

Cristina Cabanillas
Niels Frederik Garmann-Johnsen
Agnes Koschmider (Eds.)

LNBIP 460

Business Process Management Workshops


BPM 2022 International Workshops
Münster, Germany, September 11–16, 2022
Revised Selected Papers


 **Springer**


Lecture Notes in Business Information Processing


460

Series Editors

Wil van der Aalst , *RWTH Aachen University, Aachen, Germany*

John Mylopoulos , *University of Trento, Trento, Italy*

Sudha Ram , *University of Arizona, Tucson, AZ, USA*

Michael Rosemann , *Queensland University of Technology,
Brisbane, QLD, Australia*

Clemens Szyperski, *Microsoft Research, Redmond, WA, USA*

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.

Cristina Cabanillas ·
Niels Frederik Garmann-Johnsen ·
Agnes Koschmider
Editors

Business Process Management Workshops

BPM 2022 International Workshops
Münster, Germany, September 11–16, 2022
Revised Selected Papers

Editors

Cristina Cabanillas 
University of Seville
Sevilla, Spain

Niels Frederik Garmann-Johnsen 
University of Agder
Kristiansand, Norway

Agnes Koschmider 
Kiel University
Kiel, Germany

ISSN 1865-1348

ISSN 1865-1356 (electronic)

Lecture Notes in Business Information Processing

ISBN 978-3-031-25382-9

ISBN 978-3-031-25383-6 (eBook)

<https://doi.org/10.1007/978-3-031-25383-6>

© Springer Nature Switzerland AG 2023

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

Preface

BPM is an annual, international conference that covers all aspects related to Business Process Management. It has become the most prestigious forum for researchers and practitioners in the field and serves as a melting pot for experts from a mix of disciplines including Computer Science, Information Systems Engineering, and Management. The BPM conference itself is complemented by a workshop program, where workshops dedicated to more specialized themes, to cross-cutting issues, and to upcoming trends and paradigms can be easily and conveniently organized with little administrative overhead. This volume collects the proceedings of the BPM 2022 workshops held on September 12, 2022 in Münster, Germany.

BPM 2022 solicited proposals for one-day or half-day workshops to be held before the main conference. In the workshop review and acceptance process, priority was given to proposals that not only addressed an exciting topic but also promised a creative format likely to generate lively interactions and foster new ideas. Examples include panels connecting practitioners and researchers or research-in-progress papers for young researchers. Of the 10 workshop proposals submitted, the following 8 workshops were selected for co-location with BPM 2022:

- *6th International Workshop on Artificial Intelligence for Business Process Management (AI4BPM 2022)* organized by Chiara Di Francescomarino, Fabrizio Maria Maggi, Andrea Marrella, Arik Senderovich and Emilio Sulis.

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

- *6th International Workshop on BP-Meet-IoT (BP-Meet-IoT 2022)* organized by Francesco Leotta, Massimo Mecella, Estefania Serral and Victoria Torres.

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

- *18th International Workshop on Business Process Intelligence (BPI 2022)* organized by Jochen De Weerd, Marwan Hassani and Andrea Burattin.

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

- *2nd International Workshop on Business Process Management and Routine Dynamics (BPM&RD 2022)* organized by Bastian Wurm, Thomas Grisold, Waldemar Kremser and Jan Mendling.

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

- *15th International Workshop on Social and Human Aspects of Business Process Management (BPMS2 2022)* organized by Rainer Schmidt and Selmin Nurcan.

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

- *1st International Workshop on Data-Driven Business Process Optimization (BPO 2022)* organized by Arik Senderovich, Remco Dijkman and Willem van Jaarsveld.

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

- *10th International Workshop on Declarative, Decision and Hybrid Approaches to Processes (DEC2H 2022)* organized by María Teresa Gómez-López, Claudio Di Ciccio, Tijs Slaats, and Jan Vanthienen.

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

- *1st International Workshop on Natural Language Processing for Business Process Management (NLP4BPM 2022)* organized by Han van der Aa, Manuel Resinas, Adela del Río-Ortega, and Henrik Leopold.

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

All workshops together received a total of 51 submissions. Each workshop had an independent Program Committee, which was in charge of selecting the papers for publication. The workshop papers received at least three reviews per paper. Out of the 51 submissions, 26 papers were selected to be presented at the workshops. Two of the accepted papers were not included in the final proceedings because the authors decided to withdraw them. Thus, the acceptance rate was 47%.

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

Cristina Cabanillas
Niels Frederik Garmann-Johnsen
Agnes Koschmider

Contents

6th International Workshop on Artificial Intelligence for Business Process Management (AI4BPM 2022)

Constraints for Process Framing in AI-Augmented BPM	5
<i>Marco Montali</i>	
Transformer Models for Activity Mining in Knowledge-Intensive Processes ...	13
<i>Faria Khandaker, Arik Senderovich, Eric Yu, Sebastian Carbajales, and Allen Chan</i>	
Automated Intelligent Assistance with Explainable Decision Models in Knowledge-Intensive Processes	25
<i>Alexandre Goossens, Ulysse Maes, Yves Timmermans, and Jan Vanthienen</i>	
The Label Ambiguity Problem in Process Prediction	37
<i>Peter Pfeiffer, Johannes Lahann, and Peter Fettke</i>	
Situation-Aware eXplainability for Business Processes Enabled by Complex Events	45
<i>Guy Amit, Fabiana Fournier, Lior Limonad, and Inna Skarbovsky</i>	

6th International Workshop on Business Processes Meet Internet-of-Things (BP-Meet-IoT 2022)

Assessing the Suitability of Traditional Event Log Standards for IoT-Enhanced Event Logs	63
<i>Yannis Bertrand, Jochen De Weerd, and Estefanía Serral</i>	
Method to Identify Process Activities by Visualizing Sensor Events	76
<i>Flemming Weyers, Ronny Seiger, and Barbara Weber</i>	
A Holistic Framework for IoT-Aware Business Processes	89
<i>Yusuf Kirikkayis, Florian Gallik, and Manfred Reichert</i>	
vAMoS: eVent Abstraction via Motifs Search	101
<i>Gemma Di Federico and Andrea Burattin</i>	

18th International Workshop on Business Process Intelligence (BPI 2022)

Mining for Long-Term Dependencies in Causal Graphs 117
Humam Kourani, Chiara Di Francescomarino, Chiara Ghidini, Wil van der Aalst, and Sebastiaan van Zelst

