. Rev. **101**, 230–249 (2018)
8. Brock, M.E., Martin, L.E., Buckley, M.R.: Time theft in organizations: the development of the time banditry questionnaire. Int. J. Sel. Assess. **21**(3), 309–321 (2013)
9. Chullen, L.C., Dunford, B.B., Angermeier, I., Boss, W.R., Boss, A.D.: Minimizing deviant behavior in healthcare organizations: the effects of supportive leadership and job design. J. Healthc. Manag. **55**(6), 381–97 (2010)
10. Dunlop, P.D., Lee, K.: Workplace deviance, organizational citizenship behavior, and business unit performance: the bad apples do spoil the whole barrel. J. Organ. Behav. **25**(1), 67–80 (2004)
11. Eisenhardt, K.: Agency theory: an assessment and review. Acad. Manag. Rev. **14**(1), 57–74 (1989)
12. Englmaier, F.: Moral hazard, contracts and social preferences: a survey. In: Agarwal, B., Vercelli, A. (eds.) Psychology, Rationality and Economic Behaviour: Challenging Standard Assumptions, pp. 125–139. Palgrave Macmillan UK, London (2005)
13. Haag, S., Eckhardt, A., Venkatesh, V.: Deviant affordances. when tensions, deadlocks, and nonconformance generate performance. Management Information Systems Quarterly Forthcoming (2023). 10.25300/MISQ/2022/14340
14. Holmstrom, B.: Moral hazard in teams. The Bell J. of Econ. **13**(2), 324–340 (1982)
15. Jensen, M., Meckling, W.: Theory of the firm: managerial behavior, agency costs and ownership structure. J. Financ. Econ. **3**(4), 305–360 (1976)
16. Kellogg, K.C., Valentine, M.A., Christin, A.: Algorithms at work: the new contested terrain of control. Acad. Manage. Ann. **14**(1), 366–410 (2020)

17. Liden, R., Wayne, S., Jaworski, R., Bennet, N.: Social loafing: a field investigation. J. Manage. **30**(2), 285–304 (2004)
18. Pika, A., Leyer, M., Wynn, M., Fidge, C., ter Hofstede, A., van der Aalst, W.: Mining resource profiles from event logs. ACM Trans. Manage. Inf. Syst. **8**(1), 1–30 (2017)
19. Salin, D.: Ways of explaining workplace bullying: a review of enabling, motivating and precipitating structures and processes in the work environment. Hum. Relat. **56**(10), 1213–1232 (2003)
20. Song, M., Van der Aalst, W.M.P.: Towards comprehensive support for organizational mining. Decis. Support Syst. **46**, 300–317 (2008)
21. Toomer, J., Caldwell, C., Weitzenkorn, S., Clark, C.: Introduction: Catalytic Teamwork in Action, pp. 1–6. Emerald Publishing Limited (2018)
22. Treiblmaier, H.: The impact of the blockchain on the supply chain: a theory-based research framework and a call for action. Supply Chain Manage. Int. J. **23**(6), 545–559 (2018)
23. van der Waal, W., Beerepoot, I., van de Weerd, I., Reijers, H.A.: The SWORD is mightier than the interview: a framework for semi-automatic workaround detection. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds.) BPM 2022. LNCS, vol. 13420, pp. 91–106. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16103-2_9
24. Yang, J., Ouyang, C., ter Hofstede, A.H.M., van der Aalst, W.M.P., Leyer, M.: Seeing the forest for the trees: group-oriented workforce analytics. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 345–362. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_22

Business Process Management Maturity and Process Performance - A Longitudinal Study

Arjen Maris(✉), Guido Ongena, and Pascal Ravesteijn

HU University of Applied Sciences, Utrecht, The Netherlands
{arjen.maris,guido.ongena,pascal.ravesteijn}@hu.nl

Abstract. Longitudinal Business Process Management (BPM) studies are rare. BPM maturity and process performance can be used to quantify an organization's BPM evolution. This research aims to examine the growth of BPM maturity over time and its impact on process performance inside an organization in continuous transformation. Over a seven-year period, BPM maturity and process performance were measured annually at a Dutch university. During this time, the organization has undergone an organizational restructuring with a focus on process management and has temporarily switched completely to digital education propelled by the Covid-19 crisis. Based on a repeated cross-sectional study (N = 921), the results present key BPM maturity features that are critical during disruptive organizational transformations. Furthermore, we found that BPM maturity is positively related to process performance throughout organizational changes during the period of our research.

Keywords: BPM maturity · Process performance · Organizational dynamics · Longitudinal research

1 Introduction

BPM refers to the process-oriented structuring of organizational activities in order to optimize and integrate business processes, obtain a competitive advantage, and create and distribute value [1]. While BPM seems to have a strong positive impact on organizational performance in terms of efficiency and effectiveness [2], it also creates tensions when responding to contingencies [3].

BPM maturity models have been developed to measure the extent to which an organization has implemented BPM capabilities and is able to apply BPM effectively. The outcomes of these models should assist organizations in determining which BPM capabilities to deploy and how to boost performance [4]. However, most BPM maturity assessments only measure process maturity in a descriptive manner and the practical relevance of these models is unclear due to a lack of empirical evidence [5].

It has long been recommended to analyze how BPM maturity affects organizational process performance over time, for example, to show or even forecast which BPM capabilities should be taken into account to increase performance or to react to organizational changes [6, 7]. However, no long-term effect measurement has been reported so far.

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 355–371, 2023.
https://doi.org/10.1007/978-3-031-41623-1_21

Despite the observation that educational managers score a higher level of perceived BPM maturity than employees such as lecturers [8], there have been few initiatives in higher education to apply business process management [9, 10] and a number of sector-wide process initiatives have failed [11]. This could be the result of organizational dynamics [11] and less structured processes in service organizations [12], which also result in a lower BPM maturity score for service organizations compared to product organizations [13]. This sparked our interest in the use of BPM maturity models in higher education.

As a result, we aim to close this gap in BPM literature, by applying a longitudinal research design to investigate how stable BPM maturity is in relation to organizational dynamics. In particular, we aim to answer the following research question: *How do changes in BPM maturity affect higher education process performance over time?*

The next part of this article provides the theoretical foundation that focuses on the key elements of this research, BPM, BPM maturity and process performance and longitudinal research. After this section, the quantitative research approach with PLS-SEM is specified. This is followed by a description of the empirical findings in the results section. The results are discussed, and the limitations of this study are stated in the discussion section and the article ends with the conclusion and recommendations for further research.

2 Theoretical Background and Research Model

2.1 Business Process Management

BPM is defined as "a holistic organizational management practice focused on the identification, definition, analysis, continuous improvement, execution, measurement, monitoring, and analysis of intra- and inter-organizational business processes" [14]. This management practice combines business management methods like business process redesign, quality management methods like total quality management, and process-oriented digital innovations like workflow management and enterprise resource planning software [15, 16]. It differs from traditional hierarchical management in that the emphasis is on continual efficiency and effectiveness improvement of process performance through the use of BPM lifecycles and digital components [17]. Therefore, BPM is a method to manage change through business process improvement, embracing the full process life cycle, from analysis and design to implementation, automation, and execution of business processes in order to improve process performance [18].

Additionally, Grisold, et al. [19] show that BPM also can aid in process innovation and Brocke, et al. [20] take into account that organizations and their environments are always changing. Their viewpoint has ensured that BPM evolves into a broader process science approach. However, empirical research establishing dependent and independent factors, or determining whether improvements such as digital innovations support and even increase BPM and hence process performance, is limited [19, 21].

2.2 BPM Maturity and Process Performance

BPM maturity models are used to quantify and communicate an organization's ability to manage its business processes. Humphrey's capacity maturity model is one of the first in the history of process maturity models [22]. There are currently numerous BPM maturity models that are utilized for various reasons, such as descriptive, prescriptive, and comparative analyses [5, 6]. Most BPM maturity models are descriptive [5]. In our study we use the BPM maturity scan of Ravesteijn et al. [23]. This scan is inspired by the research of Rosemann, de Bruin and Hueffner [14, 24] and was first used in 2010 to measure BPM maturity of organizations in the Netherlands [25]. Subsequently a regular benchmarking research was done to examine various views on the relationship between BPM maturity and process performance [13, 23, 26]. The studies conducted show that maturity of BPM improves process performance. These justifications, however, are based on snapshots in a specific context (e.g. place) and time. We do not know whether the effect lasts over a longer period of time because these samples have never been sequenced previously. As a result, we developed our main hypothesis:

H1: BPM maturity has a long-term positive impact on process performance.

Several academics have previously operationalized BPM maturity [27]. Ravesteijn, Zoet, Spekschoor, and Loggen [23] operationalized BPM maturity in seven dimensions: Process Awareness, Process Description, Process Measurement, Process Control, Process Improvement, Resources and Knowledge and Information Technology.

The five items of the dimension *process awareness* assess higher management's recognition of the value of a process-oriented organization and inclusion in the organization's strategy. The extent to which processes and related information are recorded inside the organization is used to assess *process description* (7 items). Six *process measurement* items encapsulate the degree to which an organizational structure to monitor and manage processes is in place to improve processes. The six *process control* items are concerned with whether process owners are designated inside the company who are responsible for managing processes. The seven *process improvement* items describe how far the organization seeks to continuously improve processes and if a structure is in place to support this. Five items analyze if the organization has adequate resources (such as individuals with process knowledge) to build a "culture of process orientation" in the penultimate dimension *resources and knowledge*. The eight *information technology* items analyze the organization's ability to use IT to develop, model, and execute processes, as well as offer real-time measurement data (key performance indicators).

Based on Hüffner [28] and Rudden [29], the construct process performance is added as a dependent construct to the BPM maturity assessment. The variables that make up this construct are: Costs, Traceability, Efficiency, Lead-time, Customer focus, Continuous improvement, Quality, Measurability, Employee satisfaction, Competitive advantage, Flexibility and Comprehensibility. This leads to the conceptual model shown in Fig. 1.

2.3 Longitudinal Research on BPM

There is an apparent scarcity of longitudinal research on BPM maturity. Only a few studies in the field of BPM maturity that used a longitudinal method were found. Larsen and Bjørn-Andersen [30] used a four-wave longitudinal case study approach, which resulted in a spiral of BPM activities. They revealed this finding through a longitudinal evaluation of BPM activities at a Danish manufacturing firm. Benner and Tushman [31] investigated the photography and paint industries from a larger perspective. Their findings indicate that initiatives aimed at enhancing process management outperform

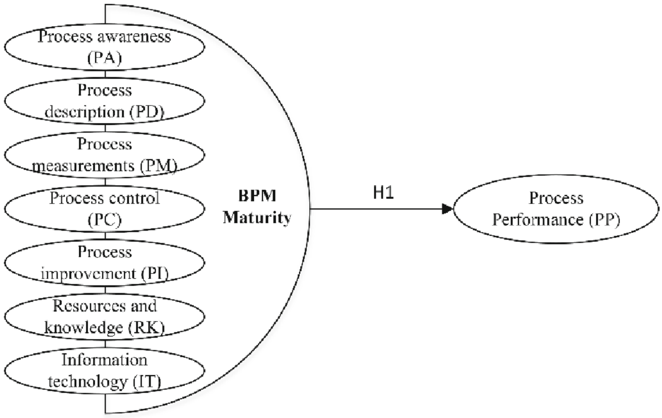


Fig. 1. Conceptual model.

technology breakthroughs in this business. More recent longitudinal maturity assessments have concentrated on specific BPM events such as Business Process Outsourcing [32], Lean Management [33], and Business Process Orientation [34]. However, all of these studies have a limited scope of BPM, are primarily focused on product organizations with relatively stable processes, and do not explore the link between BPM maturity and organizational process performance.

3 Method

Although complex longitudinal data may be easily collected and analyzed thanks to modern technology [35], there are surprisingly few studies utilizing longitudinal research. This is especially true for PLS-SEM research [36].

In this study, such a design is used in accordance with Roemer's criteria [36]. In addition, the understanding of longitudinal by Ployhart and Vandenberg is followed [37]. This means that at least three data waves are required for the exact same construct. In this study, data from six waves is employed, which provides adequate data to grasp the natural oscillation of the concept of interest in this study [38].

3.1 Data Collection and Setting

In general, the Dutch public sector has transitioned from a vertical, one-way type of accountability to a process-oriented, decentralized form of accountability [39]. This trend is also seen in higher education in the Netherlands.

The Ministry of Education, Culture and Science (ECS) formulated several performance agreements with the universities, and in accordance with the ministry's strategic agenda, universities and the ministry have chosen which quality agreements will be in place to improve higher education [40–42].

As part of the transformation initiative the support processes of our case university were restructured. Additionally, the task of standardizing the business processes and promoting process-oriented functioning more extensively inside the university was given to a group of process consultants. The main objective of this initiative was to better equip the university for difficulties related to digitalization and future rapid changes. This was the perfect opportunity to answer our research question. Therefor a six-wave longitudinal field investigation is conducted at a Dutch university of applied sciences as part of its transformation to a more process-oriented organization.

Employees (researchers, teachers, and support staff) were invited to complete a digital questionnaire on an annual basis from 2015 to 2021. Figure 2 presents the timeline of our study including the primary changes and the number of respondents each year.

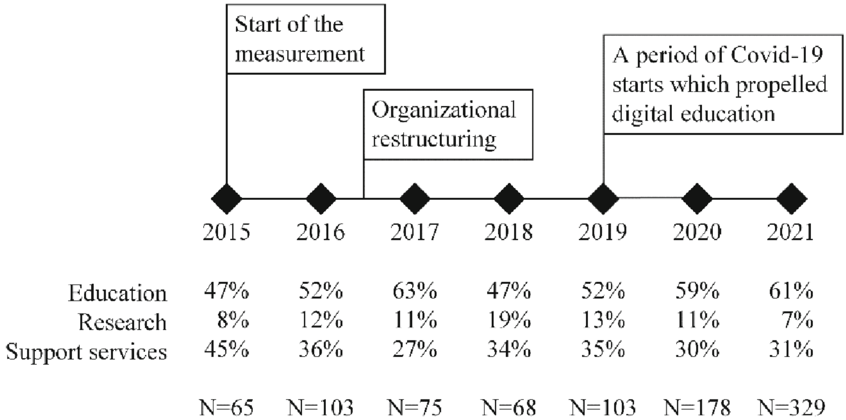


Fig. 2. Timeline of our study

3.2 Selection of Respondents

Every year in December, all workers having a link to organizational process-related duties (crossing education and curricula) were invited to participate in the survey. Throughout the first four years, the tasks that went beyond education and curricula were mostly carried out by, educational managers, members of the examination board, researchers, professors, and support services. However, there has been more collaboration between curricula in recent years, which has increased the number of invitations. By 2020, it no longer seems necessary to differentiate, so all employees have been invited to complete the survey moving forward. This resulted in a dataset of 921 completed questionnaires.

3.3 Measurements

The constructs BPM maturity and process performance are measured using the items described in paragraph 2.2. Process performance (measured with 12 items) is used as a proxy for actual performance, as has been done in other studies (e.g., [43]. Each of the BPM maturity dimensions as well as the items for process performance are scored on a 5-point Likert scale ranging from “totally disagree” to “absolutely agree”.

3.4 Analysis Techniques

Since the same indicators were measured at several points in time with different samples, this study is referred to as a repeated cross-sectional study [44]. To investigate differences in BPM maturity across time, an independent t-test was conducted on the construct score. This provides an answer to the first research question. In addition, to answer our second research question a multigroup analysis was conducted to test changes in path coefficients over time [36]. To conduct our multigroup analysis, we relied on partial least squares (PLS) path modeling in SmartPLS 4 [45]. We followed the recommendations on how to conduct multigroup analyses in PLS path modeling by Hwa, Ramayah, Memon, Chuah, and Ting [46].