What Can Database Query Processing Do for Instance-Spanning Constraints? 132
Heba Amer, Marco Montali, and Jan Van den Bussche

2nd International Workshop on Business Process Management and Routine Dynamics (BPM&RD 2022)

Effects of Concurrency in Complex Service Organizations: Evidence from Electronic Health Records 149
Brian T. Pentland, Inkyu Kim, Quan Zhang, and Julie Ryan Wolf

15th International Workshop on Social and Human Aspects of Business Process Management (BPMS2 2022)

PYP4Training - Ludifying Business Process Training 167
Tatiane Neves Lopes, Renata Mendes de Araujo, Tadeu Moreira de Classe, and Thayná Gomes

On Current Job Market Demands for Process Mining: A Descriptive Analysis of LinkedIn Vacancies 179
Simin Maleki Shamasbi, Amy Van Looy, Barbara Weber, and Maximilian Röglinger

1st International Workshop on Data-Driven Business Process Optimization (BPO 2022)

Combining Process Mining and Optimization: A Scheduling Application in Healthcare 197
Matteo Di Cunzolo, Alberto Guastalla, Roberto Aringhieri, Emilio Sulis, Ilaria Angela Amantea, Massimiliano Ronzani, Chiara Di Francescomarino, Chiara Ghidini, Paolo Fonio, and Marco Grosso

Defining Process Performance Measures in an Object-Centric Context 210
Bedilia Estrada-Torres, Adela del-Río-Ortega, and Manuel Resinas

10th International Workshop on Declarative, Decision and Hybrid Approaches to Processes (DEC2H 2022)

Design-Time Support for Fragment-Based Case Management	231
<i>Kerstin Andree, Leon Bein, Maximilian König, Caterina Mandel, Marc Rosenau, Carla Terboven, Dorina Bano, Stephan Haarmann, and Mathias Weske</i>	
Process Mining Meets Statistical Model Checking: Towards a Novel Approach to Model Validation and Enhancement	243
<i>Roberto Casalupe, Andrea Burattin, Francesca Chiaromonte, and Andrea Vandin</i>	
A Systematic Comparison of Case Management Languages	257
<i>Julia Holz, Luise Pufahl, and Ingo Weber</i>	
Declarative Guideline Conformance Checking of Clinical Treatments: A Case Study	274
<i>Joscha Grüger, Tobias Geyer, Martin Kuhn, Stephan A. Braun, and Ralph Bergmann</i>	
Improving Declarative Process Mining with <i>a Priori</i> Noise Filtering	286
<i>Axel Kjeld Fjelrad Christfort, Søren Debois, and Tijs Slaats</i>	
1st International Workshop on Natural Language Processing for Business Process Management (NLP4BPM 2022)	
Text-Aware Predictive Process Monitoring with Contextualized Word Embeddings	303
<i>Lena Cabrera, Sven Weinzierl, Sandra Zilker, and Martin Matzner</i>	
PET: An Annotated Dataset for Process Extraction from Natural Language Text Tasks	315
<i>Patrizio Bellan, Han van der Aa, Mauro Dragoni, Chiara Ghidini, and Simone Paolo Ponzetto</i>	
Supporting Event Log Extraction Based on Matching	322
<i>Vinicius Stein Dani, Henrik Leopold, Jan Martijn E. M. van der Werf, and Hajo A. Reijers</i>	
Author Index	335

**6th International Workshop on Artificial
Intelligence for Business Process
Management (AI4BPM 2022)**

6th International Workshop on Artificial Intelligence for Business Process Management

As the popularity of Artificial Intelligence (AI) continues to grow, numerous novel methodologies and techniques emerge every year and are being applied across numerous areas. Recently, there is a strong interest from both industry and academia in applying AI methods in Business Process Management (BPM). The application of AI is impacting additional areas where process management perspectives become more relevant, including Industrial Engineering, IoT, and Healthcare. The use of AI in BPM has been discussed as the next disruptive technology that will touch upon almost all business process activities performed by humans. In some cases, AI will dramatically simplify human interaction with processes, while in other cases it will enable full automation of tasks that have traditionally required manual labor. We believe that over time, AI may lead to entirely new paradigms for business process management in all of its aspects: modeling, analysis, automation, implementation, and monitoring. For example, instead of BPM models centered around processes or cases, we anticipate models that are based fundamentally on goals. Moreover, these models will fully enable constant improvement and adaptation based on continuous experiential learning with little to none human intervention after the learning phase has been completed.

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

This year, a keynote speech by Marco Montali enriched the program of the workshop offering a view on the usage of constraints for process (re)framing. In addition, the workshop attracted 8 international submissions on different topics including activity mining, predictive process monitoring, explainability, fault detection, and knowledge graphs. All submissions were reviewed by at least 3 program committee members (or their sub-reviewers) and eventually 4 papers were accepted. We believe that the accepted papers provided an interesting mix of conceptual and technical contributions that are of interest for the AI4BPM community.

Pfeiffer et al. discuss the label ambiguity problem that one often meets when performing predictive process monitoring using ML techniques. They offer an empirical survey of the problem using open datasets from past Business Process Intelligence Challenges. Khandaker et al. propose the use of transformer models to enhance intent and activity recognition in emails to improve the skills of digital assistants and bots. In Goossens et al. the authors construct an intelligent assistant based on decision models. Specifically, they achieve their goal by employing decision modeling and notation (DMN) for enhancing the assistant with explainable recommendations. Lastly, Amit et al. used complex event processing together with methods

from business process management and explainable AI to achieve adequate explanations of process execution outcomes. Their solution enables us to hypothesize about any plausible causal situation to be examined for its possible effect on process execution outcomes.

November 2022

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

Organization

Program Committee

Han van der Aa	University of Mannheim, Germany
Annalisa Appice	University of Bari, Italy
Matteo Baldoni	University of Turin, Italy
Patrizio Bellan	FBK, Italy
Ralph Bergmann	University of Trier, Germany
Tathagata Chakraborti	IBM Research AI, USA
Marco Comuzzi	Ulsan Institute, Republic of Korea
Francesco Corcoglioniti	Free University of Bozen-Bolzano, Italy
Massimiliano de Leoni	University of Padova, Italy
Riccardo De Masellis	Uppsala University, Sweden
Ivan Donadello	Free University of Bozen-Bolzano, Italy
Joerg Evermann	Memorial University of Newfoundland, Canada
Stephan Fahrenkrog-Petersen	Humboldt University of Berlin, Germany
Francesco Folino	CNR, Italy
Rick Hull	New York University, USA
Krzysztof Kluza	AGH University of Science and Technology, Poland
Henrik Leopold	Kühne Logistics University, Germany
Francesco Leotta	Sapienza University of Rome, Italy
Lior Limonad	IBM Research AI, Israel
Roberto Micalizio	University of Turin, Italy
Marco Pegoraro	RWTH Aachen University, Germany
Luigi Pontieri	CNR, Italy
Jana-Rebecca Rehse	University of Mannheim, Germany
Andrey Rivkin	Free University of Bozen-Bolzano, Italy
Williams Rizzi	Fondazione Bruno Kessler, Italy
Tijs Slaats	University of Copenhagen, Denmark
Daniele Theseider Dupré	University of Eastern Piedmont, Italy
Hagen Voelzer	IBM Zurich Research Lab, Switzerland



Constraints for Process Framing in AI-Augmented BPM

Marco Montali^(✉) 

Free University of Bozen-Bolzano, Bolzano, Italy
montali@inf.unibz.it

Abstract. A recent research manifesto has introduced the vision of AI-Augmented BPM (ABPM), where BPM systems are infused with AI to continuously adapt and improve a set of business processes with respect to one or more performance indicators. In the ABPM lifecycle, process modelling is lifted to the more general notion of *process framing*, which aims at capturing the boundaries within which the executions of one or more processes of interest should be confined. In this paper, I argue in favour of constraint-based declarative process specifications for process framing. I provide a list of key features that are needed towards this goal, and show how they are matched by research milestones recently obtained in this setting. In particular, I discuss how to deal with deviations, uncertainty, and object-centric processes.

1 Introduction

The increasing availability of event data tracing the execution of business processes has led to a paradigm shift in business process management, moving from a pure model-driven approach to a balanced mix of model-driven and data-driven forces, fully incarnated by the *process mining* paradigm [3]. In parallel, artificial intelligence techniques are rapidly advancing, both in isolation and when it comes to their integration towards managing complex systems - as witnessed by the growing interest in *integrative AI*.

This provides a unique opportunity: the possibility of augmenting BPM with AI. A recent research manifesto describes the vision for *AI-augmented BPM* [14], where process-aware information systems become capable to dynamically unfold and adapt execution flows by exploring and enabling improvement opportunities, in autonomy and through continuous conversation with their human principals. As stated in [14], an *AI-augmented BPM system (ABPMS)* is

a process-aware information system that relies on trustworthy AI technology to reason and act upon data, within a set of restrictions, with the aim to continuously adapt and improve a set of business processes with respect to one or more performance indicators.

The lifecycle of an ABPMS expands that of a classical BPMS in two directions: on the one hand, the traditional lifecycle phases are continuously iterated, and infused with AI capabilities; on the other hand, the lifecycle includes

additional tasks that can only be realised with AI support, namely those of adaptation, explanation, and continuous improvement.

One particularly relevant aspect when transitioning from the lifecycle of a traditional BPMS to that of an ABPMS, is that process modelling is lifted to the more general notion of *process framing*. Framing aims at capturing the boundaries within which the executions of one or more processes of interest should be confined. This enables the key feature of *framed autonomy*: an ABPMS can autonomously decide how to progress the execution, as long as the boundaries imposed by the frame are respected.

In this respect, framing appears compatible with the way processes are captured using constraint-based, declarative approaches, such as *Declare* [24, 25] and *DCR Graphs* [18]. These approaches indeed enjoy flexibility by design by focussing on the elicitation of the relevant process boundaries, leaving the process executors free to decide how to behave as long as those boundaries are respected.

There are however several challenges that need to be tackled when declarative approaches are used for process framing. In this short paper, I concentrate on three particularly relevant challenges:

- the need of detecting and handling behaviours that break the boundaries, deviating from what is expected;
- the need of considering uncertain boundaries, where constraints may not necessarily always hold;
- the need of bringing data into the picture, in particular to handle the interaction of multiple processes simultaneously operating over different, related objects.

Using *Declare* as a reference framework for declarative process framing, I briefly discuss how such challenges are tackled by research milestones I achieved in collaboration with several colleagues. I finally conclude by recalling other challenges that I believe need to be faced next.

2 A Brief Recap of *Declare*

Declare is a language and notation for the declarative specification of processes based on temporal constraints. Every constraint separates conforming traces that satisfy the constraint, from non-conforming traces that violate it. A full specification consists of a set of constraints interpreted conjunctively, so that a trace is conforming to a specification if and only if it satisfies all constraints in the specification.

Every constraint has a semantics based on Linear Temporal Logic over finite traces (LTL_f) [11]. On the one hand, this provides a formal semantics for constraints, unambiguously characterising the notion of conforming trace to a specification. On the other hand, since every LTL_f formula can be captured as a finite-state automaton recognising all and only its conforming traces, automata-based techniques can be employed for reasoning, enactment, and a variety of

analysis and process mining tasks. In particular, every Declare specification can be encoded into a so-called *global automaton* that accounts for the constraints contained in the specification, as well as their interplay. Importantly, such global automaton can always be made deterministic, given the fact that the logic is interpreted on finite traces (this does not hold for infinite-trace logics).

For convenience, Declare comes with repertoire of pre-defined constraint templates, which can be grounded on concrete process activities. Each template has a notation that hides its LTL_f representation and that can be used to define specifications graphically.

An in-depth overview of Declare, its semantics, and the usage of automata for reasoning, enactment, analysis, and process mining, can be found in [13].

3 Boundaries that Can be Broken

The first challenge we need to tackle when framing a process is how to interpret the elicited constraints: are they hard constraints? Or soft constraints? Can they be violated, or are they satisfied by design? The answer is domain-specific, and has to be pondered constraint by constraint. In fact, the same Declare specification could mix constraints that account for physical rules, best practices, policies, norms, etc. For example, a Declare constraint stating that a car cannot start until one inserts the key is radically different from one indicating that a paid order should be later delivered. In this respect, Declare poses the same conceptual challenges that emerge in the usage of control-flow constructs in procedural process modelling languages [4].

The interpretation of constraints is further complicated by the fact that constraints that are hard when considered at the information system level may be soft in the real world, and vice-versa. For example, it may be impossible to indicate, inside the information system of an order-to-delivery company, that a cancelled order has been shipped, whereas this could happen in reality due to a human mistake. Even the distinction between the information system level and the real world is increasingly getting blurred, considering the fact that contemporary software is more and more directly connected to actuators in the real world [7].