4 Findings

4.1 Evaluation of Measurement Models

We first focus on evaluating our measurement models before we continue with investigating the development of BPM maturity over time. We modelled the BPM maturity dimensions and performance as a reflective construct. Typically, reflective constructs are evaluated by the internal consistency reliability, convergent validity, and discriminant validity. First, reliability is assessed by Cronbach's Alpha and composite reliability. Since the values range between 0.70 and 0.95 (Table 1) these are considered "satisfactory to good" [47]. Second, convergent validity is assessed. The metric used for evaluating a construct's convergent validity is the average variance extracted (AVE) for all items on each construct. Table 1 presents acceptable AVE's as these are above the 0.5 threshold. Thirdly, discriminant validity is assessed by means of the heterotrait-monotrait ratio of correlations (HTMT). Recent literature shows that this criterion outperforms the Fornell-Larcker criterion and the examination of cross-loadings [48]. If the HTMT value is below 0.90, discriminant validity has been established between two reflective constructs. Table 1 presents values below this threshold. Hence, values fulfill all quality criteria.

Table 1. Quality criteria for reliability and validity of the reflective measures

	PA	PD	PM	PC	PI	RK	IT	PP
PA								
PD	0.737							
PM	0.698	0.836						
PC	0.704	0.802	0.882					
PI	0.750	0.765	0.772	0.846				
RK	0.723	0.756	0.736	0.805	0.899			

(*continued*)

Table 1. (*continued*)

	PA	PD	PM	PC	PI	RK	IT	PP
IT	0.478	0.541	0.601	0.638	0.531	0.578		
PP	0.658	0.659	0.678	0.665	0.803	0.789	0.479	
Cronbach's Alpha	0.905	0.869	0.766	0.870	0.887	0.881	0.859	0.932
Composite Reliability	0.925	0.902	0.849	0.906	0.914	0.913	0.905	0.942
AVE	0.639	0.606	0.586	0.660	0.640	0.678	0.704	0.574

In line with prior research, we modelled BPM maturity as a second-order formative construct. Following the guidelines of Cenfetelli and Bassellier [49] we first examined potential collinearity issues by assessing the variance inflation factor (VIF). Table 2 presents the VIF values of the measures used, which show satisfactory values below the threshold of 5 [47]. We then assessed the measures weights and respective significance level. The weights of four BPM maturity dimensions present satisfactory significance levels while three were found to be non-significant. However, if an indicator weight is not significant, it is not necessarily interpreted as evidence of poor measurement model quality [49]. Instead, the indicator's absolute contribution to the construct is then considered. This contribution is reflected by its loadings. Hair, Hult, Ringle, and Sarstedt [50] suggests one should consider deleting the indicator when loadings show a value below 0.50, when the weight is non-significant. Since the loadings of the indicators are above this threshold, we can conclude that this measure considerably contributes to the construct. We therefore deemed that it would be prudent not to remove any of the dimensions.

Table 2. Quality criteria of the formative measures

	VIF	Weight	Significant	Loading
Process awareness	1.858	0.116	*n.s*	0.713
Process description	3.372	0.034	*n.s*	0.800
Process measurement	3.479	0.167	$p < 0.1$	0.810
Process control	3.674	−0.203	$p < 0.05$	0.762
Process improvement	3.334	0.573	$p < 0.01$	0.947
Resources and knowledge	3.074	0.369	$p < 0.01$	0.892
Information Technology	1.647	0.068	*n.s*	0.549

In addition to the above quality criteria, we furthermore need to assess measurement invariance of composite models (MICOM) over time as recommended by multigroup analyses [51]. MICOM entails a three-step process: (1) the configurational invariance assessment, (2) the establishment of compositional invariance assessment and (3) the assessment of equal means (a) and variances (b). Table 3 shows the summary of the results of the permutation tests for all the constructs. The correlation between the composite scores using the weights obtained from the various subsequent years are close to 1 and above the 5% quantile with one exception between 2017 (t_2) and 2018 (t_3) on process improvement. Thus, it can be concluded that there is compositional invariance for the most established. The differences of mean values (step 3a) and variance (step 3b) for each construct score were estimated for the years. The results revealed that for most comparisons, zero is included in the mean and variance difference confidence intervals indicating the establishment of partial measurement invariance.

Table 3. Results of invariance measurement testing using permutation

	Time	PA	PD	PM	PC	PI	RK	IT	PP
Configurational Invariance (Step 1)	$t_0 - t_1$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_1 - t_2$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_2 - t_3$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_3 - t_4$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_4 - t_5$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_5 - t_6$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Compositional Invariance (Step 2)	$t_0 - t_1$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_1 - t_2$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_2 - t_3$	Yes	Yes	Yes	Yes	**No**	Yes	Yes	Yes
	$t_3 - t_4$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_4 - t_5$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_5 - t_6$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Equal Mean Assessment (Step 3a)	$t_0 - t_1$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_1 - t_2$	Yes	**No**	Yes	Yes	Yes	Yes	Yes	Yes
	$t_2 - t_3$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_3 - t_4$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	**No**
	$t_4 - t_5$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_5 - t_6$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Equal Variance Assessment (Step 3b)	$t_0 - t_1$	Yes	Yes	**No**	Yes	Yes	Yes	**No**	Yes
	$t_1 - t_2$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_2 - t_3$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_3 - t_4$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	$t_4 - t_5$	**No**	Yes	**No**	Yes	Yes	Yes	Yes	Yes
	$t_5 - t_6$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

4.2 Evaluation of the Changes in Constructs over Time

Like other researchers (e.g., [52]), we assess the trajectory of our model (Fig. 3) to see whether other non-linear growth models might also be suitable. The BPM maturity trajectory suggests that a linear model might be prevalent since we observe no – or a negative – growth between t_1 and t_3 and a linear growth between t_3 and t_6. Process performance follows a similar line with the exception between t_4 and t_5 which shows a negative growth. The growth and decline between t_0 and t_1, and t_1 and t_2 are significant for both BPM maturity ($t = -1.415$, $p < .10$; $t = 1.488$, $p < .10$) and process performance ($t = -2.174$, $p < .10$; $t = 1.586$, $p < .10$). Similarly, the BPM maturity presents substantial growth between the last two years ($t = -1.682$, $p < .05$) as well as process performance ($t = -1.530$, $p < .10$). An exception to this harmonious relationship is between t_3 and t_4, which presents only a significant growth of process performance ($t = -3.318$, $p < .01$).

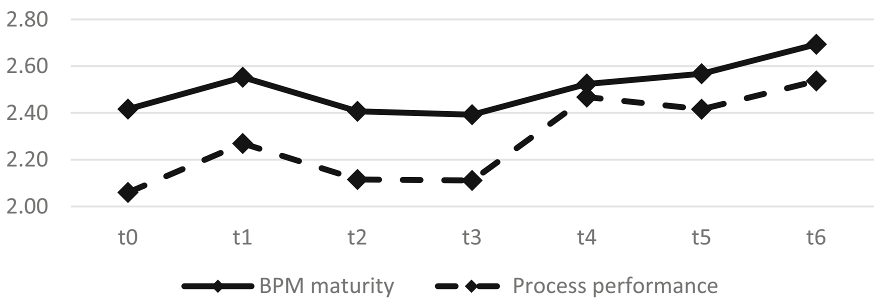


Fig. 3. The trajectory of the mean of BPM maturity and process performance over time

Additionally, independent samples t-tests are used to examine the changes in constructs over time of each of the different underlying BPM maturity dimensions. Table 4 presents the results of these calculations. The results show three significant differences. First, the dimension process description showed a significant decline between t_1 and t_2 ($t = 2.143$, $p < .05$). Second, the dimension process improvement significantly augmented in the last year ($t = -1.981$, $p < .05$) meaning that continues betterment of the processes takes a leap when other dimensions are in stable. Third, information technology has a significant improvement between t_4 and t_5 ($t = -2.303$, $p < .05$).

4.3 Evaluation of the Effects Over Time

In the case of repeated cross-sectional data, PLS path models need to be created separately. Thus, one model is created for each sample in time [36]. The tested structural models with path coefficients and its respective significance are shown in Table 5. The results present that the proposed hypothesis is significant at every timestamp. Therefore, the hypothesis is supported. This means that an increased BPM maturity leads to an improvement in process performance.

To test the significance of changes in the effects over time, this study conducted multigroup analysis with the six sets and compared the path coefficients between these

Table 4. Results of the changes in levels of the constructs

Construct	Time	M t	SD t	M t + 1	SD t + 1	Mean difference	t-value	Significant
PA	$t_0 - t_1$	2.781	0.727	2.828	0.686	−0.047	−0.422	*n.s*
	$t_1 - t_2$	2.828	0.686	2.673	0.792	0.154	1.388	*n.s*
	$t_2 - t_3$	2.673	0.792	2.739	0.810	−0.066	−0.490	*n.s*
	$t_3 - t_4$	2.739	0.810	2.709	0.761	0.030	0.248	*n.s*
	$t_4 - t_5$	2.709	0.761	2.872	0.798	−0.164	−1.609	*n.s*
	$t_5 - t_6$	2.872	0.798	2.913	0.820	−0.041	−0.479	*n.s*
PD	$t_0 - t_1$	2.515	0.685	2.571	0.710	−0.056	−0.503	*n.s*
	$\boldsymbol{t_1 - t_2}$	**2.571**	**0.710**	**2.333**	**0.760**	**0.238**	**2.143**	$\boldsymbol{p < 0.05}$
	$t_2 - t_3$	2.333	0.760	2.370	0.936	−0.037	−0.259	*n.s*
	$t_3 - t_4$	2.370	0.936	2.649	0.910	−0.279	−1.938	*n.s*
	$t_4 - t_5$	2.649	0.910	2.647	0.877	0.002	0.020	*n.s*
	$t_5 - t_6$	2.647	0.877	2.662	0.860	−0.015	−0.174	*n.s*
PM	$t_0 - t_1$	2.305	0.581	2.458	0.758	−0.154	−1.395	*n.s*
	$t_1 - t_2$	2.458	0.758	2.245	0.812	0.213	1.795	*n.s*
	$t_2 - t_3$	2.245	0.812	2.241	0.822	0.004	0.030	*n.s*
	$t_3 - t_4$	2.241	0.822	2.398	0.811	−0.157	−1.231	*n.s*
	$t_4 - t_5$	2.398	0.811	2.492	0.906	−0.094	−0.844	*n.s*
	$t_5 - t_6$	2.492	0.906	2.662	0.866	−0.170	−1.893	*n.s*
PC	$t_0 - t_1$	2.394	0.726	2.550	0.766	−0.156	−1.309	*n.s*
	$t_1 - t_2$	2.550	0.766	2.355	0.822	0.195	1.625	*n.s*
	$t_2 - t_3$	2.355	0.822	2.329	0.807	0.025	0.185	*n.s*
	$t_3 - t_4$	2.329	0.807	2.538	0.888	−0.208	−1.558	*n.s*
	$t_4 - t_5$	2.538	0.888	2.696	0.865	−0.158	−1.418	*n.s*
	$t_5 - t_6$	2.696	0.865	2.780	0.906	−0.084	−0.936	*n.s*
PI	$t_0 - t_1$	2.387	0.771	2.602	0.783	−0.215	−1.742	*n.s*
	$t_1 - t_2$	2.602	0.783	2.496	0.857	0.106	0.860	*n.s*
	$t_2 - t_3$	2.496	0.857	2.483	0.863	0.013	0.088	*n.s*
	$t_3 - t_4$	2.483	0.863	2.602	0.916	−0.119	−0.851	*n.s*
	$t_4 - t_5$	2.602	0.916	2.604	0.956	−0.002	−0.013	*n.s*
	$\boldsymbol{t_5 - t_6}$	**2.604**	**0.956**	**2.781**	**0.902**	**−0.178**	**−1.981**	$\boldsymbol{p < 0.05}$
RK	$t_0 - t_1$	2.335	0.813	2.505	0.801	−0.170	−1.334	*n.s*

(*continued*)

Table 4. (*continued*)

Construct	Time	M t	SD t	M t + 1	SD t + 1	Mean difference	t-value	Significant
	$t_1 - t_2$	2.505	0.801	2.417	0.863	0.088	0.702	*n.s*
	$t_2 - t_3$	2.417	0.863	2.467	0.806	−0.050	−0.359	*n.s*
	$t_3 - t_4$	2.467	0.806	2.573	0.782	−0.106	−0.857	*n.s*
	$t_4 - t_5$	2.573	0.782	2.618	0.886	−0.046	−0.424	*n.s*
	$t_5 - t_6$	2.618	0.886	2.701	0.902	−0.083	−0.937	*n.s*
IT	$t_0 - t_1$	2.200	0.697	2.355	0.852	−0.155	−1.230	*n.s*
	$t_1 - t_2$	2.355	0.852	2.330	0.817	0.026	0.201	*n.s*
	$t_2 - t_3$	2.330	0.817	2.120	0.722	0.210	1.621	*n.s*
	$t_3 - t_4$	2.120	0.722	2.200	0.834	−0.080	−0.647	*n.s*
	$t_4 - t_5$	**2.200**	**0.834**	**2.464**	**0.931**	**−0.265**	**−2.303**	**$p < 0.05$**
	$t_5 - t_6$	2.464	0.931	2.474	0.920	−0.010	−0.104	*n.s*

Table 5. Results of the test of significance of the direct effects

Time	Relationship	N	Path Coefficient	*t*-value	Significant
t_0	BPM → Process performance	65	0.641	10.938	$p < 0.01$
t_1	BPM → Process performance	103	0.673	9.476	$p < 0.01$
t_2	BPM → Process performance	75	0.800	17.864	$p < 0.01$
t_3	BPM → Process performance	68	0.814	21.959	$p < 0.01$
t_4	BPM → Process performance	103	0.680	12.159	$p < 0.01$
t_5	BPM → Process performance	178	0.810	28.074	$p < 0.01$
t_6	BPM → Process performance	329	0.723	20.714	$p < 0.01$

six time points. Table 6 presents three significant differences in path coefficients. The effect of BPM maturity on process performance starts to fluctuate significantly after four years of slow increasement. After t_3, the relation between BPM maturity and process performance descended significantly after which it increased significantly the year after. Although less than before, in the final year (t_5–t_6) the path coefficient declined again.

Table 6. Results of the multigroup analysis

Time	Path coefficient t	Path coefficient t + 1	CIs (Bias corrected)	Path coefficient differences	Significant
$t_0 - t_1$	0.641	0.673	[0.501, 0.736]	−0.032	*n.s*
$t_1 - t_2$	0.673	0.800	[0.514, 0.792]	−0.126	*n.s*
$t_2 - t_3$	0.800	0.814	[0.689, 0.872]	−0.015	*n.s*
$t_3 - t_4$	0.814	0.680	[0.719, 0.872]	0.134	$p < 0.05$
$t_4 - t_5$	0.680	0.810	[0.536, 0.772]	−0.130	$p < 0.05$
$t_5 - t_6$	0.810	0.723	[0.743, 0.860]	0.087	$p < 0.1$

5 Discussion

5.1 Implications to Theory and Practice

The results of our study provide several important implications to both theory and practice. Overall, we observed a significant drop in BPM maturity between t_1 and t_2. This result coincided with the organizational transformation in which the hierarchical structure was changed into a more process-oriented organization. Moreover, the results suggest an effect on the enacted BPM practices by digitalization. BPM maturity and process performance have increased substantially since the start of Covid-19 (after t_4 effects measured in t_5), when the organization was forced to make all courses available online. Although scholars stipulate the importance of context in successful BPM implementation [53], the context of organizational dynamics that unfold over time are often neglected. The results in this study call on recognizing temporality as a contextual factor as the complexity of an organization and its dynamics affect the dynamics of business processes that are performed.

In more detail we observed that the dimension *Process description* decreased significantly during organizational restructuring (between t_1-t_2). This could indicate that the restructuring resulted in an unclear division of labor, which resulted in a call for new working arrangements in the form of process descriptions. The BPM maturity dimensions also indicate what was required in the second disruptive event, t_4-t_6 (Covid-19). There was an immediate need for improving IT resources during this period. This seems to provide empirical evidence that the dynamics emerging from digitalization defies the established logics of BPM [54]. Although it is impossible to anticipate how a process will be performed in the future [55], it does mean that management faces an ongoing gap about how the process is doing over time [56]. This ongoing interaction between digitalization and BPM also strengthens the importance to implement an organizational culture that fosters the continuous exploration of innovation opportunities [19] as innovations can be used to improve BPM maturity. Though this doesn't mean that the performance will increase likewise.