I consequently bring forward a pragmatic approach for framing. At reasoning time, constraints are considered hard, to clearly distinguish conforming from nonconforming traces. At runtime, instead, every constraint can in principle be violated, which calls for techniques to *detect* violations as soon as possible, as well as for techniques to *react* to violations, by dynamically activating a different framing depending on which violations actually occurred. For example, the process may require that, whenever a shipped order is cancelled (violating a constraint that forbids to do so), a compensation payment must occur. The following question is then how to technically handle violation detection and reaction.

Detection is tackled by constructing a *monitor* for the Declare specification of interest [10,22]. This is done by modifying the construction of the global automaton in two ways.

First and foremost, every automaton state is labeled with a monitoring indicator, which combines past and future-tense reasoning by telling whether it is a satisfaction state or not, and whether continuing the execution from that state may or not potentially lead to a violation. This allows for the early detection of a violation that do not necessarily arise because the current execution is directly violating a single constraint, but does so when considering multiple constraints at once [9, 23].

Secondly, when composing the local automata encoding each constraint into the global automaton, no automata minimization nor trimming is applied. This guarantees that violation states corresponding to different violation conditions are separately represented, allowing the monitor to continue to provide a meaningful feedback even after a violation has been detected. If constraints are associated to penalties upon violation, then the aggregated, overall penalty can be computed per state, in turn providing the basis to reason on the penalties associated to different continuations of the current execution [5].

Reaction is tackled by defining special constraints (called *metaconstraints*) that can predicate on the satisfaction/violation of other constraints. This calls for using richer temporal logics over finite traces, with the good property that they can still be encoded into finite-state automata for reasoning and monitoring [10].

An ongoing study is to understand how these techniques can be lifted to reasoning and monitoring execution traces where events carry data attributes (such as strings, real/integer numbers, or more complex relations), and constraints are equipped with data-aware conditions. Some crucial technical results have been already obtained towards this goal [5, 8, 12, 16].

4 Boundaries that are Not Crisp

When framing a process, it may happen that some relevant constraints do not need to hold in all possible executions, but only in an expected ratio of them. For example, one may want to ensure that no more than 10% of the orders gets a fast-track shipping.

This triggers the need for dealing with uncertainty in framing, going beyond detecting and handling violations. In fact, in an uncertain setting the fact that a constraint is satisfied or violated may be perfectly acceptable, as long as this agrees with the degree of uncertainty of constraints. In particular, in the case where every constraint is subject to uncertainty, this in turn implicitly results in multiple frames, each describing a variant of the process together with its expected probability (or probability range). Following on the example before, a Declare specification containing only the aforementioned fast-track uncertain constraint would actually implicitly capture two variants of the process: one where a fast-track shipping is granted, and the other where this is instead not an option. Given a (multi)set of process traces, no more than 10% of them should belong to the first variant, which also implies that at least 90% should actually belong to the second.

Declare has been recently extended to deal with this form of uncertainty [5, 21], relying on a well-behaved fragment of a probabilistic variant of LTL_f [20]. Combined reasoning on probabilities and time can be applied to single out the different process variants (called scenarios in [5, 21]) and the possible probability masses they can be assigned to. This can then be used to relate traces as well as full event logs to process variants [5].

5 Boundaries on the Evolution of Multiple Objects

We have so far considered framing applied to a single process through constraints, where each execution characterises an instance of the process, implicitly evolving a single, case object. This assumption is typically too restrictive, as processes often need to co-evolve multiple objects connected through one-to-many and many-to-many relations. This is, for example, the case in an order-to-delivery process where multiple customer orders may result in multiple shipped packages, each containing items belonging to one or more orders.

These so-called *object-centric processes* are being increasingly studied in BPM and process mining [1, 15], with a variety of process modelling languages to capture them [2, 6, 17, 26].

In a declarative setting, shifting from a single-case to an object-centric perspective essentially amounts to move from a setting where constraints are global and implicitly apply to each single case and its evolution, to one where constraints are *scoped* by objects and their relationships, and locally apply to their instances. An example of object-scoped constraint is: if an order is open, then *that* order is later closed or cancelled. An example of relationship-scoped constraint is: if an order gets opened, then the customer *owning that* order is expected to sign a GDPR consent. Since each customer may own multiple orders, such scoping mechanisms ensure that each constraint is instantiated multiple times, and each constraint instance is evolved by activities that indirectly co-refer to each other because they operate either over the same object instance, or over related object instances. For example, when customer Jane signs a GDPR consent, all the instances of the relationship-scoped constraint above that refer to orders owned by Jane become satisfied (and so will, immediately, future orders opened by Jane).

This has been realised in the context of Declare through so-called *object-centric behavioral constraints* [6], which couple a Declare-like constraint specification over a set of activities with a UML-like data model over a set of classes and relationships, connecting these two components via object- and relationship-scoping mechanisms. Research on this fascinating approach is still at its infancy, with some initial results on discovery [19] and reasoning [6].

6 Conclusion and Further Challenges

We have reviewed how a variety of research milestones achieved in the area of constraint-based, declarative process modelling can provide a solid basis for

framing processes in an AI-augmented BPM system. In particular, we have discussed how to handle constraints that can be violated, that are uncertain, and that apply to multiple, related objects.

A number of further challenges need to be solved in this setting. I mention here three that are particularly pressing.

First of all, it is important to pair constraint-based framing with AI techniques that actually take decisions, autonomously and together with their human principals, on what to do next by exploring at best the conforming space defined by the frame. This calls for methods combining constraint-based framing with learning, strategy synthesis and goal-directed behaviour.

Secondly, one should consider that the activities to which constraints apply are often under the responsibility of different process stakeholders. For example, in an order-to-delivery process, the BPM system controls whether to accept or reject an order, while the customer decides whether an order gets paid or cancelled. This calls for adversarial synthesis and strategic reasoning over multi-party constraint-based process frames.

Finally, constraints should be explored by the system not only for enactment, but also to converse with human principals about the frame(s) they induce, which violations may occur, which continuations are the most promising, which process variant is likely emerging when choosing a course of execution, which objects and constraints are affected by the execution of an action, etc. This calls for integrating constraint-based reasoning and monitoring with conversational interfaces and natural language processing.