Finally, we observed that over the last two years the demand for *process improvement* has significantly increased (Table 4). This might be because of the increasing amount of digitization, fueled by the digital transition related to Covid-19, which caused the as-is processes to be no longer deemed fit the changing organizational context (e.g. online teaching and working from home). This is in line with most well-known BPM lifecycles [57].

So, all of the aforementioned significant findings can be connected to crucial events that had an organizational strategy-level impact. To determine if the emphasis is or has been on the appropriate BPM capabilities, related to organizational dynamics, the BPM maturity model is thus appropriate for usage at this level within an organization. Because of how we have used it, the BPM maturity model is less suitable for use at the operational level. This is primarily due to the fact that we did not discriminate between the many procedures that our respondents are participating in. We observe that the number of respondents rises annually. This makes it possible to use more differentiation in the future, which might also provide better understanding of how specific processes change over time.

5.2 Limitations and Further Research

This study is not without limitations. First, a single case organization serves as the foundation for the research findings. Since there are not many differences between BPM in a university and the management of business processes in an organization, we do not expect BPM maturity development to be substantially different in these two contexts. However, one should be cautious about generalizing the findings to other settings. Future research should examine whether our findings hold across different types of organizations.

Second, although this study suggests an influence of the digital organizational evolution on organizational practices and business processes, this is not empirically examined in our study. Although, recent studies present a direct relationship between BPM and digital innovation (e.g., [58]), hitherto there is a lack of longitudinal research that takes into account the complexity and (digital) dynamics of an organization over time. Hence, an interesting avenue of research is to empirically investigate the role of digital transformation and its relation to BPM for a longer period.

Third, related to the used method, the design of a repeated cross-sectional study implies that responses at the different points in time cannot be traced back to the individual employee. It is thus unclear whether one employee has taken part in multiple surveys and what his/her responses have been. An analysis at the individual level, which can be taken up by future studies, could provide more rich insights in the individual evolution of the employees over time.

Fourth, we are aware that qualitative information may aid in a more accurate interpretation of the findings. As a result, we have spoken with support staff, researchers and teachers and have gotten thoughtful responses to our request to complete the questionnaire. These responses, while acknowledging our findings, were not documented or analyzed. Therefore, any qualitative information is excluded. It could be of use to collect these data in the future so that the findings would also be beneficial.

6 Conclusion

In this study we addressed the research question: How do changes in BPM maturity affect higher education process performance over time? The findings indicate that BPM maturity is affected strongly by organizational dynamics. BPM maturity diminishes with organization restructuring, but the differences in average BPM maturity scores are not significant over this period. In greater detail, one of the dimensions of BPM maturity (p*rocess description*) has been significantly reduced. This is consistent with what happened, because organizational restructuring results in new ways of working, rendering old descriptions obsolete.

The discrepancies in BPM maturity over the last three years are significant. During this phase, BPM maturity increases. That was also the moment when Covid-19 forced the organization to completely switch to digital education. During this time, the element of information technology was significantly improved first, followed by the element of process innovation the following year. On these grounds, it is concluded that BPM maturity grows more firmly in the context of digitalization than in the context of an organizational restructuring.

When examining how these changes in BPM maturity affect process performance, the samples reveal a positive association. Each year's samples demonstrate a significant positive relation between BPM maturity and process performance. Although the changes are minor (± 0.13), the strength of this association varies from year to year, and the variances are significant when looking at the last three years. As a result, we conclude that BPM maturity has a positive effect on process performance in both the short and long term.

Given that no longitudinal qualitative research have been found in the direction of BPM maturity, these findings are complimentary within the field of BPM. The findings shed light on how organizational dynamics affect the development of BPM maturity within one organization. The latter is crucial for organizational management because they have to prioritize which BPM capabilities need attention.

Acknowledgements. We would like to thank all minor BIM students and colleagues who contributed to the data collection. We specifically acknowledge B. de Waal, S. Joku, and E. Heutink for their commitment and contribution to this research project.

References

1. Abdulkader, B., Magni, D., Cillo, V., Papa, A., Micera, R.: Aligning firm's value system and open innovation: a new framework of business process management beyond the business model innovation. BPM J. **26**(5), 999–1020 (2020). https://doi.org/10.1108/BPMJ-05-2020-0231
2. Lizano-Mora, H., Palos-Sánchez, P.R., Aguayo-Camacho, M.: The evolution of business process management: a bibliometric analysis. IEEE Access **9**, 51088–51105 (2021). https://doi.org/10.1109/ACCESS.2021.3066340
3. 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

4. Van Looy, A., Poels, G., Snoeck, M.: Evaluating business process maturity models. J. Assoc. Inform. Syst. **18** (2017). https://doi.org/10.17705/1jais.00460
5. Tarhan, A., Turetken, O., Reijers, H.A.: Business process maturity models: a systematic literature review. Inf. Softw. Technol. **75**, 122–134 (2016). https://doi.org/10.1016/j.infsof.2016.01.010
6. 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
7. Van Looy, A., Trkman, P., Clarysse, E.: A configuration taxonomy of business process orientation. Bus. Inf. Syst. Eng. **64**(2), 133–147 (2021). https://doi.org/10.1007/s12599-021-00700-4
8. De Waal, B.M.E., Joku, S., Ravesteyn, P.: Do differences between managers and employees matter? A case study on BPM maturity and process performance. In: Ndaba, Z., Mokoteli, T. (eds.) Proceedings of the 5th ICMLG 2017, 16–17 March, Johannesburg, South Africa, pp. 101–110 (2017)
9. Matkovic, P., Pavlićević, V., Tumbas, P.: Assessment of business process maturity in higher education. In: 11th International Technology, Education and Development Conference, INTED2017 Proceedings, 6–8 March, 2017, IATED, Valencia, pp. 6891–6898 (2017). https://doi.org/10.21125/inted.2017.1600
10. Kahloun, F., Ayachi-Ghannouchi, S.: A prototype for continuous improvement of processes and their results in the field of higher education. Bus. Process. Manag. J. **26**, 168–190 (2019). https://doi.org/10.1108/BPMJ-05-2018-0148
11. Harmsen, R., Braband, G.: The OECD and higher education policy: agenda-setting, organizational dynamics and the construction of convening authority. High Educ Policy **34**, 582–602 (2021). https://doi.org/10.1057/s41307-019-00153-1
12. Goni, J.I.C., Van Looy, A.: Process innovation capability in less-structured business processes: a systematic literature review. Bus. Process. Manag. J. **28**, 557–584 (2022). https://doi.org/10.1108/BPMJ-07-2021-0487
13. Ongena, G., Ravesteijn, P.: Business process management maturity and performance: a multi group analysis of sectors and organization sizes. Bus. Process. Manag. J. **26**, 132–149 (2019). https://doi.org/10.1108/BPMJ-08-2018-0224
14. Rosemann, M., de Bruin, T., Hueffner, T.: A model for business process management maturity. ACIS 2004 Proc. **7** (2004)
15. Ravesteijn, P.: Factors and competences for business process management systems implementation. University of Utrecht, Utrecht (2011)
16. Harmon, P.:. The scope and evolution of business process management. In: Brocke, J.v., Rosemann, M. (eds.) Handbook on Business Process Management 1. IHIS, pp. 37–81. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-00416-2_3
17. Chountalas, P.T., Lagodimos, A.G.: Paradigms in business process management specifications: a critical overview. Bus. Process. Manag. J. **25**, 1040–1069 (2018). https://doi.org/10.1108/BPMJ-01-2018-0023
18. Gudelj, M., Delic, M., Kuzmanovic, B., Tesic, Z., Tasic, N.: Business process management model as an approach to process orientation. Int. J. Simul. Model. **20**, 255–266 (2021). https://doi.org/10.2507/IJSIMM20-2-554
19. Grisold, T., Brocke, J. vom, Gross, S., Mendling, J., Röglinger, M., Stelzl, K.: Digital innovation and business process management: Opportunities and challenges as perceived by practitioners. Commun. Assoc. Inform. Syst. **49** (2021). https://doi.org/10.17705/1CAIS.04927
20. vom Brocke, J., et al.: Process science: the interdisciplinary study of continuous change (2021). https://doi.org/10.2139/ssrn.3916817
21. Mendling, J.: From scientific process management to process science: Towards an empirical research agenda for business process management. In: ZEUS (2016)

22. Humphrey, W.S.: Characterizing the software process: a maturity framework. IEEE Softw. **5**, 73–79 (1988). https://doi.org/10.1109/52.2014
23. Ravesteijn, P., Zoet, M., Spekschoor, J., Loggen, R.: Is there dependence between process maturity and process performance?. Commun. IIMA **12**(2), Article 5 (2012). https://doi.org/10.58729/1941-6687.1188
24. Rosemann, M., de Bruin, T.: Towards a business process management maturity model. In: Proceedings of the 13th European Conference on Information Systems, pp. 521–532 (2005)
25. Loggen, R., Havenith, M., Spekschoor, J., Versendaal, J., Ravesteijn, P.: Business process management in Nederland - 2011. Capgemini, Utrecht (2011)
26. Janssen, K., Ravesteijn, P.: Business processes management in the Netherlands and Portugal: the effect of BPM maturity on BPM performance. J. Int. Technol. Inform. Manage. **24** (2015). https://doi.org/10.58729/1941-6679.1034
27. Froger, M., Bénaben, F., Truptil, S., Boissel-Dallier, N.: A non-linear business process management maturity framework to apprehend future challenges. Int. J. Inf. Manage. **49**, 290–300 (2019). https://doi.org/10.1016/j.ijinfomgt.2019.05.013
28. Hüffner, T.: The BPM Maturity Model: Towards a Framework for Assessing the Business Process Management Maturity of Organisations. GRIN Verlag, Norderstedt (2007). ISBN:978-3-638-73691-6
29. Rudden, J.: Making the case for BPM: a benefits checklist (2007). www.bptrends.com
30. Larsen, M.H., Bjorn-Andersen, N.: From reengineering to process management-a longitudinal study of BPR in a Danish manufacturing company. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences, p. 13. IEEE Computer Society, Maui, HI, USA (2001)
31. Benner, M.J., Tushman, M.: Process management and technological innovation: a longitudinal study of the photography and paint industries. Adm. Sci. Q. **47**, 676–707 (2002). https://doi.org/10.2307/3094913
32. Holkeri, J.: Outsourcing in aviation technical and information technology services: a comparative study of maturity development. JIEM. **15**, 367 (2022). https://doi.org/10.3926/jiem.3697
33. Jayaraman, R.: A longitudinal study of maturity of Lean management in Indian manufacturing using exploratory factor analysis and confirmatory factor analysis. J. Appl. Bus. Econ. **22**, 231–262 (2020)
34. Christiansson, M.-T., Rentzhog, O.: Lessons from the "BPO journey" in a public housing company: toward a strategy for BPO. Bus. Process. Manag. J. **26**, 373–404 (2019). https://doi.org/10.1108/BPMJ-04-2017-0091
35. Hassett, M.E., Paavilainen-Mäntymäki, E.: Longitudinal research in organizations: an introduction. In: Handbook of Longitudinal Research Methods in Organisation and Business Studies, pp. 1–22. Edward Elgar Publishing, Cheltenham (2013)
36. Roemer, E.: A tutorial on the use of PLS path modeling in longitudinal studies. Ind. Manag. Data Syst. **116**, 1901–1921 (2016). https://doi.org/10.1108/IMDS-07-2015-0317
37. Ployhart, R.E., Vandenberg, R.J.: Longitudinal research: the theory, design, and analysis of change. J. Manag. **36**, 94–120 (2010). https://doi.org/10.1177/0149206309352110
38. Kehr, F., Kowatsch, T.: Quantitative longitudinal research: a review of IS literature, and a set of methodological guidelines, vol. 18 (2015)
39. Agostino, D., Saliterer, I., Steccolini, I.: Digitalization, accounting and accountability: a literature review and reflections on future research in public services. Finan. Account. Manage. **38**, 152–176 (2022). https://doi.org/10.1111/faam.12301
40. Ministry of ECS: Houdbaar voor de toekomst: Strategische agenda hoger onderwijs en onderzoek (2019). https://www.rijksoverheid.nl/documenten/publicaties/2019/12/02/strategische-agenda-hoger-onderwijs-en-onderzoek

41. Ministry of ECS: De waarde(n) van weten: Strategische agenda Hoger Onderwijs en Onderzoek 2015–2025 (2015). https://www.rijksoverheid.nl/documenten/beleidsnota-s/2015/07/07/de-waarde-n-van-weten-strategische-agenda-hoger-onderwijs-en-onderzoek-2015-2025
42. Ministry of ECS: Kwaliteit in verscheidenheid: Strategische agenda Hoger Onderwijs, Onderzoek en Wetenschap (2011)
43. Škrinjar, R., Trkman, P.: Increasing process orientation with business process management: critical practices. Int. J. Inf. Manage. **33**, 48–60 (2013). https://doi.org/10.1016/j.ijinfomgt.2012.05.011
44. Frees, E.W.: Longitudinal and Panel Data: Analysis and Applications in the Social Sciences. Cabridge University Press, Cabridge (2004)
45. Ringle, C.M., Wende, S., Becker, J.-M.: SmartPLS 4 (2022). https://www.smartpls.com
46. Hwa, C., Ramayah, T., Memon, M., Chuah, F., Ting, H.: Multigroup analysis using smartPLS: step-by-step guidelines for business research. Asian J. Bus. Res. **10**, 1–19 (2020). https://doi.org/10.14707/ajbr.200087
47. Hair, J.F., Risher, J.J., Sarstedt, M., Ringle, C.M.: When to use and how to report the results of PLS-SEM. Eur. Bus. Rev. **31**, 2–24 (2019). https://doi.org/10.1108/EBR-11-2018-0203
48. Henseler, J., Ringle, C.M., Sarstedt, M.: A new criterion for assessing discriminant validity in variance-based structural equation modeling. J. Acad. Mark. Sci. **43**(1), 115–135 (2014). https://doi.org/10.1007/s11747-014-0403-8
49. Cenfetelli, R.T., Bassellier, G.: Interpretation of formative measurement in information systems research. MIS Q. **33**, 689–707 (2009). https://doi.org/10.2307/20650323
50. Hair, J.F., Hult, G.T.M., Ringle, C.M., Sarstedt, M. (eds.): A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM). Sage, Los Angeles (2017)
51. Henseler, J., Ringle, C.M., Sarstedt, M.: Testing measurement invariance of composites using partial least squares. Int. Mark. Rev. **33**, 405–431 (2016). https://doi.org/10.1108/IMR-09-2014-0304
52. Söllner, M., Pavlou, P.: A longitudinal perspective on trust in IT artefacts. Presented at the European Conference on Information Systems (ECIS), Instanbul, Turkey June 15 (2016)
53. vom Brocke, J., Baier, M.-S., Schmiedel, T., Stelzl, K., Röglinger, M., Wehking, C.: Context-aware business process management. Bus. Inf. Syst. Eng. **63**(5), 533–550 (2021). https://doi.org/10.1007/s12599-021-00685-0
54. Baiyere, A., Salmela, H., Tapanainen, T.: Digital transformation and the new logics of business process management. Eur. J. Inf. Syst. **29**, 238–259 (2020). https://doi.org/10.1080/0960085X.2020.1718007
55. 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
56. Pentland, B., Liu, P., Kremser, W., Hærem, T.: The dynamics of drift in digitized processes. Manag. Inf. Syst. Q. **44**, 19–47 (2020)
57. Macedo de Morais, R., Kazan, S., Inês Dallavalle de Pádua, S., Lucirton Costa, A.: An analysis of BPM lifecycles: from a literature review to a framework proposal. Bus. Process Manage. J. **20**, 412–432 (2014). https://doi.org/10.1108/BPMJ-03-2013-0035
58. Van Looy, A.: A quantitative and qualitative study of the link between business process management and digital innovation. Inform. Manage. **58**, 103413 (2021). https://doi.org/10.1016/j.im.2020.103413

From Automatic Workaround Detection to Process Improvement: A Case Study

Nesi Outmazgin[1](✉), Wouter van der Waal[2], Iris Beerepoot[2], Irit Hadar[1], Inge van de Weerd[2], and Pnina Soffer[1]

[1] University of Haifa, Mount Carmel, 3498838 Haifa, Israel
nesi@zefat.ac.il, hadari@haifa.ac.il, spnina@is.haifa.ac.il
[2] Utrecht University, Utrecht, The Netherlands
{w.g.vanderwaal,i.m.beerepoot,g.c.vandeweerd}@uu.nl

Abstract. The improvement of business processes through learning and investigating workarounds has attracted research attention in recent years. Workarounds can be considered as a symptom of needed process improvements but adopting them does not necessarily lead to an appropriate one. Hence, identifying and understanding the underlying problems or perceived barriers that motivate workarounds is essential for suggesting an appropriate process improvement solution. In this paper, we propose a streamlined end-to-end approach that attempts to leverage workarounds to improve processes. This approach is based on two pillars: (1) a semi-automated workarounds detection by using the SWORD framework, which consists of twenty-two patterns to detect workarounds from events logs. (2) workarounds investigation and analysis using a motivational model that serves to reveal problems that lie under the identified workarounds. This analysis contributes toward proposing tailored and targeted process improvements. We report on an industrial case study that demonstrates the proposed approach, from workaround detection to proposing tailored process improvements. The improvements have been accepted by the organization and are currently being implemented.

Keywords: Business process improvements · Workarounds · Automatic detection · Motivational analysis · Event logs · Case study

1 Introduction

The quest to systematically improve business processes has been ongoing for several decades, including the development of methods, techniques, conceptual frameworks, and models for process improvements and redesign projects [1, 2, 3]. A fruitful way towards process improvement is to use the knowledge of workarounds as sources of innovation [4, 5]. Most suggestions made in this direction so far propose to improve processes by adopting the workaround as an official practice [6–8]. This, however, is not always a good solution, as workarounds may entail risks and favor specific goals over others, which are not necessarily of lesser importance [9–11]. Some approaches suggest techniques for investigating workarounds and analyzing their actual impact on

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 372–390, 2023.
https://doi.org/10.1007/978-3-031-41623-1_22

the process, as a possible basis for evolving the process and improving its design [12, 13]. Additional suggestions provide generic actions that can be taken upon detection of workarounds [14].

A recent step towards using workarounds for process improvement is the workaround motivational model [15] based on the Theory of Planned Behavior (TPB) [16], that comprehensively explains the motivation for workarounds. According to this explanation, workaround motivation stems from conflicts and misalignments among goals or with respect to the official process [15]. Furthermore, workarounds are executed when enabled by managerial, social, and technological factors in the organizational reality. Process improvement directions can then rely on this analysis and aim at resolving the identified conflicts while reducing the enabling situational factors. This approach as well as others relate to known workarounds, i.e., workarounds that have already been identified before process analysis, mostly through interviews and observations.

However, the use of qualitative methods is labor-intensive, and process participants may not disclose their workarounds when they are aware of being observed [17]. For a practically applicable workaround-based improvement, a holistic end-to-end method that would encompass all the steps from automatic identification and quantification of workarounds to indication of process improvement possibilities is needed. Such a method has not been proposed so far.

The use of process mining techniques for workaround detection has already been proposed, initially based on a predefined set of patterns, limited in the types of workarounds that could be detected [18]. A recent attempt to bridge this gap by detecting various types of deviations that may reflect workarounds is the SWORD framework, which is a semi-automated detection approach that uses 22 patterns to identify potential workarounds in event logs. Whether any pattern can be used in a particular situation is dependent on the characteristics of the data in the event log at hand [19]. This framework, therefore, provides good support for detection and quantification of workarounds.

This paper introduces such a holistic approach and demonstrates it via a case study. The approach has two main pillars: (a) the SWORD framework [19] for workaround discovery from event logs, which serves as an initial identification of workarounds that take place in a process, and (b) the TPB-based motivational model of workarounds [15], which supports the analysis of the conflicts that motivate workarounds as well as their enabling factors.

These can finally be targeted by proposed process improvement solutions. The steps of the proposed approach are demonstrated and discussed through a case study, from workarounds detection to actual process improvements in the organizational setting.

2 Background

This section presents the foundations that underlie our proposed method. Specifically, we elaborate on the SWORD framework, used for automatic workaround detection, and the workaround motivational model, used for explaining workaround motivations.

2.1 The SWORD Framework

The SWORD framework allows for the detection of workarounds without prior knowledge, i.e., avoiding the need to perform observations or interviews [19]. It consists of twenty-two patterns that describe differences between traces in event data. These differences are split over four different perspectives that may be considered during workaround detection [20, 21]: Control-flow, Data, Resource, and Time.

The *Control-Flow* perspective describes patterns that relate to activity order or frequency. For example, an activity may be skipped completely in rare cases. Patterns in the *Data* perspective monitor data fields. For example, information may have to be registered using specific forms, but workers may feel like they need more flexibility and decide to register it in free-text fields. The *Resource* perspective is focused on the specific workers involved in a trace. Worker 1 may be dependent on work from someone else. If this is not finished in time, worker 1 may decide to do their work for them. While this solves their immediate problem, they may not be officially authorized for the task. Finally, the *Time* perspective contains patterns that are concerned with when activities are executed. For example, a trace may usually take a day, but there can be rare cases where the task takes a week to finish. This longer trace duration may indicate that workers are delaying finalizing the registration of the task. There can be multiple reasons for such a delay. In some cases, it is more time-efficient to wait, so that multiple registrations can be finished at the same time. In other cases, it may be advantageous to delay registration until the next month for KPI values for certain businesses. The delay may also be an error due to a worker who forgot the registration. Whether such a delay would be considered a workaround depends highly on the domain and scenario and should therefore be evaluated by an expert before concluding whether the indicated trace is a workaround or not.

For application of the framework, it is important to note that the various patterns have different data requirements. This means that not all patterns can be applied to a given dataset. For example, if we investigate the duration of a trace, only timestamps for the events are required, but if we check which resource types executed a certain event, we need both activity names and the resource type that executed it. In order to apply the framework, we first determine which patterns can be applied to the data following the data requirements in [19], then we only apply those patterns, and finally, we let a domain expert evaluate the traces that are indicated by the patterns to determine if it is a workaround, rare normative behavior, an error, or anything else.

2.2 The Workaround Motivational Model

The workaround motivational model, presented in [15], is based on the Theory of Planned Behavior (TPB) [16]. It extends this theory with elements that specifically explain the decision to work around processes. TPB aims to explain behavior as stemming from intentions that are formed from the interplay of three forces: (1) the personal attitude towards the behavior, which considers personal expectations of benefits and risks associated with the behavior, (2) the subjective norm, which is the subjective perception of how the individual should behave, and (3) the perceived behavioral control of the individual, or perceived capability to engage in the behavior.

The workaround adaptation of TPB, illustrated in Fig. 1, attempts to explain workaround intentions by refining these three forces to relevant elements and distinguishing motivating elements from enabling ones. Enabling elements make workarounds possible or easy to perform if and when a motivation for performing them exists (due to motivating elements). According to this model, workaround motivation stems from misalignments and conflicts between different parts of the subjective norm, namely, perceived organizational goals, perceived goals of the local unit (e.g., department, team), and the standard processes to be followed. In addition, these elements (together or separately) can be in conflict with personal interests (attitudes towards behavior). Enabling elements of the model include (1) poor organizational control - which makes workarounds unrisky for the individual and affect the attitude towards them, (2) workarounds supportive atmosphere, (3) unclarity of expectations - which affect the subjective norm regarding workarounds, and (4) the existence of workaround opportunities (e.g., related to the process definition or to its support systems), which make them possible.

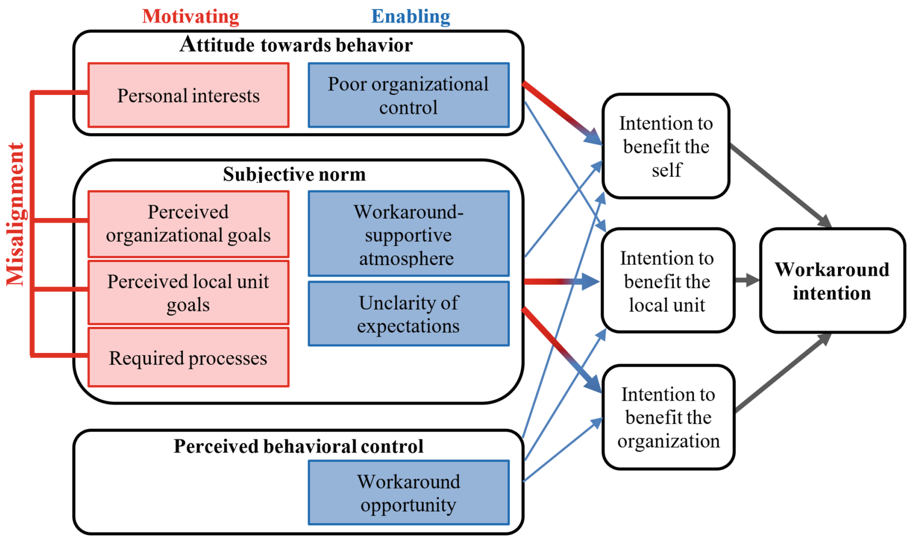


Fig. 1. The workaround motivational model.

In summary, according to the motivational model, workarounds are motivated by conflicts among perceived goals, process requirements, and/or personal interests, and are enabled by a combination of managerial, social, process-related, and technological issues. When a workaround is known to exist, analyzing the situation to identify the specific elements of the model that are relevant is a first step for improving the process. Improvement should then aim at resolving the identified conflicts and removing or reducing the effect of the enabling elements.

3 The Proposed Approach

In an effort to combine the SWORD workaround mining framework and the TPB-based workaround motivational model, we propose the approach outlined in Fig. 2. As process mining is a major element of our approach and our ultimate goal is to achieve process improvement, the existing PM^2 methodology [22] provides a logical skeleton for our method. We describe the steps as well as similarities and differences to the original methodology below.

Similar to PM^2, the approach starts with planning and extraction. Here, a process is chosen, and possible questions are defined. Initial data is collected, such as process documentation and event data for the process of study. After extraction, PM^2 prescribes that different analysis iterations are completed. In the context of workaround detection and analysis, we distinguish two types of analyses: (1) workaround mining and (2) motivational analysis. Workaround mining can take place once or multiple times. It consists of three steps, equal to the ones proposed in PM^2: data processing, mining & analysis, and evaluation. After the evaluation step, new data may be extracted. Once workaround mining is completed, the motivational analysis starts. This is done by performing interviews with domain experts. The interview data is processed so that the information can be mapped to the TPB model and verified in an evaluation. We expand the step of process improvement as opposed to the original PM^2 method, to include the development of suggestions as well as a systematic assessment of the process improvement suggestions.

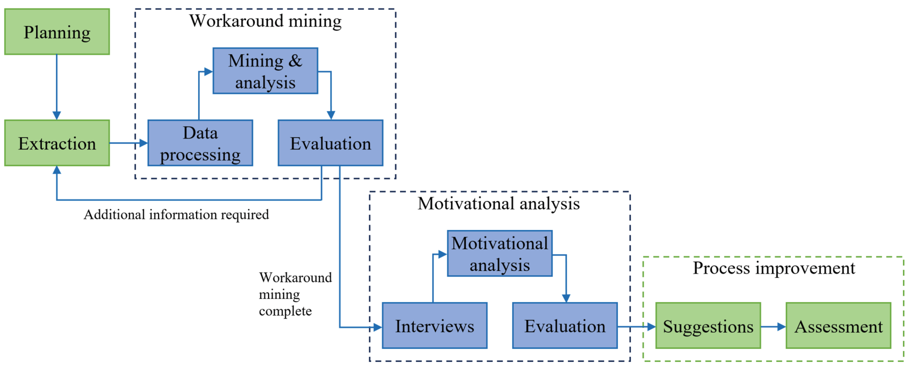


Fig. 2. The proposed approach from automatic workaround detection to process improvement.

4 Case Study

This section reports a case study, where the approach outlined in the previous section was applied. Through a combination of SWORD analysis and the TPB-based motivational model, we provide an illustration of the applicability of the end-to-end approach from workaround detection to process improvement.

4.1 Setting

The case study was performed in a public academic college with over 600 employees. As a government-funded organization, it must comply with government regulations in all its processes, especially with the processes related to purchasing, budget management, and academic administration. In this case study, the purchase requisition process was studied. It is one of the central processes, managing all the purchases for all the departments and faculties. Every purchase made in the organization, regardless of its amount and type, must be examined and go through rounds of approvals. This process is annually audited by an external auditor, who reports to the director of the organization and to the state auditor. Hence, non-compliance with any regulation or law may expose the organization, its management, and even its employees to lawsuits.

The process description is as follows. For purchasing goods and services, each department has a yearly budget that is based on a yearly work plan. To accomplish the purchases in an orderly and controlled manner, an automated process that includes several approval rounds is followed using the ERP system. The process starts by initiators who create and update the purchase requisitions in the system, then the requisition goes through the approval rounds, including the department manager, the buyer, CFO, CEO, and the Director (depending on several conditions). Each approver examines the purchase requisition according to relevant business rules, and can approve, cancel, or return the requisition back for more information. These approval rounds are iterated until the requisition is fully approved.

4.2 Procedure

The study was conducted along the following steps.

1. **Initial data collection:** this included obtaining a high-level description of the process from the process owner, collecting existing documentation (e.g., ISO work procedures), studying the relevant functionality and user interface of the information system, and obtained event logs that cover the past two years.
2. **Analysis of the process logs using the SWORD framework:** we selected the applicable patterns for the elicited event logs and used them to discover traces that deviated from the norm in various ways.
3. **Assessment of the SWORD results:** the results of the SWORD analysis were shown to the process owner and discussed. The goals were (a) to assess which of the identified patterns could indicate workarounds, (b) to prioritize further analysis, and (c) to elicit additional information about the identified workarounds. In particular, we were referred to relevant employees who were involved in the workarounds.

4. **Semi-structured interviews:** we conducted semi-structured interviews with eight employees of different roles, to whom we were referred. The interviews focused on (a) the process as viewed by the interviewees, (b) perceived process goals at the organizational and departmental level, and the extent to which these goals were aligned with each other, and (c) the discovered workarounds, seeking to understand how and why these were performed, and management response if any. The interviews took 45–90 min each, and were conducted in the offices of the organization. All the interviews were audio recorded and transcribed. Later on, complementary phone calls and emails were made, seeking additional explanations and validation.
5. **Motivational analysis based on the interviews:** in this step, we followed a deductive coding approach [23], where the motivational model (Fig. 1) served as a basis for analyzing the interview text. For the motivating elements, conflicts among perceived goals and between the process and perceived goals were analyzed using goal models (as described in [15]). For the enabling elements, we looked for statements that could indicate manifestations of such. For example, "this is done by everyone" was considered as an indication of a supportive social atmosphere. When specific features of the information system were indicated (e.g., a possibility to approve several requisitions as a batch), we validated the existence (or absence) of these features in the system.
6. **Process improvement suggestions:** we suggested process improvement directions aimed at resolving the issues identified by the motivational analysis. In particular, the solutions were aimed to reduce the workaround motivation by resolving identified conflicts between the process and perceived goals. Personal interests were addressed by modifications in the reward system. We further suggested ways for removing or reducing the effect of enabling factors – specifically those related to system functionality and organizational control.
7. **Assessment of suggested improvements by the organization:** the suggested improvements were presented in a meeting with the process owner and several process stakeholders, the IT manager, and an external consultant specializing in purchasing processes. First, we started presented the SWORD results and the motivational analysis. We explained how the proposed improvements would address the identified problems, and how they could be implemented in the organization. As a result, the management decided to implement the suggested improvements with slight adjustments adapted to the organizational atmosphere.

In the next sections, we provide the results of the workaround mining, motivational analysis, and process improvement phases, following the steps outlined in Fig. 2.

4.3 Workaround Mining

After planning and extraction, we possessed event logs of the purchase process for a period of two years. The log included 5,908 completed cases and 38,333 events.