Acknowledgments. The author wants to thank all the people involved in the lines of research mentioned in this paper, and in particular: Wil van der Aalst, Anti Alman, Alessandro Artale, Federico Chesani, Giuseppe De Giacomo, Riccardo De Masellis, Claudio Di Ciccio, Marlon Dumas, Dirk Fahland, Paolo Felli, Alessandro Gianola, Alisa Kovtunova, Fabrizio Maggi, Andrea Marrella, Paola Mello, Jan Mendling, Fabio Patrizi, Rafael Penaloza, Maja Pesic, Andrey Rivkin, Michael Westergaard, and Sarah Winkler.

This work is partially supported by the Italian Ministry of University and Research under the PRIN programme, grant B87G22000450001 (PINPOINT), and by the UNIBZ projects SMART-APP and QUEST.

References

1. van der Aalst, W.M.P.: Object-centric process mining: dealing with divergence and convergence in event data. In: Ölveczky, P.C., Salaün, G. (eds.) SEFM 2019. LNCS, vol. 11724, pp. 3–25. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30446-1_1
2. van der Aalst, W.M.P., Berti, A.: Discovering object-centric Petri nets. *Fundam. Informaticae* **175**(1–4), 1–40 (2020)
3. 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

4. Adamo, G., Di Francescomarino, C., Ghidini, C., Maggi, F.M.: Beyond arrows in process models: a user study on activity dependences and their rationales. *Inf. Syst.* **100**, 101762 (2021)
5. Alman, A., Maggi, F.M., Montali, M., Patrizi, F., Rivkin, A.: Multi-model monitoring framework for hybrid process specifications. In: Franch, X., Poels, G., Gailly, F., Snoeck, M. (eds.) *CAiSE 2022*. LNCS, vol. 13295, pp. 319–335. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-07472-1_19
6. Artale, A., Kovtunova, A., Montali, M., van der Aalst, W.M.P.: Modeling and reasoning over declarative data-aware processes with object-centric behavioral constraints. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) *BPM 2019*. LNCS, vol. 11675, pp. 139–156. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_11
7. Baskerville, R.L., Myers, M.D., Yoo, Y.: Digital first: the ontological reversal and new challenges for information systems research. *MIS Q.* **44**(2) (2020)
8. Calvanese, D., De Giacomo, G., Montali, M., Patrizi, F.: Verification and monitoring for first-order LTL with persistence-preserving quantification over finite and infinite traces. In: De Raedt, L. (ed.) *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI 2022)*, pp. 2553–2560. ijcai.org (2022)
9. De Giacomo, G., De Masellis, R., Grasso, M., Maggi, F.M., Montali, M.: Monitoring business metaconstraints based on LTL and LDL for finite traces. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) *BPM 2014*. LNCS, vol. 8659, pp. 1–17. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10172-9_1
10. De Giacomo, G., De Masellis, R., Maggi, F.M., Montali, M.: Monitoring constraints and metaconstraints with temporal logics on finite traces. *ACM Trans. Softw. Eng. Methodol.* (2022, to appear)
11. De Giacomo, G., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: Rossi, F. (ed.) *IJCAI*, pp. 854–860. *IJCAI/AAAI* (2013)
12. De Masellis, R., Maggi, F.M., Montali, M.: Monitoring data-aware business constraints with finite state automata. In: Zhang, H., Huang, L., Richardson, I. (eds.) *ICSSP*, pp. 134–143. ACM (2014). <https://doi.org/10.1145/2600821.2600835>. <http://dl.acm.org/citation.cfm?id=2600821>
13. Di Ciccio, C., Montali, M.: Declarative process specifications: reasoning, discovery, monitoring. In: van der Aalst, W.M.P., Carmona, J. (eds.) *Process Mining Handbook*. LNBIP, vol. 448, pp. 108–152. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_4
14. Dumas, M., et al.: Augmented business process management systems: a research manifesto. *CoRR* abs/2201.12855 (2022). <https://arxiv.org/abs/2201.12855>
15. Fahland, D.: Process mining over multiple behavioral dimensions with event knowledge graphs. In: van der Aalst, W.M., Carmona, J. (eds.) *Process Mining Handbook*. LNBIP, vol. 448, pp. 274–319. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_9
16. Felli, P., Montali, M., Winkler, S.: Linear-time verification of data-aware dynamic systems with arithmetic. In: *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022)*, pp. 5642–5650. AAAI Press (2022)
17. Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: Petri net-based object-centric processes with read-only data. *Inf. Syst.* **107**, 102011 (2022)
18. Hildebrandt, T.T., Mukkamala, R.R.: Declarative event-based workflow as distributed dynamic condition response graphs. In: *PLACES*. *EPTCS*, vol. 69, pp. 59–73 (2010)

19. Li, G., de Carvalho, R.M., van der Aalst, W.M.P.: Automatic discovery of object-centric behavioral constraint models. In: Abramowicz, W. (ed.) BIS 2017. LNBIP, vol. 288, pp. 43–58. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59336-4_4
20. Maggi, F.M., Montali, M., Peñaloza, R.: Temporal logics over finite traces with uncertainty. In: Proceedings of the 34 AAAI Conference on Artificial Intelligence (AAAI 2020), pp. 10218–10225. AAAI Press (2020)
21. Maggi, F.M., Montali, M., Peñaloza, R., Alman, A.: Extending temporal business constraints with uncertainty. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 35–54. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_3
22. Maggi, F.M., Montali, M., Westergaard, M., van der Aalst, W.M.P.: Monitoring business constraints with linear temporal logic: an approach based on colored automata. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 132–147. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23059-2_13
23. Maggi, F.M., Westergaard, M., Montali, M., van der Aalst, W.M.P.: Runtime verification of LTL-based declarative process models. In: Khurshid, S., Sen, K. (eds.) RV 2011. LNCS, vol. 7186, pp. 131–146. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29860-8_11
24. Montali, M.: Specification and Verification of Declarative Open Interaction Models: A Logic-Based Approach. LNBIP, vol. 56. Springer, Heidelberg (2010). <https://doi.org/10.1007/978-3-642-14538-4>
25. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: EDOC, pp. 287–300 (2007)
26. Polyvyanyy, A., van der Werf, J.M.E.M., Overbeek, S., Brouwers, R.: Information systems modeling: language, verification, and tool support. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 194–212. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_13



Transformer Models for Activity Mining in Knowledge-Intensive Processes

Faria Khandaker¹(✉), Arik Senderovich², Eric Yu¹, Sebastian Carbajales³,
and Allen Chan³

¹ Faculty of Information, University of Toronto, Toronto, Canada
f.khandaker@mail.utoronto.ca, eric.yu@utoronto.ca