The available event log followed a case (requisition) focus; The activities related to each case were available, together with the corresponding timestamps and (pseudonymized) resources. Based on this available data, as well as the process description, we decided to apply the patterns following this focus and searching for deviations between cases. The event log contained columns with a dedicated Case ID, an activity name, a timestamp, and a (pseudonymized) resource ID, but no corresponding resource roles. Based on this information combined with the data-requirements as described in [19], we determined that we could apply the following SWORD patterns: "Occurrence of directly repeating activity", "Frequent occurrence of activity", "Number of resources out of bounds", "Occurrence of activity outside of time period", "Delay between start of trace and activity is out of bounds", "Time between activities out of bounds", and "Duration of trace is out of bounds".

Each pattern applied to the event log ranked the traces in a unique way, where the top-ranked traces were the most likely workaround candidates. Since every trace is assigned a Z-score, we needed to determine a threshold to determine which patterns could be considered a likely workaround. As there was no available guideline for this, we investigated both a Z-score of 2 and 3, where 3 is traditionally a rather conservative value when evaluating Z-scores. An overview of the number of "interesting" traces can be seen in Table 1.

Since the numbers varied strongly between patterns and there were far too many traces even with the conservative measure, we evaluated the top three traces for each pattern instead with a domain expert to determine if the pattern led to workarounds in this context.

After talking with the expert, it turned out that two of the eight patterns could indeed point to a workaround: "Frequent occurrence of activity" and "Occurrence of activity outside of time period". Specifically, a trace where a case is "reopened" can point to a workaround. After a case is reopened, the CFO needs to reapprove it before it should be closed, but in three of the six cases where a case was reopened, it was closed without this activity occurring, which was confirmed to be a workaround.

The "Occurrence of activity outside of time period" pointed to traces where the CEO, the CFO, or the buyer approved cases at unusual times, like 2 AM. While this is not necessarily against procedures, it did point to another issue: sometimes there are more cases approved in a day than is reasonably possible. We decided to investigate this observation further by changing the perspective of our analysis.

Table 1. The number of traces that can be considered strong deviations from the norm given Z-scores of 2 and 3 for each pattern.

Pattern	# Deviation with Threshold (M + 2SD)	# Deviation with Threshold (M + 3SD)
Occurrence of directly repeating activity	37	37
Frequent occurrence of activity	1379	541
Number of resources out of bounds	621	0
Number of resources out of bounds (relative to number of activities)	774	48
Occurrence of activity outside of time period	465	73
Delay between start of trace and activity is out of bounds	297	160
Time between activities out of bounds	463	385
Duration of trace is out of bounds	310	199

While our initial analysis was from a case (requisition) perspective, investigating the behavior of resources makes more sense from a resource perspective. To do so, we constructed a new case ID by combining the resource identifier and the date, effectively seeing a full workday for each resource as a trace. To investigate how often this high frequency of approving cases occurred for a single resource, we applied the "Frequent occurrence of activity" pattern in this new perspective.

Table 2 shows the main results concerning frequencies of the approval activities. We consider any day where the number of approvals was more than two standard deviations from the mean to be deviating. This means that the CEO could approve at most 10 cases a day, and a buyer or CFO could approve 8. With this analysis we have found 181 days where the CEO approved more cases in a single day. In the most extreme case, there were 78 case approvals in a single day, which would not be possible without applying a workaround. We found similar results for the CFO and Buyer approval, where we found 236 and 231 days with a too high number of occurrences respectively. According to the expert, at most eight cases could be approved a day, so the number of deviating days for the CEO would be even higher with this standard; 204 instead of 181.

Table 2. The mean number of repetitions and standard deviation for the approval activities on any single day, as well as the suggested thresholds for considering a trace deviating from the norm and the number of deviations in the event log.

Activity	Max	Mean (daily)	Standard deviation	Threshold (M + 2SD)	# Deviations
Buyer Approval	51	0.887	3.65	8	231
CEO Approval	84	0.860	5.03	10	181
CFO Approval	53	0.873	3.80	8	236

In summary, we identified two workaround types: (a) reopening a requisition without re-approving it, and (b) reporting a large number of approvals together (probably after they have been already given manually).

4.4 Motivational Analysis

The motivational analysis followed the elements of the motivational model. Focusing on the two types of workarounds that were detected, we now present each one with the associated model elements.

Workaround A (Reopen - Update - No Additional Approval): After the purchase is approved, department initiators or even buyers reopen and update the purchase requisition without reapproving it through the regular approval rounds. The reopen activity allows one to update the purchase requisition, but any update after the purchase is approved requires a transfer back through the regular approval rounds.

Organizational goals:

a. Supervise and control all the purchases in the organization − the primary process goal.
b. Approve each purchase.
c. Achieve economic efficiency

Goals of the department:

a. Meet departmental KPI targets.
b. Provide good service.

These goals depend on the ability to provide quick and high-quality responses to the requests of the department customers

Personal interests of the initiators or department managers:

a. Get rewarded for performance.
b. Use specific products or services that are familiar and easier to handle.

Identified misalignments:

a. Local-unit goals vs. organizational goals, as implemented in the process: to support economic efficiency, the process has different approval trails for different products or based on the total cost. If a requisition is expected to require a long approval trail (due to specific products that are preferred or to general cost), this is in conflict with the local-unit goals of meeting KPIs and providing a good (and quick) service delivery. To avoid long approval rounds, a requisition is filed for a different product or smaller amounts, so approval is relatively quickly. After the approval is given, the actual quantities or products are entered. An alternative scenario is when there is actually an update (e.g., of quantity) or an error in the requisition is spotted, and the initiator wants to avoid additional time for approval, since delays in the purchase may, again, reduce the level of service provided and the departmental KPI values.
b. Personal interests vs. organizational goals, as implemented in the process: This relates to two issues. First, departmental KPI values are reflected in individual rewards, and hence meeting the KPI targets is also a personal interest of the employees. Second, requisition initiators and department managers may prefer specific products or services they are familiar with and find easy to handle. The organizational goal of economic efficiency may lead to the preference of alternative products, thus the approval may not be immediate (or may not be granted at all).

Enabling elements:

a. Workaround-supportive atmosphere: the workaround is performed by most of the initiators and department managers, who share the perception that the process is very strict, and hampers their work.
b. Poor organizational control: the process is not monitored, and no sanctions are taken against employees who work around it.
c. Workaround opportunity: it is possible (technologically) to reopen, update and close purchase requisitions without the necessity for reapproval.

Analyzing the workaround intentions shows that the departmental initiators and managers act primarily with the intention to benefit their local unit goals when they try to promote good delivery time for specific products or services through the reopen activity. This intention is supported by their social environment, as well as by poor organizational control and a lack of technological control of this option.

Workaround B (Batch Approvals): Batch reporting of approvals after they have been manually given. Approvers in the purchase department approve dozens of purchase requisitions in one day or even in one hour. This is unreasonable since each requisition requires time for examination and inquiries for additional information. A main result of this workaround is that the actual status of a requisition and its approval process are not reflected in the IS.

Organizational goals: a. Supervise and control all the purchases in the organization. b. Meet the legal regulations in the purchase process. c. Increase organizational and economic efficiency.
Goals of the purchase department: a. Increase flexibility in purchase documentation. b. Achieve economic and efficient purchasing.
Personal interests of the initiators or department managers: a. Make the purchasing process appear appropriate to auditors. b. Minimize the effort associated with approvals
Identified misalignments: a. Local-unit goals vs. business process: the process (as implemented) does not allow the flexibility required by the buyer to enter the quotes and compare them automatically by preconfigured rules. This inflexibility motivates the buyer to create parallel documentation in Word and Excel files rather than to handle requisitions via the information system. As a result, the approvers informally examine the requisitions, making inquiries and approving the requisitions via email anyway, and reporting to the IS in a post-hoc manner. b. Personal interests vs. organizational goals: to achieve the organizational goal of meeting the legal regulations in the purchase process, an external audit of the purchasing process is carried out periodically. Facing this, approvers want the process as recorded in the system to appear compliant with required procedure and entail short response times. Approving manually through emails or phone calls and reporting in retrospect, they can ensure the procedure and response times appear as they should be.
Enabling elements: a. Workaround-supportive atmosphere: the workaround is performed by all the approvers in the purchase department. b. Poor organizational control: the process is not monitored, and no organizational sanctions are taken against the approvers. c. Workaround opportunity: The information system supports the approval of requisitions as a batch in an automated procedure

Analyzing the workaround intentions shows that the motivation stems from a lack of a proper support for the approval decision making in the information system, so a parallel Excel and email-based process takes place. This process has no transparency through the information system, and eventually, in order to meet audited regulations, reports are made in the system.

4.5 Process Improvement

Based on the above analysis of motivating and enabling factors of the identified workarounds, the following process improvements were suggested. As mentioned, the improvements have largely been accepted by management and are currently implemented.

Addressing Workaround A (Reopen - Update - No Additional Approval).
Improvement 1. To address the enabling technological factors, suggest changing the process flow and its gateway conditions so the reopen activity must go back to the approvers, except for small and well-known updates that meet clear conditions. For example, allowing to decrease the quantity of the products but not to increase, removing products from the list but not adding additional ones, changing the description of the products, etc. The guiding line is to allow changes that do not involve the supplier, the goods, or an increase in the total amount of the purchase that is already approved. While these updates will be immediate, any other update will require reapproval.

Improvement 2. To address the motivational factors that are associated with the KPI targets, we suggest to adjust the KPIs that concern meeting SLA thresholds. Delay times spent waiting for other department approvals will not be considered as part of the total service time, so KPI values and personal awards will not be affected by approval times. Yet, the time taken for high-cost requisitions will remain long.

Addressing Workaround B (Batch Approvals)
To address the motivational factors, we suggested two improvements that focus on technological support for the approval process.

Improvement 1. Add internal tools to the system functionality that support comparing quotes through the process, so the approvers have the information needed for making decisions without a need for additional Excel files.

Improvement 2. Add an alternative option for communication between the approvers and the initiators that allows making quick inquiries (instant messaging) without delaying the process, and in a way that is compliant with the required procedure.

The idea behind these improvements is to give the approvers all the required tools and information to examine and approve purchases, and still remain compliant with the required procedure. Note that transparency will be increased (which may still be against personal interests, but with a less risk implied to individuals by audits).

Since the approvers are part of the organizational management, who decides about sanctions and control policies, we suggested addressing the enabling factors only after full implementation of the improvements that address the motivational ones.

5 Discussion and Lessons Learned

In this paper we contribute to the body of work that attempts to utilize workarounds for improving processes in two main ways. First, by proposing a streamlined end-to-end process, from a semi-automated detection of workarounds using the SWORD framework to proposing tailored and targeted process improvements. Second, by showing how the theoretical motivational model can serve for revealing problems that lie under the identified workarounds.

While utilizing workarounds as a source for process improvement has been suggested before (e.g., [6, 14, 19]), our approach differs from other proposals in identifying and addressing the root causes – the motivation for, and enablement of, performing workarounds. We reveal the perceived obstacles that motivate workarounds in the form of goal and process misalignments. The solutions we propose are hence not directly tied to the actual workaround, which is considered to be merely a symptom. To this end, we analyze the identified workarounds through the motivational model to understand their underlying root reasons.

Since workarounds may differ, the motivating and enabling elements are examined for each situation separately. As a result, the proposed process improvements primarily aim to reduce the misalignments, and additionally, to reduce the enabling factors. We note that addressing only enabling elements (e.g., limiting the flexibility allowed by the IS, or introducing disciplinary responses to workarounds) without addressing the motivation (namely, the underlying problems) may result in different ways of working around the unsolved problems.

The use of this process, from semi-automatic workaround detection using the SWORD framework to targeted process improvements, was found effective and led to practical solutions that were accepted by the relevant stakeholders. While performing this process, the following lessons have been learned.

Lesson 1: The SWORD framework highlights process deviations, which are not necessarily workarounds. While in this study we focused on the deviations identified as workarounds, other identified deviations can also provide valuable information and lead to improvements. For example, cases with a high number of back-and-forth transitions between activities ("Ping Pong"), or cases with an exceptionally high duration (e.g., six months and more). All these cases, while considered legitimate process behavior, are not as efficient as expected, and improvements can also target them.

Lesson 2: While the motivational analysis provides useful insights about the reasons and enablers of workarounds, it relates to workarounds that are known to exist, and requires elicitation of additional information from the involved employees. SWORD provides a good starting point for such an elicitation by systematically identifying workarounds. Furthermore, as workarounds often involve violations of organizational regulations, employees might not tend to disclose information about them and admit taking part in this behavior. When confronted with the SWORD results, they are more likely to cooperate and explain what is done and why.

Lesson 3: Motivational analysis highlights problems to be solved, not necessarily possible solutions. Yet, with clearly identified problems, focused solutions can be proposed.

Taking a broader perspective, the reported case study shows that the motivational analysis may lead to a diverse set of improvement directions, which goes far beyond the improvements that could be suggested based on merely observing the workaround. Implementing these solutions might lead to new, unanticipated, conflicts, which may motivate new forms of workarounds. Taking this into consideration, it is important to repeat workaround detection and analysis periodically and achieve an ongoing improvement cycle. As the SWORD framework is capable of semi-automatically detecting a large variety of workarounds, which may not be known a-priori, it forms an essential ingredient in this cycle.

6 Related Work

The idea of improving processes based on workarounds knowledge has already been suggested. The simplest way would be to suggest that the process can be improved by adopting the practiced workaround [6, 8, 14]. However, many studies show that workarounds may impose risks, such as reduced quality of products, financial losses, violation of privacy regulations, potential lawsuits, and more [8]. Alternative ways of improving processes based on analyzing workarounds include [24], who suggested an analysis approach based on goal modeling to highlight improvement directions. Another approach was proposed by Beerepoot et al. [25] based on a set of workarounds that were studied and analyzed. They proposed and included a set of contextual activities that can be taken upon workaround detection for improving the process. All these approaches relate to workarounds that are known to exist, but do not address the detection of workarounds. Hence, they are comparable to the motivational analysis of our proposed approach. Furthermore, none of them relies on a theoretical basis, as opposed to our motivational analysis.

Concerning workaround identification, namely, the workaround mining of our proposed approach, a few automated approaches have been proposed. Outmazgin and Soffer [18] proposed four generic patterns and showed how these can successfully be detected in an event log using process mining techniques. A designated algorithm for detecting a specific workaround pattern, the "split case" workaround, has been developed by [26]. Weinzierl et al. [17] proposed a supervised learning approach for detecting workarounds of a predefined set of patterns in event logs. The SWORD framework [19] is less restrictive in terms of the workaround patterns that are sought, and in fact, forms the first part of our proposed approach.

In summary, while various related approaches cover parts of our proposed approach, to the best of our knowledge this is the first end-to-end approach from automatic workaround detection to process improvement suggestions.

7 Conclusions

In this paper, we proposed and applied an approach that starts with a data-driven detection of workarounds and ends with proposing focused and tailored process improvements. Each detected workaround was assessed with certain organizational stakeholders. Then, the outputs of the assessments were analyzed using the motivational model for identifying two types of factors (motivating and enabling). Finally, process improvements were proposed for each workaround situation.

We found that this procedure can guide improvements to processes in a fast and targeted manner. In fact, the improvements we proposed were evaluated in the case study organization and found adequate to the extent that they are currently being implemented. This indicates the potential of the suggested procedure, which aims to leverage workarounds for process improvement by addressing their sources rather than the workarounds themselves, which are rather a symptom than a solution.

When using this approach in practice, it is important to consider and address the following challenges. First, the ability to export process event logs from the IS, because not every IS keeps event logs in the required format, and if so, the privacy of the data must be taken care of. Second, since the motivational analysis can also reveal unethical and illegal behavior among the process participants, it is important to encourage cooperative thinking in the interviews rather than audit thinking, which can pose a threat to the process participants.

Several limitations need to be acknowledged. First, the procedure proposed in this paper was thus far implemented in a single case study. Additional implementations in different organizations would provide a more generalizable view of the benefits of this approach. Second, the SWORD framework can detect workarounds only through a process event log. As explained, not all workarounds can be detected by event logs. Additional sources for detecting workarounds may be considered. Third, the motivational analysis highlights problems to be solved, not necessarily possible solutions, so additional constructed ways for process improvement based on the motivational analysis may be considered. Future research can focus on combining additional data sources or ways for workaround detection and on more constructed ways for proposing process improvements.