² School of Information Technology, York University, Toronto, Canada
sariks@yorku.ca

³ IBM Centre for Advanced Studies, Markham, Canada
{sebastia, avchan}@ca.ibm.com

Abstract. Mining useful information to analyze knowledge-intensive business processes requires data that describes activities of knowledge workers. Emails are widely used in organizations to provide support in the functioning of knowledge-intensive processes. The recent COVID-19 pandemic has increased reliance on technologies such as email to help facilitate communication within organizations to make up for the lack of face-to-face contact. In this work, we propose an activity mining technique, which receives an incoming email message, classifies the sender's intent and translates it into a set of business process activities. Specifically, we leverage deep learning language models to first classify the email body into a group of intents, which are then mapped to related activities. To our knowledge, we propose the first *transfer-learning* based solution for mining activity information from emails. The effectiveness of our solution was evaluated on real-world data coming from email exchanges between knowledge workers. Our results based on unsupervised experiments and a field study show that transformer models can be used to semantically label emails and that mapping activities to matched intents is highly accurate.

Keywords: Activity mining · Knowledge-intensive processes · Email intent classification · Email mining · Transfer learning

1 Introduction

Managing business processes is critical for various aspects of our lives: from the trucks that supply our cities with food to the doctors appointments we book over the phone. Some processes are highly structured, with every detail explicitly documented, and activities being well-ordered, e.g., in production and supply chain settings. Others, such as those present in knowledge work, are more flexible and dynamic and are often ad-hoc without a pre-defined order between the different activities [1]. Support for knowledge-intensive processes has been under investigation since the advent of email as email is the most common form of knowledge and information exchange in these types of processes [2, 3]. Moreover, during the COVID-19 pandemic, the need for collaborative workflow support has increased [4].

In this paper, we address the challenge of mining activities from emails. Solving this task can guide decision support systems and AI-based digital assistants in recommending the next best action(s) based on an incoming email [5]. The problem of extracting business process information from emails was studied in the past. van der Aalst et al. [6] presented the EmailAnalyzer tool which looks at email participants and tags in the subject line to create process mining logs to perform process discovery [7]. Others have attempted to extract business processes through investigating intent recognition, task recognition [8,9] and topic modelling [9–11] from the contents of the email body. However, these approaches were conducted with an abundance of manually labelled data. In practice, manually labelling every incoming and outgoing email is not scalable, and laws around privacy may hinder companies from taking and treating their existing emails as a training corpus for machine learning models [3].

To overcome the lack of labelled data, we mine *individual* email bodies for the intents and subsequently, activities that they contain using transformer models. The use of pre-trained transformer language models allows us to reconcile the limited availability of training data by leveraging the concept of transfer learning, i.e., the use of knowledge from one domain to complete tasks in another [12]. Furthermore, we define a generalized and flexible ‘world’ of intents (i.e., a general intent taxonomy) in which intent recognition could take place. Our framework where user intent acts as an intermediary for the appropriate task, allows for the most representative activities to be extracted.

To bring intent recognition and activity mining together we define two matching problems (email to intent and intent to activity) and provide an activity mining pipeline engineered to analyze individual emails for the sender intent (or intents) followed by the matching of activities to the identified intents. Our main contributions are threefold:

1. We present a general framework for activity mining from email data, which includes a novel intermediate knowledge layer, namely *intent*.
2. We introduce an unsupervised approach that uses transfer learning to extract intent information from emails.
3. We provide a weighted mapping between activities and intents based on an extensive field study.

Our results indicate that fine-tuned pre-trained transformer language models return more accurate intent matches than models which were not fine-tuned. Although the intent matching accuracy score is 35% (for our highest performing model), the activity mapping accuracy for the same model is 81%.

2 Background

In this part, we provide the background on the techniques we used to construct our solution, namely language models, zero-shot classification, and taxonomies in emails.

2.1 Language Models and Zero-Shot Classification

Language models allow for computers to ingest and analyze natural language to perform human-like tasks such as question-answering, text summarization, inference and sentiment analysis [13], to name a few. Language models like BERT [14], and BART [15]

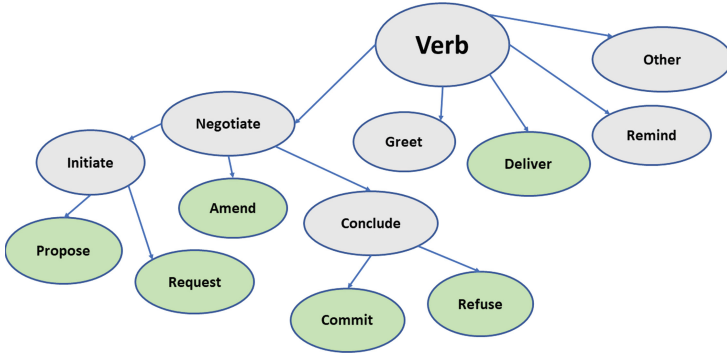


Fig. 1. The diagram of the verbs from the taxonomy detailed in Cohen et al. [18]. Green ellipses indicate verbs which were used to create our candidate intent labels.

are able to perform *transfer learning* which allows models to be trained on one large corpus from one domain but then be applied to a different domain or task [12]. Specifically, transfer learning allows for the learning of features without the need for annotated data to capture every possible example.

Zero-shot classification infers the relationship between a sequence and a label without prior supervision (the name comes from the fact that these methods observe zero training samples); the labels in the test set are completely different from those seen by the model during training. The model matches labels to the sequences in the test set through the use of transfer learning. Specifically, the inference is done by comparing the embeddings of the sequence and the labels using cosine similarity or other metrics [16]. Zero-shot classification (ZsC) has been successfully used for various applications such as computer vision [17]. In this work, we applied ZsCs to map emails to intents by using pre-trained Transformers; we refer to them as Zero-shot Transformers (ZsT).

2.2 Taxonomy and Recipient Intent

Taxonomy refers to the ordered arrangement of groups and categories. We draw inspiration from Speech Act Theory [19], to identify request and commitment utterances in email communication. The taxonomy that we adopt in this work was proposed by Cohen et al. [18]. Specifically, it consists of separate sub-taxonomies for verbs and nouns which describe words commonly used in email exchanges (see Fig. 1 for the verb part of the taxonomy). These include categories for request and commitment content, such as Request, Commit, Propose, Amend and Refuse for verbs; Activity, Information and Meeting are used for nouns.

3 Problem Formulation

We are aiming to answer two research questions, namely *how can one detect the sender intents contained in an email without access to an email corpus?* and, *once the sender*

intent is extracted, what is the activity associated with the intent? The latter question is translated, in our work, into mining the activity performed (or to be performed) based on the email received. Our solution uses intent as an intermediary to map email text to activities, since we assume that intents are the ‘hidden’ layer that motivates action. Below, we formulate the two problems that we are solving in this work.

We analyze the email bodies at the *sentence level granularity*, similar to other proposed intent and task-mining techniques (see [2,6]). Thus, we let \mathcal{S} be a universe of sentences that can appear in emails¹. Further, let \mathcal{I} be the universe of possible intents, e.g., as they are represented in the Cohen et al. taxonomy [20]. Next, let \mathcal{A} be a universe of possible activities that comprise the underlying business process.

Formally, given an email containing sentences $S \subseteq \mathcal{S}$, we wish to map every sentence $s \in S$ to a set of intents $I \subseteq \mathcal{I}$, and in turn, map each of the intents in I to a set of activities $A \subseteq \mathcal{A}$. In other words, we assume that sentences can contain multiple intents, which can then be mapped to multiple activities. In essence, we wish to solve a double matching problem, namely we are *classifying* sentences into intents, and then *mapping* intents into actions.

Definition 1 (The Problem of Intent Classification and Activity Mapping). *Given an email represented as a set of sentences S , find a pair of functions ϕ and θ that classify sentences in S into intents, and then maps intents into activities, i.e., $\phi : \mathcal{S} \rightarrow 2^{\mathcal{I}}$, $\theta : \mathcal{I} \rightarrow 2^{\mathcal{A}}$.*

For example, the sentence ‘Let us schedule a meeting on Monday at 10 AM to discuss the document you have sent’ can be classified onto the verb and noun labels of “proposing” and “meeting” respectively (see Fig. 1). These two intents can be then mapped onto an activity of ‘Set-up Appointment’.

4 Framework for Intent Classification and Activity Mapping

Figure 2 presents an overview of the general framework that we propose to solve Problem 1. When an email message enters the inbox, the body of the message undergoes preprocessing through the *Email Processor* module where it is cleaned (e.g., stripped of URLs, bullet points, special punctuation marks, etc.), and tokenized into sentences. These sentences are then fed into the *Intent Classifier*, along with a list of intent labels and each sentence is classified into a set of intents thus providing ϕ . The intents are then sent into the *Activity Mapper* module to be mapped to specific activities, providing θ . In total, we analyzed 214 sentences from two different email threads which together contained 64 different emails. Below, we provide some details into the main two components that instantiate the framework, namely the Intent Classifier and the Activity Mapper.

¹ Note that a sentence can be represented as a bag of words or a sequence; our problem formulation is agnostic to how sentences are defined.

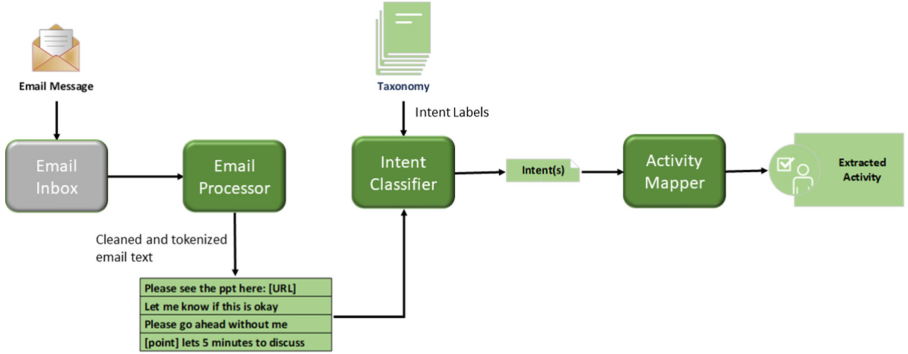


Fig. 2. The email intent classification and activity mapping solution.

Table 1. Examples of intent labels used to classify emails adapted from Cohen et al. [18] taxonomy.

#	Intent label
1	A meeting is being committed
2	A meeting is being proposed
3	A meeting is being refused
4	A meeting is being requested
5	An activity is being committed
6	An activity is being proposed

4.1 Mapping Sentences to Intents

Email data does not typically come with explicitly labelled intents, which prevents the use of supervised learning. To solve the problem in an unsupervised fashion, we first create a universe of intents, \mathcal{I} , from a well-known intent taxonomy. In this work, we used the Cohen et al. taxonomy [18], which is a pair of trees that describes the verbs and nouns frequently used in organizational emails (see Fig. 1 for the verb tree). To create \mathcal{I} , we flattened the taxonomy tree into the intent labels in Table 1 as our current approach does not support hierarchical information; we intend to add the hierarchy information in future work. To add context of how the noun and the verb should be understood together (example: deliver can refer to the giving of birth as well as the giving of an arbitrary object) the candidate intent labels are used to create present tense sentences. For example, instead of ‘Request Activity’, the sentence ‘An activity is being requested’ is used as the classification label.

Once we obtain the universe of intent labels from a user, we use pre-trained Zero-shot Transformer (ZsT) models to leverage transfer learning and match each sentence to our set of candidate intent labels. Specifically, given an email represented by a set of sentences $S \subseteq \mathcal{S}$, transformer models act as the main building block of ϕ , producing the bridge between a given sentence and a set of intents, namely $\phi(s) = I_s \subseteq \mathcal{I}, \forall s \in S$.

Table 2. Sample of recipient activities and their descriptions borrowed from the Lin et al. [8] analysis of the Avocado Email Corpus.

#	Activity labels	Descriptions
1	Reply-YesNo	Short Yes-no replies to questions
2	Reply-Ack	Acknowledgement such as ‘got it’
3	Reply-Other	Email replies without investigating some resources
4	Investigate	Gathering additional information to address issue
5	Send New-Email	Sending an email that is not a part of the current thread

Note that ZsC allows for a probabilistic approach to mapping sentences to intents. ZsC techniques return an ordered set of intent labels according to the probability of how well each intent matches the input sequence. Thus, we could use this probability distribution for the next step of activity mapping. Instead, we choose the top- K intents and plan to develop a probabilistic approach in future work.