Acknowledgement. The work was partly supported by the Israel Science Foundation under grant agreement 669/17.

References

1. Martins, P.V., Zacarias, M.: An agile business process improvement methodology. Procedia Comput. Sci. **121**, 129–136 (2017)
2. Satyal, S., Weber, I., Paik, H.Y., Di Ciccio, C., Mendling, J.: Business process improvement with the AB-BPM methodology. Inf. Syst. **84**, 283–298 (2019)
3. Vanwersch, R.J., et al.: A critical evaluation and framework of business process improvement methods. Bus. Inf. Syst. Eng. **58**(1), 43–53 (2016)
4. Boudreau, P., Vieru, D., Paquette, G., Heon, M.: The workarounds process as a source of knowledge creation and management. In: Hawaii International Conference on System Sciences (HICSS), pp. 4292–4301. IEEE (2016)
5. Röder, N., Wiesche, M., Schermann, M., Krcmar, H.: Workaround aware business process modeling. In: Wirtschaftsinformatik, pp. 482–496 (2015)
6. Alter, S.: A workaround design system for anticipating, designing, and/or preventing workarounds. In: Gaaloul, K., Schmidt, R., Nurcan, S., Guerreiro, S., Ma, Q. (eds.) CAISE 2015. LNBIP, vol. 214, pp. 489–498. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19237-6_31
7. Van de Weerd, I., Vollers, P., Beerepoot, I., Fantinato, M.: Workarounds in retail work systems: prevent, redesign, adopt or ignore?. ECIS 2019 (2019)
8. Friedman, A., et al.: A typology of electronic health record workarounds in small-to-medium size primary care practices. J. Am. Med. Inform. Asso. (21) e78–e83 (2014)
9. Ferneley, E.H., Sobreperez, P.: Resist, comply or workaround? an examination of different facets of user engagement with information systems. Eu. J. Inform. Syst. **15**(4), 345–356 (2006)
10. Röder, N., Wiesche, M., Schermann, M., Krcmar, H.: Why managers tolerate workarounds–the role of information systems (2014)

11. Koppel, R., Smith, S., Blythe, J., Kothari, V.: Workarounds to computer access in healthcare organizations: you want my password or a dead patient?. In: Driving Quality in Informatics: Fulfilling the Promise, pp. 215–220. IOS Press (2015)
12. Weinzierl, S., Wolf, V., Pauli, T., Beverungen, D., Matzner, M.: Detecting workarounds in business processes-a deep learning method for analyzing event logs. In: Proceedings of the 28th European Conference on Information Systems (ECIS) (2020)
13. Beerepoot, I., Lu, X., Van De Weerd, I., Alexander Reijers, H.: Seeing the Signs of Workarounds: A Mixed-Methods Approach to the Detection of Nurses' Process Deviations (2021)
14. Beerepoot, I., van de Weerd, I., Reijers, H.A.: The potential of workarounds for improving processes. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) BPM 2019. LNBIP, vol. 362, pp. 338–350. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37453-2_28
15. Soffer, P., Outmazgin, N., Hadar, I., Tzafrir, S.: Why work around the process? Analyzing workarounds through the lens of the theory of planned behavior. Bus. Inform. Syst. Eng. (2023)
16. Ajzen, I.: The theory of planned behavior: Reactions and reflections. Psychol. Health **26**, 1113–1127 (2011)
17. Weinzierl, S., Wolf, V., Pauli, T., Beverungen, D., Matzner, M.: Detecting temporal workarounds in business processes–a deep-learning-based method for analyzing event log data. J. Bus. Anal. **5**(1), 76–100 (2022)
18. Outmazgin, N., Soffer, P.: A process mining-based analysis of business process work-arounds. Softw. Syst. Model. **15**(2), 309–323 (2014). https://doi.org/10.1007/s10270-014-0420-6
19. van der Waal, Wouter, Beerepoot, Iris, van de Weerd, Inge, Reijers, Hajo A.: The SWORD is mightier than the interview: a framework for semi-automatic WORkaround detection. In: Di Ciccio, C., Dijkman, R., del Río, A., Ortega, S.R.-M. (eds.) Business Process Management: 20th International Conference, BPM 2022, Münster, Germany, September 11–16, 2022, Proceedings, pp. 91–106. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-16103-2_9
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 (2015). https://doi.org/10.1007/s00607-015-0441-1
21. Beerepoot, I., Lu, X., van de Weerd, I., Reijers, H.A.: Seeing the signs of workarounds: a mixed-methods approach to the detection of nurses' process deviations. In: Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS), pp. 3763–3772 (2021). https://doi.org/10.24251/hicss.2021.456
22. van Eck, M.L., Xixi, L., Leemans, S.J.J., van der Aalst, W.M.P.: PM: a process mining project methodology. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) Advanced Information Systems Engineering: 27th International Conference, CAiSE 2015, Stockholm, Sweden, June 8-12, 2015, Proceedings, pp. 297–313. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-19069-3_19
23. Miles, M.B., Huberman, A.M., Saldaña, J.: Qualitative Data Analysis: A Methods Sourcebook. 3rd edn. Sage Publications (2014)
24. Outmazgin, N., Soffer, P., Hadar, I.: Workarounds in business processes: a goal-based analysis. In: Dustdar, S., Eric, Yu., Salinesi, C., Rieu, D., Pant, V. (eds.) CAiSE 2020. LNCS, vol. 12127, pp. 368–383. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49435-3_23

25. Beerepoot, I., van de Weerd, I., Reijers, H.A.: Business process improvement activities: differences in organizational size, culture, and resources. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) Business Process Management: 17th International Conference, BPM 2019, Vienna, Austria, September 1–6, 2019, Proceedings, pp. 402–418. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_26
26. Dubinsky, Y., Soffer, P.: Detecting the "Split-Cases" workaround in event logs. In: Augusto, A., Gill, A., Nurcan, S., Reinhartz-Berger, I., Schmidt, R., Zdravkovic, J. (eds.) BPMDS/EMMSAD -2021. LNBIP, vol. 421, pp. 47–61. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79186-5_4

The Impact of Leadership on Business Performance. The Role of Process Performance

Kamila Malewska(✉), Gabriela Roszyk-Kowalska, and Michał Chomicki

Poznań University of Economics and Business, al. Niepodległości 10, 61-875 Poznań, Poland
k.malewska@ue.poznan.pl

Abstract. In a review of the literature on the subject, evidence is found for the impact of leadership on business performance, and the relationship between business process management (BPM) and business performance. It was also recognized that researchers often use mediating variables to explain this relationship. These issues justify the attempt to fill the demonstrated cognitive gap regarding the use of BPM as a mediating variable between leadership and business performance. The aim of the paper is to identify the relationship between leadership, process performance and business performance. The research was conducted using a survey questionnaire and the CAWI method. The survey was conducted in 2023 among 300 randomly selected Polish enterprises. A structural approach was adopted for modelling purposes. The study confirmed that transformational leadership significantly influences the effectiveness and efficiency of processes (process performance), which further influences business performance. Furthermore, it has been proven that responsible leadership correlates with process efficiency, influencing process effectiveness, which in turn impacts business performance. Additionally, it was demonstrated that the level of process maturity correlates most strongly with transformational leadership and business performance. The article contributes to the development of knowledge in the area of leadership and process management by pointing to their important role in achieving planned levels of business performance. Further research directions in this field are also indicated.

Keywords: leadership · BPM · process performance · effectiveness · efficiency · business perfoRmance

1 Introduction

Companies constantly take efforts to adapt to a changing and dynamic environment, while at the same time striving to meet increasingly complex customer needs. These activities are expected to translate into business performance and competitive advantage. Consequently, new ways of management and appropriate types of leadership are being sought to enable efficient management in these ambiguous conditions [1, 2]. The definition of leadership considered as appropriate has evolved over the years [3, 4], but regardless of the type of leadership, researchers have successively proven its impact on business performance [5, 6].

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 391–407, 2023.
https://doi.org/10.1007/978-3-031-41623-1_23

Another construct that is often explored in relation to business performance is business process management (BPM). In recent years, BPM has become a kind of "umbrella" for all approaches to organizational improvement in the context of process management. Companies are looking for ways to increase the efficiency of their business processes, and this is possible through the use of BPM, which provides continuous modelling, analysis, stimulation and evaluation of corporate activities and processes. The benefits of this solution, including improved business performance, have been proven by many researchers [7, 8].

To summarize, the impact of leadership on business performance, as well as the relation between the use of BPM and business performance, are well established and proven in the literature. When analysing research on the relationship between leadership and business performance, it can be seen that researchers often use mediating variables in the form of different types of management concepts to justify this relationship, such as knowledge management, organizational learning, innovation management, project management or team management [9–11]. There is a perceived cognitive gap regarding the use of BPM as a mediating variable between leadership and business performance, especially in view of the empirically proven impact of the BPM concept on corporate performance. Taking into account the above considerations, it seems highly justified to carry out both theoretical and empirical research into the impact of leadership on business performance, taking into account the role of process performance. The aim of the paper is thus to identify the relationship between leadership, process performance and business performance. This article is the result of a broader research project on business process determinants and business process maturity.

The paper is structured as follows. Section 2 provides the theoretical background, discussing types of leadership, the impact of process performance on business performance, and the interaction between leadership, process performance and business performance. Section 3 describes the methods and the research sample. Section 4 provides and discusses the results of the empirical research. The final section of the article contains conclusions, research limitations and further research directions.

2 Theoretical Background

2.1 Types of Leadership – A Conceptual Perspectiveṇ

Leadership style is assumed to be characteristic behaviour or behavioural patterns manifested in the process of managing, leading and motivating a group of people, and which influence their actions [9]. There is no one universal and generally accepted typology of leadership. Many types of leadership are analysed in the literature, from traditional ones such as transactional leadership [12] to contemporary ones such as agile leadership [13]. According to one of the most recognized leadership theories - The Full Range of Leadership Theory - a distinction is made between transformational, transactional and laissez-faire leadership styles [14]. Due to the underlying assumptions of the laissez-faire style, it is difficult to call it leadership, as it implies in its essence the lack of leadership. Laissez-faire leaders do not take action or decisions, do not inspire or motivate their subordinates, so it is difficult to describe their behaviour as leading [15]. Therefore, this leadership style has been excluded both theoretically and empirically from further consideration.

Researchers have proven that under conditions of uncertainty, one extremely important type of leadership that fosters competitive advantage is transformational leadership [16]. They have also demonstrated that it has an impact on company performance [17]. According to transformational leadership theory, it is assumed that through their behaviour, leaders stimulate an innovative mindset in employees that increases their productivity and consequently translates into the performance of the whole organization [18]. A transformational leader aims to improve performance by delegating decision-making authority, increasing employees' autonomy regarding how they perform tasks, fostering learning at both the individual and organizational level, and promoting creative approaches to problem solving by making the best use of existing resources [19]. Transformational leaders are recognized as leaders who help employees realize and develop their potential by identifying needs and collectively determining how to meet them. By communicating the organization's mission and vision, they increase employees' level of identification with the organization [20].

Transactional leadership is defined as an exchange relationship based on contingent reward occurring between a leader and employees. The leader's activities involve setting objectives, monitoring their implementation and checking the results achieved [21]. There are three basic factors that are taken into account when measuring this style of leadership: (1) contingent leadership, which involves setting precise requirements for employees and rewarding them when they fulfil their responsibilities (material or psychological rewards), (2) active management by exception - the leader's aim is to ensure that standards are met through active corrective transactions, (3) passive management by exception - leaders take action when employees' decisions have caused serious problems (passive corrective transactions) [14]. This three-factor measurement of transactional style has been criticised by Tyssen, Wald and Spieth as it alleges an overlap between laissez-faire style assumptions and passive management by exception, and a negative correlation between active and passive management by exception [22]. Therefore, empirical studies of this leadership style are often limited to analysis of the two-factor model of transactional leadership (contingent reward and active management by exception) [21].

One of the trends in contemporary considerations into leadership is responsible leadership, which emerged from research in the fields of ethics, leadership and corporate social responsibility [23]. In the literature, the concept of responsible leadership is analysed from different perspectives. Burton-Jones defines responsible leadership as a multi-level phenomenon that applies to individuals, groups and organizations. It focuses on the importance of performance, ethical behaviour, respect for stakeholders and sustainable practices in economic, social and environmental aspects [24]. According to Rok [25], this type of leadership is based on building relationships that enable benefits to be gained by solving socially relevant problems. Maak [26] highlights the role of responsible leadership in accumulating social capital and running a socially responsible business. Although there are differences in defining responsible leadership, most academic studies emphasize the importance of both internal and external relationships with stakeholders [27]. Responsible leaders develop strong social relationships in the workplace and, through proper communication and climate, are able to maximize employee potential. Responsible leadership plays a key role in building social norms of responsibility in the organization, which encourages employees to support each other in achieving their goals. By building an ethical work environment, responsible leaders have a direct impact on the satisfaction and commitment of their employees and an indirect impact on their performance [28].

2.2 The Impact of Process Performance on Business Performance

Performance provides an organization with a basis for evaluating progress towards predetermined goals, identifying areas of strengths and weaknesses, and guiding future activities for initiating improvements to its operations. Mouzas [29] indicates two measures for assessing performance: effectiveness and efficiency. While some may treat these measures as synonymous, there is a fundamental difference between them.

Effectiveness is concerned with organization output, i.e. sales, value added, quality, cost reduction or innovation. It measures the extent to which a company achieves its objectives or how the effects of its operation interact with the economic and social environment. Typically, effectiveness is defined by an organization's policy objectives or the extent to which an organization meets its own goals [30].

Efficiency describes the relationship between inputs and outputs, in other words how successfully inputs are transformed into outputs. Therefore efficiency is the ability to do something or produce something without wasting materials, time or energy [31].

The most desirable state is when a company is run efficiently and effectively, as this results in the continued survival of the organization and the achievement of its goals with minimum cost. When a business is efficient but ineffective the company is slowly bankrupting, as even though its costs are under control it cannot achieve its objectives. Meanwhile, being inefficient but effective means that a company can achieve its goals but at a high cost [32].

A lack of improvement in business processes results in redundant operations, inefficiency and reduced competitiveness, which ultimately affects a company's ability to succeed in the long and short term [33]. Applying business processes management (BPM) practices enables different parts of an organization to effectively and efficiently co-create value, and ultimately provide satisfaction to the company's customers [34].

At a certain level of an organization's process maturity, integrated process management is necessary, both internally - i.e. the establishment of process dependencies and hierarchies (process architecture), and externally - i.e. integrated into the enterprise management system [35]. The effect of integrated process architecture management is to align it with the company's strategy. The business architecture of BPM results in both process efficiency and process improvements being translated into business performance. This requires collaboration and appropriate competence on the part of the accounting and process teams in the area of process instrumentation and project improvement so as to continuously create net value. The significant efficiency of BPM comes not only from the fact that it maximises gross value by developing new capabilities and improvements, but also from the fact that this is accomplished with a minimum of cost, time and waste [8].

2.3 The Interaction Between Leadership, Process Performance and Business Performance

The effectiveness of the functioning of any organization, both in the operational dimension related to day-to-day decisions, and in the strategic dimension related to its development, depend on the specific characteristics of its leader. Effective organizational management requires not only an organizational vision and the ability to communicate it, but, above all, the ability to motivate employees to take action to achieve the formulated goals [2]. Thus, appropriate leadership is necessary for the proper functioning of any organization, understood as the achievement of its objectives. The question arises as to what type of leadership is most appropriate [3]. Over the years, new types of leadership have been formulated. Their emergence resulted from changes in the environment in which organizations operate [36]. There has been a debate in the literature for years about the impact of leadership on business performance. Over the years, researchers have proven that particular types of leadership have a positive impact on business performance [6, 37]. It seems reasonable to seek an answer to the question of which types of leadership distinguished in the literature have the most significant impact on the business performance of modern-day companies. The traditional theories of development prevalent in the 1970s assumed the development of the enterprise as part of economic growth, and traditional (transactional) leadership was exclusively profit-oriented [38]. Today's leaders find themselves in a completely different reality, and have to find a balance between the different interrelated aspects of the organization's functioning (economic, social and often also environmental) [39]. Therefore, it seems reasonable to conclude that among the types of leadership analysed in this article (transactional, transformational and responsible), transformational leadership and responsible leadership in particular should have a positive impact on business performance.

Another variable that is often analysed in terms of its impact on business performance is BPM. In recent years, there has been a trend towards the use of the BPM concept by an increasing number of companies, for which its application is becoming everyday practice. This is the result of a desire to optimize processes regardless of the area in which the organization operates [40]. The use of BPM in management practice enables significant optimization and productivity gains, along with simultaneous cost reductions. It also allows for a better understanding of the organization's functioning and how its constituent activities can be improved. This means adopting a process-centric approach to improving business performance. The use of BPM has both managerial and technical implications, and therefore requires close cooperation between managers and information technologists to ensure effective, flexible and transparent business processes [41]. The results of empirical studies indicate that many positive outcomes are achieved by organizations as a consequence of the successful implementation of BPM techniques and principles, including an impact on overall business performance [8].

Based on the above considerations, the impact of leadership on business performance and the impact of BPM on business performance have been indisputably proven by researchers. In empirical studies conducted on the relation between leadership and business performance, researchers often use mediators. Among the analysed mediating variables are various management concepts such as innovation management, project management, team management, knowledge management, organizational learning and safety management [10, 11, 42–44]. However, there are no studies on the impact of the type of leadership on business performance with process performance treated as a mediator.

Looking for a theoretical basis to explain the relationship between leadership style and process performance, one can refer to Kanter's theory of structural empowerment [45]. This assumes the existence of appropriate social structures in the workplace (e.g., a particular leadership style) that enable individuals to achieve their goals through access to learning and development opportunities, information, support and resources. This is particularly justified in the case of a transformational leadership style. It would be cognitively interesting to also verify this relationship with regard to the other leadership styles analyzed in the article, i.e. transactional and responsible. Therefore, we suggest the following three hypotheses:

H1: Transactional leadership has a positive impact on business performance with a mediating role of process performance.
H2: Transformational leadership has a positive impact on business performance with a mediating role of process performance.
H3: Responsible leadership has a positive impact on business performance with a mediating role of process performance.

3 Research Method and Data Acquisition

3.1 Research Tools

All of the constructs presented in the theoretical section of the paper were measured using validated research tools developed by other researchers from each research area. For all constructs the items were assessed using a 5-point Likert scale.

In terms of transactional leadership, we applied the proposalby Podsakoff et al. [46]. The construct consisted of 3 items: (1) expressing dissatisfaction with an employee's failure to perform a task, (2) reporting a poorly performed task, (3) pointing out errors in a poorly performed task.

In terms of transformational leadership, we adopted the proposal by Hoai et al. [47]. This construct consisted of 5 items: (1) a clear vision of the future, (2) leading by example, (3) encouraging people to think about old problems in a new way (through challenges), (4) motivating employees to be proud of being part of the organization, (5) awareness of the organization's aspirations over a 5-year perspective.

A research tool developed by Voegtlin [48] was applied to responsible leadership. The construct consisted of 5 items: (1) awareness of relevant stakeholder claims, (2) awareness of the implications of decisions for stakeholders, (3) stakeholder involvement in the decision-making process, (4) consideration of various stakeholder claims before a decision is made, (5) achieving consensus among stakeholders.

To measure process performance, we applied a ten-item construct proposed by Schmiedel et al. [49] based on a procedure formulated by MacKenzie et al. [50]. In this view, the process performance construct consists of two sub-constructs: process efficiency and process effectiveness, each of which describes five indicators.

The business performance construct was measured using a tool proposed by Wang et al. [51]. The tool included 9 items: (1) relative product quality, (2) new product success,

(3) customer retention rate, (4) sales level, (5) sales growth rate, (6) relative market share, (7) return on equity, (8) gross profit margin, (9) return on investment.

The survey answers were obtained with use of the CAWI method. The research was conducted in January 2023 on a random selection of Polish enterprises. The survey respondents were: specialists (62.66%), mid-level managers (9.00%), line managers (8.33%), senior managers (7.33%) and other employees. The variables were scored on a scale of 1 to 5, were 1 meant “I strongly disagree”, and 5 “I strongly agree”. The respondents’ Likert item scores were averaged for all the items comprising each Likert scale, resulting in a pseudo-quantitative representation of respondents’ opinions on a given construct. Altogether 300 responses were collected, providing results for 2022. The collected data was checked by two independent experts for correctness, then coded and analysed. For the purpose of statistical analysis, the Python statsmodels package version 0.13.5 was used [52].

3.2 Description of the Research Sample

The structure of the research sample in terms of personnel, process maturity and company activity is presented in the table below. The maturity of the processes was determined according to a model developed by MacCormak and Johnson [53]. The biggest group in terms of personnel was made up of large companies (39.66%). The largest group in terms of process maturity were companies representing the second level of maturity (38% of the research sample). For company activity, the Polish Classification of Activities 2007 was applied (GUS 2023), with the most numerous being services at 53% (Table 1).

Table 1. Characteristics of the research sample for the year 2022, n = 300.

Personnel		Process Maturity		Industry (Polish Classification of Activity)	
1–9	11.33%	PM_1	13.67%	Agriculture, forestry, hunting and fishing (code A)	0.67%
10–49	27.67%	PM_2	38.00%	Manufacturing (code C)	39.00%
50–249	21.33%	PM_3	35.00%	Construction (code F)	2.33%
250 +	39.67%	PM_4	13.33%	Wholesale and retail trade (code G)	5.00%
				Services (codes H-S)	53.00%

4 Research Results

For the purposes of modelling, the structural approach was adopted, whilethe underlying management and organization theory drove the process of model selection. The following regression tables were created using the OLS method with Newey-West heteroskedasticity and autocorrelation of robust standard errors with Bartlett Kernel in the case of models flawed with either[54]. In some cases, it was prudent to control for two nominal variables that are ordinal: process maturity level (PM_x) and company size (Size_x). This was accomplished through the use of dummy variables corresponding to (n-1) categories derived from these questions. The first modelled variable was Process Effectiveness (PEffn). The available variables for selection purposes were: Transformational Leadership (TfL), Responsible Leadership (SL), Transactional Leadership (TsL), and Process Efficiency (PEffc). Only the statistically significant models are presented in Table 2 below.

Table 2. Results of OLS regression modelling – Process Effectiveness.

	Model_1	Model_2	Model_3	Model_4	Model_5	Model_6
Intercept	2.348*** (0.129)	1.494*** (0.167)	2.654*** (0.152)	2.488*** (0.143)	2.519*** (0.143)	1.716*** (0.181)
TfL	0.246*** (0.036)	0.092** (0.04)	0.193*** (0.039)	0.199*** (0.039)	0.208*** (0.039)	
PEffc		0.369*** (0.051)				0.401*** (0.046)
PM_1			−0.386*** (0.115)	−0.236** (0.111)	−0.289*** (0.109)	−0.188* (0.101)
PM_2			−0.183** (0.075)			
PM_3				0.155** (0.076)		
R-squared	0.134	0.266	0.171	0.166	0.155	0.261
Adj. R-squared	0.131	0.261	0.163	0.158	0.149	0.256
Observations	300	300	300	300	300	300
F-stat (p-value)	46.26 (0.00)	53.73 (0.00)	20.36 (0.00)	19.68 (0.00)	27.16 (0.00)	52.39 (0.00)
Jarque-Bera (p-value)	1.09 (0.58)	1.86 (0.39)	0.81 (0.67)	1.24 (0.54)	1.02 (0.6)	1.34 (0.51)
Durbin-Watson	2.18	2.24	2.16	2.15	2.17	2.22
Breusch-Pagan (p-value)	0.22 (0.64)	0.93 (0.63)	0.56 (0.91)	0.13 (0.99)	0.33 (0.85)	0.44 (0.8)

Standard errors in parentheses.
$* p < .1$, $** p < .05$, $***p < .01$.

The only statistically significant variable of the three leadership measures was transformational leadership, rendering it the main driver of the regression model. Since Process Efficiency is a prerequisite for Process Effectiveness, it was considered as an additional driver in the model. In view of the fact that, as a result of including dummy variables controlling for process maturity level and company size, one of the two independent variables lost statistical significance, Model 2 was chosen as the champion.

Model 2 has the highest adjusted R2 statistic, at 0.261, a value that is acceptable in the domain of social science. The magnitude and sign of the coefficients align with scientific intuition and the underlying theory – both of the drivers positively impact the forecasted value. The basic diagnostics of the model indicate that the residuals are normally distributed, homoscedastic and slightly negatively autocorrelated, but the Durbin-Watson test value of 2.24 does not indicate an impact on the model estimation process.

It is worth noting that PM_1 and PM_2, if included in the regression models, have negative coefficients, effectively penalizing companies with a lack of or low level of process maturity. This was expected, since low maturity indicates a lower ability to perform processes with a positive economic result.

The next modelled variable was Process Efficiency (PEffc). The available variables for selection purposes were: Transformational Leadership (TfL), Responsible Leadership (RL), and Transactional Leadership (TsL). Only the statistically significant models are presented in Table 3.

Table 3. Results of OLS regression modelling – Process Efficiency.

	Model_1	Model_2	Model_3
Intercept	1.783*** (0.226)	2.029*** (0.256)	1.833*** (0.22)
TfL	0.272*** (0.041)	0.222*** (0.042)	0.264*** (0.04)
RL	0.286*** (0.061)	0.281*** (0.062)	0.287*** (0.06)
PM_1		−0.400*** (0.127)	
Size_1			−0.243* (0.134)
R-squared	0.348	0.376	0.358
Adj. R-squared	0.343	0.370	0.352
Observations	300	300	300
F-stat (p-value)	56.13 (0.00)	46.21 (0.00)	39.66 (0.00)
Jarque-Bera (p-value)	12.03 (0.00)	6.04 (0.05)	6.57 (0.04)

(*continued*)

Table 3. (*continued*)

	Model_1	Model_2	Model_3
Durbin-Watson	1.85	1.86	1.88
Breusch-Pagan (p-value)	20.1 (0.00)	22.02 (0.00)	23.83 (0.00)

Standard errors in parentheses.
** p < .1, ** p < .05, ***p < .01.*

Model 2 was selected as the champion model. It has the highest adjusted R2 value, at 0.370, which is more than acceptable in the domain of social science. The residuals are normally distributed, but heteroscedastic (the Newey-West robust standard errors were employed in the process of model estimation). The residuals also exhibit positive autocorrelation, but the Durbin-Watson test value of 1.86 does not indicate that autocorrelation impacts the model estimation process. The two variables that account for a significant part of the variability are transformational and responsible leadership. It should be noted that the champion model specification includes a dummy variable representing companies lacking process maturity. The magnitudes and signsof the coefficients align with the underlying theories and intuition.

The last modelled variable is Business Performance. Conceptually, Process Effectiveness (PEffn) and Process Efficiency (PEffc) were expected to drive business performance. At the initial modelling stage, it turned out that models fitted to the whole dataset were flawed, with residuals that were not normally distributed. Upon visual inspection of the residual histogram and the distribution of the Business Performance variable, it was decided to reduce the sample based on the outliers of the dependent variable. The sample reduction rules were developed based on the distance from the median of 1.5 IQR or more (effectively excluding the underperforming companies with BP <= 2) (Fig. 1).

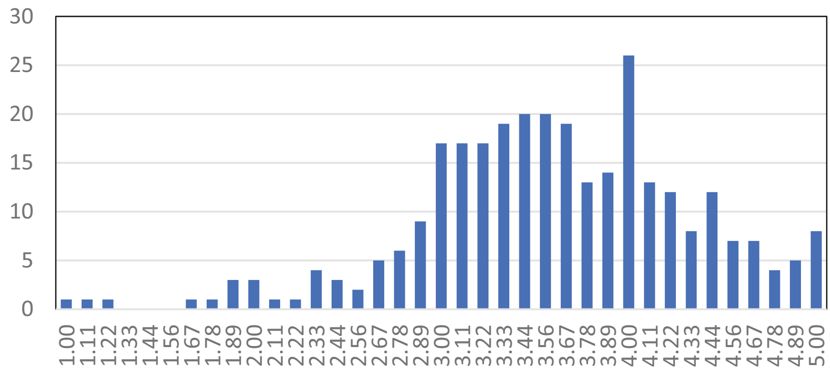


Fig. 1. Distribution of Business Performance (with outliers subject to exclusion).

The number of excluded respondents was around 3.5% of the sample size (11 entities; 6 – lack of PM, 3 – low-level PM, and 2 –mid-level PM). There was also a practical justification for this exclusion – the underperforming entities that were stricken with post-covid economics should not impact the inferential process (Table 4).

Table 4. Results of OLS regression modelling – Business Performance.

	Model_1	Model_2	Model_3	Model_4
Intercept	1.504*** (0.176)	1.796*** (0.197)	1.480*** (0.175)	1.757*** (0.197)
PEffn	0.165*** (0.053)	0.134** (0.053)	0.169*** (0.053)	0.139*** (0.052)
PEffc	0.433*** (0.047)	0.412*** (0.047)	0.421*** (0.047)	0.403*** (0.047)
PM_1		−0.214** (0.100)		−0.194* (0.099)
PM_2		−0.221*** (0.063)		−0.218*** (0.063)
Size_5			0.140** (0.060)	0.132** (0.059)
R-squared	0.367	0.395	0.379	0.405
Adj. R-squared	0.362	0.386	0.372	0.395
Observations	289	289	289	289
F-stat (p-value)	82.76 (0.00)	46.33 (0.00)	57.88 (0.00)	38.59 (0.00)
Jarque-Bera (p-value)	2.21 (0.33)	2.33 (0.31)	2.68 (0.26)	2.7 (0.26)
Durbin-Watson	1.90	1.93	1.89	1.92
Breusch-Pagan (p-value)	1.22 (0.54)	7.31 (0.12)	1.09 (0.78)	7.13 (0.21)

Standard errors in parentheses.
** p < .1, ** p < .05, ***p < .01.*

The models presented above are all similar in terms of the model fit (the adjusted R2 varies between 0.36 and 0.40). The models have coefficient signs and magnitudes that align with underlying theories. The residuals are characterised by a minor level of positive autocorrelation that does not impact the estimation reliability. Surprisingly, Business Performance is driven mostly by PEffc, and not the expected PEffn. The higher impact of PEffc on Business Performance can be explained by the fact that PEffc is a better predictor of BP in the longer term.

5 Discussion and Further Research Directions

The results of the analyses provide a basis for rejecting hypothesis H1, according to which transactional leadership has a positive impact on business performance with a mediating role of process performance. Analysis of the data showed that this leadership style does not affect business performance (statistically insignificant variable). In addition, it correlates the least with process performance, consisting of process efficiency and process effectiveness. Therefore, this type of leadership was removed from the model and was not analysedfurther.This is in line with the assumptions formulated by Klarin [38] and Correia [39], according to which transactional leadership has a low impact on business performance.However, it seems interesting that there is no relationship between a transactional leadership style and business performance, since in its assumptions, it is the most profit-oriented of the analyzed leadership styles. This can be explained by the fact that today's organizations cannot focus solely on the economic aspect of their operations. They must also try to effectively connect it with social and environmental aspects, and the transformational (social aspect) and responsible (environmental aspect) styles are more oriented toward these threads.

The research justifies the conclusion that transformational leadership positively impacts business performance with a mediating role of process performance. In relation to this leadership style, the impact on business performance was identified of both sub-constructs that form the process performance construct (process efficiency and process effectiveness). This is in line with Dumdum [17] and Judge &Piccol [18], who claim that leaders stimulate innovative thinking in employees through their behaviour, which increases employees' productivity and consequently the business performance of the whole organization. These reflections, however, do not highlight the importance of BPM, but rather look at the direct impact of transformational leadership on the functioning of the organization. Our research clearly shows that transformational leadership significantly impacts process performance, which is relevant to achieving the company's desired business outcomes. On the other hand, researchers emphasize the role of the use of BPM techniques on the overall performance of the organization [8]. However, there is a lack of research that disaggregates process performance into process effectiveness and process efficiency, and determines the impact of each individual construct on business performance.

The study also showed a relationship between responsible leadership style and business performance, with a mediating role of process efficiency. For this type of leadership, an impact on process effectiveness was not identified. Similarly to transformational leadership, there is also a lack of publications on the impact of BPM on company performance in relation to responsible leadership. The mediating influence of other management concepts is analysed, but not BPM [10, 11, 44].

It is important to emphasize that the results of our study extend knowledge on the relationship between the type of leadership, the use of BPM and business performance. They point to the significant, mediating role of BPM between transformational and responsible leadership and business performance. Both transformational and responsible leadership enable employees to develop social skills, which contribute to the organization's value creation and positively impact business performance, with a mediating role of process efficiency. Accordingly, managers' efforts (transformational and responsible leaders) provide the organization with opportunities to improve processes, motivate and develop employees, and increase the effectiveness of information systems. The ability to learn and to increase the competence of teams of people led by transformational and responsible managers affects both business performance and the impact of BPM on business performance, as evidenced in the research investigation. What is surprising about the regularity shown in the modelling is that business performance depends primarily on process efficiency rather than the expected process effectiveness (according to the literature review) [29, 31]. Nonetheless, the more significant impact of process efficiency on business performance may be justified by the fact that process efficiency is a better explanatory variable for BPM in the long term. Thus, it is postulated that further research should attempt to explain this phenomenon in a shorter perspective of organizational activity, which may be especially relevant for organizational change processes. Another area of research worth undertaking in terms of the correlation between transformational and responsible leadership and the business outcomes achieved by organizations with the mediating role of process efficiency, may be related to the high level of unique competencies assessed by the number and type of very rare, specialized competencies and/or exceptionally high business performance. It also seems reasonable to seek answers to the question of how leadership style affects BPM practices, that is, how BPM is implemented depending on the type of leadership.

6 Conclusions and Research Limitations

In conclusion, on the basis of the research and as a result of the statistical analyses carried out, it was indicated that transactional leadership correlates least with the efficiency and effectiveness of processes (process performance) and their level of maturity. Data analysis showed that this leadership style did not influence business performance (a statistically insignificant variable) and it was therefore removed from the model. On this basis, it can be noted that first hypothesis (H1: Transactional leadership has a positive impact on business performance with a mediating role for process performance) has not been confirmed.

In contrast, responsible leadership correlates with process efficiency, influencing process effectiveness, which further impacts business performance. Conversely, transformational leadership significantly affects the effectiveness and efficiency of processes, which further influences business performance (as shown in the model). Therefore, the level of maturity of the processes correlates best with transformational leadership and business performance. On this basis, it can be concluded that second hypothesis (H2: Transformational leadership has a positive impact on business performance with a mediating role for process performance) has been confirmed. The third hypothesis (H3: Responsible leadership has a positive impact on business performance with a mediating role for process performance) has been partially verified. This was because the study assumed that process performance consists of two sub-constructs: process efficiency and process effectiveness. The mediating role of process efficiency between responsible leadership style and business performance was proven. However, the mediating role of process effectiveness could not be confirmed.

Some limitations can be identified with regard to the research procedure. Firstly, a single respondent approach was used. It would be advisable to extend the study by inviting managers responsible for different processes in the organization to explore the role of process performance in the relationship between different types of leadership and business performance.

Secondly, the survey respondents carried out a self-assessment. Several drawbacks associated with research using self-assessment may lead to questionable results. Although criticized and likely to produce less reliable results than a survey using standardized and validated measurement instruments, self-assessment remains a necessary choice in cases of larger samples, geographical dispersion of respondents, and the need to maintain the cost regime of surveys.

Finally, the measurements were performed based on Likert scales mostly comprising of 5 Likert items (with an exception for transactional leadership – 3 items, and business performance – 9 items). This approach allowed the research team to obtain construct measurement on pseudo-quantitative scales that allow for n_items (n_choices-1) + 1 possible outcomes per scale. In the case of the scale with the lowest number of items – transactional leadership – there are 13 possible outcomes (for the others it is 21, while for business performance it is 37), which should ensure the validity of the quantitative analysis.

Acknowledgement. The project was financed within the Regional Initiative for Excellence program of the Minister of Science and Higher Education of Poland, in the years 2019-2022, grant no. 004/RID/2018/19, financing 3,000,000 PLN.

References

1. İşcan, Ö.F., Ersarı, G., Naktiyok, A.: Effect of leadership style on perceived organizational performance and innovation: the role of transformational leadership beyond the impact of transactional leadership – an application among Turkish SME's. Procedia - Soc. Behav. Sci. **150**, 881–889 (2014). https://doi.org/10.1016/j.sbspro.2014.09.097
2. Chen, Y., Ning, R., Yang, T., Feng, S., Yang, C.: Is transformational leadership always good for employee task performance? Examining curvilinear and moderated relationships. Front. Bus. Res. China **12**(1), 1–28 (2018). https://doi.org/10.1186/s11782-018-0044-8
3. Kjellström, S., Stålne, K., Törnblom, O.: Six ways of understanding leadership development: an exploration of increasing complexity. Leadership **16**, 434–460 (2020)
4. Mai, N.K., Do, T.T., Phan, N.A.: The impact of leadership traits and organizational learning on business innovation. J. Innov. Knowl. **7** (2022). https://doi.org/10.1016/j.jik.2022.100204
5. Chatterjee, S., Chaudhuri, R., Vrontis, D., Giovando, G.: Journal of Innovation digital leadership capability. J. Innov. Knowl. **8**, 100334 (2023). https://doi.org/10.1016/j.jik.2023.100334
6. Yang, M., Al, A., Salameh, A.A.: Leadership, capability and performance: a study among private higher education institutions in Indonesia. Heliyon. **9**, e13026 (2023). https://doi.org/10.1016/j.heliyon.2023.e13026
7. Capgemini: Global Business Process Management Report (2012)
8. Dumitriu, D.: Research on the trend and potential impact of adopting BPM techniques over general performance of the organization. Procedia Manuf. **22**, 575–582 (2018). https://doi.org/10.1016/j.promfg.2018.03.084
9. Aboramadan, M., Dahleez, K.A.: Leadership styles and employees' work outcomes in nonprofit organizations: the role of work engagement. J. Manag. Dev. **39**, 869–893 (2020). https://doi.org/10.1108/JMD-12-2019-0499

10. Siddiquei, A.N., Fisher, C.D., Hrivnak, G.A.: Temporal leadership, team processes, and project team task performance. Int. J. Proj. Manag. **40**, 715–724 (2022). https://doi.org/10.1016/j.ijproman.2022.08.005
11. Qalati, S.A., Zafar, Z., Fan, M., Sánchez Limón, M.L., Khaskheli, M.B.: Employee performance under transformational leadership and organizational citizenship behavior: a mediated model. Heliyon. **8**, e11374 (2022). https://doi.org/10.1016/j.heliyon.2022.e11374
12. Bass, B.M.: Transformational leadership: learning to share the vision. Organ. Dyn. **18**, 19–31 (1990)
13. Luis, E.C., Celma, D.: Circular economy. A review and bibliometric analysis. Sustain. **12** (2020). https://doi.org/10.3390/SU12166381
14. Antonakis, J., Avolio, B.J., Sivasubramaniam, N.: Context and leadership: an examination of the nine-factor full-range leadership theory using the Multifactor Leadership Questionnaire (2003). https://doi.org/10.1016/S1048-9843(03)00030-4
15. Ågotnes, K.W., et al.: Daily work pressure and exposure to bullying-related negative acts: the role of daily transformational and laissez-faire leadership. Eur. Manag. J. **39**, 423–433 (2021). https://doi.org/10.1016/j.emj.2020.09.011
16. Nemanich, L.A., Keller, R.T.: Transformational leadership in an acquisition: a field study of employees. Leadersh. Q. **18**, 49–68 (2007). https://doi.org/10.1016/j.leaqua.2006.11.003
17. Dumdum, U.R., Lowe, K.B., Avolio, B.J.: A meta-analysis of transformational and transactional leadership correlates of effectiveness and satisfaction: an update and extensio. In: Avolio, B.J., Yammarino, F.J. (eds.) Transformational and charismatic leadership: The road ahead, pp. 36–66. Elsevier Science, Oxford (2002)
18. Judge, T.A., Piccol, R.F.: Transformational and transactional leadership: a meta-analytic test of their relative validity. J. Appl. Psychol. **89**, 755–768 (2004). https://doi.org/10.1037/0021-9010.89.5.755
19. Gumusluoglu, L., Ilsev, A.: Transformational leadership, creativity, and organizational innovation. J. Bus. Res. **62**, 461–473 (2009). https://doi.org/10.1016/j.jbusres.2007.07.032
20. Peterson, S.J., Walumbwa, F.O., Byron, K., Myrowitz, J.: CEO positive psychological traits, transformational leadership, and firm performance in high technology start-up and established firms. J. Manage. **35**, 348–368 (2009). https://doi.org/10.1177/0149206307312512
21. Aga, D.A.: Transactional leadership and project success: the moderating role of goal clarity. Procedia Comput. Sci. **100**, 517–525 (2016). https://doi.org/10.1016/j.procs.2016.09.190
22. Tyssen, A.K., Wald, A., Spieth, P.: The challenge of transactional and transformational leadership in projects. Int. J. Proj. Manag. **32**, 365–375 (2014). https://doi.org/10.1016/j.ijproman.2013.05.010
23. Cyfert, S., Szumowski, W., Dyduch, W., Zastempowski, M., Chudziński, P.: The power of moving fast: responsible leadership, psychological empowerment and workforce agility in energy sector firms. Heliyon. **8**, e11188 (2022). https://doi.org/10.1016/j.heliyon.2022.e11188
24. Szczepańska-Woszczyna, K., Dacko-Pikiewicz, Z., Lis, M.: Responsible leadership: a real need or transient curiosity. Procedia - Soc. Behav. Sci. **213**, 546–551 (2015). https://doi.org/10.1016/j.sbspro.2015.11.448
25. Rok, B.: Naprawdę odpowiedzialne przywództwo. Kompend. CSR. 13 (2014)
26. Maak, T.: Responsible leadership, stakeholder engagement, and the emergence of social capital. J. Bus. Ethics. **74**, 329–343 (2007). https://doi.org/10.1007/s10551-007-9510-5
27. Maak, T., Pless, N.M., Voegtlin, C.: Business statesman or shareholder advocate? CEO responsible leadership styles and the micro-foundations of political CSR. J. Manag. Stud. **53**, 463–493 (2016). https://doi.org/10.1111/joms.12195
28. Doh, J.P., Quigley, N.R.: Responsible leadership and stakeholder management: influence pathways and organizational outcomes. Acad. Manag. Perspect. **28**, 255–274 (2014). https://doi.org/10.5465/amp.2014.0013

29. Mouzas, S.: Efficiency versus effectiveness in business networks. J. Bus. Res. **59**, 1124–1132 (2006). https://doi.org/10.1016/j.jbusres.2006.09.018
30. Kafashpoor, A., Shakoori, N., Sadeghian, S.: Linking organizational culture, structure, Leadership Style , strategy , and organizational effectiveness: Mediating role of knowledge management **10**, 158–172 (2013)
31. Wilson, M., Wnuk, K., Silvander, J., Gorschek, T.: A Literature Review on the Effectiveness and Efficiency of Business Modeling (2018). https://doi.org/10.5277/e-Inf180111
32. Zokaei, A.K., Simons, D.W.: Value chain analysis in consumer focus improvement a case study of the UK red meat industry (2017). https://doi.org/10.1108/09574090610689934
33. Kasim, T., Haračić, M., Haračić, M.: The improvement of business efficiency through business process management. Econ. Rev. J. Econ. Bus. **XVI** (2018)
34. IBM: What is business process management? (2018)
35. Cyfert, S.: Strategiczne doskonalenie architektury procesów w zarządzaniu przedsiębiorstwem. Wydawnictwo Akademii Ekonomicznej w Poznaniu, Poznań (2006)
36. Piwowar-Sulej, K., Iqbal, Q.: Leadership styles and sustainable performance: a systematic literature review. J. Clean. Prod. **382**, 134600 (2022). https://doi.org/10.1016/j.jclepro.2022.134600
37. Aman-ullah, A., Mehmood, W., Amin, S., Abbas, Y.A.: Journal of innovation. J. Innov. Knowl. **7**, 100261 (2022). https://doi.org/10.1016/j.jik.2022.100261
38. Klarin, T.: The concept of sustainable development: from its beginning to the contemporary issues **21**, 67–94 (2018). https://doi.org/10.2478/zireb-2018-0005
39. Correia, M.S.: Sustainability: An Overview of the Triple Bottom Line and Sustainability Implementation (2020). https://doi.org/10.4018/IJoSE.2019010103
40. Hofstede, A.H.M., Aalst, W.M.P., Adams, M., Russell, N. (eds.): Modern Business Process Automation: YAWL and its Support Environment. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-03121-2
41. von Rosing, M., Scheer, A.-W., von Scheel, H.: he Complete Business Process Handbook. Elsevier Inc, Body of Knowledge from Process Modeling to BPM (2015)
42. Yıldırım, N., Birinci, S.: Impacts of organizational culture and leadership on business performance: a case study on acquisitions. Procedia - Soc. Behav. Sci. **75**, 71–82 (2013). https://doi.org/10.1016/j.sbspro.2013.04.009
43. Arsezen-Otamis, P., Arikan-Saltik, I., Babacan, S.: The relationship between paternalistic leadership and business performance in small tourism businesses: the moderating role of affective organizational commitment. Procedia - Soc. Behav. Sci. **207**, 90–97 (2015). https://doi.org/10.1016/j.sbspro.2015.10.150
44. Vargas, M.I.R.: Determinant factors for small business to achieve innovation, high performance and competitiveness: organizational learning and leadership style. Procedia - Soc. Behav. Sci. **169**, 43–52 (2015). https://doi.org/10.1016/j.sbspro.2015.01.284
45. Kanter, R.M.: Men and Women of the Corporation. Basic Books, New York (1977)
46. Podsakoff, P.M., Abearne, M., MacKenzie, S.B.: Organizational citizenship behavior and the quantity and quality of work group performance. J. Appl. Psychol. **82**, 262–270 (1997)
47. Hoai, T.T., Hung, B.Q., Nguyen, N.P.: The impact of internal control systems on the intensity of innovation and organizational performance of public sector organizations in Vietnam: the moderating role of transformational leadership. Heliyon. **8**, e08954 (2022). https://doi.org/10.1016/j.heliyon.2022.e08954
48. Voegtlin, C.: Development of a scale measuring discursive responsible leadership. Responsible Leadersh. 57–73 (2012). https://doi.org/10.1007/s10551-011-1020-9
49. Schmiedel, T., Recker, J., vom Brocke, J.: The relation between BPM culture, BPM methods, and process performance: evidence from quantitative field studies. Inf. Manag. **57**, 103175 (2020). https://doi.org/10.1016/j.im.2019.103175

50. MacKenzie, S.B., Podsakoff, P.M., Podsakoff, N.P.: Construct measurement and validation procedures in MIS and behavioral research: Integrating new and existing techniques. MIS Q. Manag. Inf. Syst. **35**, 293–334 (2011). https://doi.org/10.2307/23044045
51. Wang, E., Klein, G., Jiang, J.J.: IT support in manufacturing firms for a knowledge management dynamic capability link to performance. Int. J. Prod. Res. **45**, 2419–2434 (2007). https://doi.org/10.1080/00207540601020437
52. Seabold, S., Perktold, J.: Statsmodels: Econometric and statistical modelingwith python. In: Proceedings of the 9th Python in Science Conference. p. Seabold, S., Perktold, J. (2010). Statsmodels: E (2010)
53. MacCormack, K.P., Johnson, W.C.: Business Process Orientation Book Business Process Orientation. Gaining the E-Business Competitive Advantage. CRC Press, Abingdon (1998). https://doi.org/10.4324/9780367815608
54. Newey, W.K., West, K.D.: A simple PSD HAC covariance Matrix.pdf. Econometrica **55**, 703–708 (1987). https://doi.org/10.2307/1913610

Author Index

C. Di Francescomarino et al. (Eds.): BPM 2023 Forum, LNBIP 490, pp. 409–410, 2023.
https://doi.org/10.1007/978-3-031-41623-1