Below, we provide the details of the models that we used to realize ϕ . Specifically, we applied four state-of-the-art Transformer models and reported on the best performing model. The models are:

1. The default Multi Natural Language Inference (MNLI) BART which was trained on 433000 sentence pairs annotated with textual entailment information through the MultiNLI corpus.
2. MNLI BART fine-tuned on Yahoo Answers, which is a variation of the default MNLI BART that has been fine-tuned on five Yahoo Answers categories: ‘Society and Culture’, ‘Health, Computers and Internet’, ‘Business and Finance’, and ‘Family and Relationships’.
3. Distil BART, a smaller and faster version of the default MNLI BART.
4. Distil BERT, a smaller and faster version of a BERT model and has been pre-trained on the MNLI corpus.

These models were selected based on the frequency of downloads (to show popularity) and the recency of updates (to show model maintenance). For the rest of the paper, the Yahoo Answers fine-tuned MNLI BART will be referred to as Yahoo Bart and the default Multi Natural Language Inference (MNLI) BART will be referred to as MNLI Bart. The output of the four models is an ordered list (most relevant to least) of the intent labels per input sentence.²

4.2 Mapping Intents to Activities

Once the intent set $I_s = \phi(s), \forall s \in S$ has been extracted, it needs to be further processed to be useful for the recipient. In other words, we need to create a mapping θ between every $i \in I_s$ and a set of activities to complete our solution. In this work, we considered the set of email activities proposed in [8] (see Table 2 for details). For

² For more information about these pre-trained models visit <https://huggingface.co/>.

example, intents can be mapped to possible activities such as ‘Reply-YesNo’, ‘Reply-Ack’, ‘Reply-Other’; each slightly differ from each other as mentioned in [8]. To create the mapping between intents and activities we conducted an extensive field study that involved knowledge workers who were asked to label intents from our taxonomy to a set of actions proposed in Table 2.

Field studies are a common practice in information systems and software development, c.f., [21]. Below, we provide further details about the field study. In order to create rules which define what activities are most likely to be associated with our intent labels, we asked six of our email thread participants to fill out a survey where each of our intent labels were shown and they were required to select all activities which they deemed relevant for said intent, based on their experiences in work settings. The email participants were selected to do the labelling to ensure that the labelling was done with as much context preserved as possible. Processing the results of the survey included assigning each activity with a ‘weight’ based on how many participants selected it to be relevant for the intent in question. For example, if all of our domain experts selected the activity “Reply Other” for the intent label “An opinion is being delivered”, then that activity would have a weight of 1, which is the highest weight possible according to our criteria. The weights of each activity per email body were summed together and ordered and the higher the summed weights of an activity, the more relevant they were thought to be to the email.

5 Evaluation

In this part, we present the empirical evaluation of our solution, demonstrating its relevance and capabilities to detect intents and map these intents to activities. In Sect. 5.1 we describe the use of real-world data coming from two chains of email exchanges between knowledge workers. To measure the accuracy of our approach we use both unsupervised and supervised learning metrics, which we present in Sect. 5.2. The main findings of the experiment are provided in Sect. 5.3.

5.1 Test Data

One of the motivations for our work is the lack of labelled email data (in real-world applications). However, in order to evaluate the accuracy of the Transformer models, we require a ground-truth dataset. Since the models used in our experiments are pre-trained, we only require a labelled test set. Yet, since all the datasets we used were unlabelled, we had to perform a labelling procedure. To this end, we conducted an additional labeling session with the participants of our field study, similar to the one described in Sect. 4.2, during which the six field study participants labelled all the sentences with the two most representative intents from our list of flattened intent labels (see Table 1). The results of this annotation were used as test data in our experiments.

5.2 Evaluation Metrics

We analyzed and evaluated the results using common metrics from unsupervised and supervised learning. Below, we provide an overview of the two types of evaluation measures.

Unsupervised Evaluation Metrics. For unsupervised evaluation, clustering was performed on the sentence embeddings. The sentence embedding pipeline from the Sentence Transformers Python library generated these embeddings using each of the ZsT models outlined in Sect. 4.1. Each sentence vector was clustered using the Hierarchical Density-based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm, since the more well-known K-means and Agglomerative clustering techniques are too sensitive to outliers and require the desired number of clusters to be known in advance [9]. The quality of the clusters were evaluated using both the Silhouette Distance Score (SDS) and the Calinski-Harabasz Index (CHI) (see [22]). The SDS measures the closeness of the points in the same cluster: scores close to 1 indicate clusters are tightly packed, scores close to 0 indicate overlapping clusters and scores close to -1 indicate that data points are assigned to the wrong clusters, meaning a different cluster is more similar. CHI looks at the sum of the intra-cluster and inter-cluster dispersion; the higher the score, the more dense and well separated the clusters are although there is no agreed upon 'cut off' point [22].

Supervised Evaluation Metrics. Supervised evaluation was performed by comparing the annotated ground truth sentence intents with those predicted from the four Transformer models. Measurements of Accuracy, weighted Precision, Recall, and F1-Score are common evaluation metrics for Machine Learning models and these metrics were used to evaluate the frequency of misclassified intents. Similarly, the activity mining evaluation consisted of measuring the accuracy between activities mapped to the ground truth intents and those mapped to the predicted intents.

5.3 Results

In the experiment, we applied each of the ZsT models we mentioned in Sect. 4.1, and then used the results of the field study (mentioned in Sects. 4.2 and 5.1) to measure the accuracy of intent classification and activity mining. Below, we report the main findings of the experiment.

Unsupervised Intent Mining. The results of our unsupervised comparison are presented in Table 3. Overall for unsupervised evaluation methods, DistilBERT and Yahoo BART had the best performance among the models with the highest SDS and CHI. This strongly indicates that one should consider one of these two ZsT models when applying our framework. The hovering of the SDS around the zero mark (see Table 3) indicate loose and slightly overlapping clusters which confirms our assumption that a single sentence can contain more than one intent. For example a 'meeting' can be considered an 'activity' and may be grouped together despite 'meeting' having different contextual meaning than 'activity'.

The slightly negative SDS for some metrics indicate that similarities between clusters caused an overlap of data-points, which demonstrates that there are subtleties in written language that cannot be determined purely semantically and require broader organizational context for an accurate intent classification methodology.

Table 3. Clustering metrics of the four different language models tested.

Model type	SDS	CHI
Yahoo BART	0.265	189.106
DistilBERT	0.363	183.509
MNLI BART	-0.055	86.951
DistilBART	-0.246	38.481

Table 4. The Accuracy, Precision and Recall metrics of the four different language models tested.

Model type	Accuracy	Weighted F1-score	Weighted precision	Weighted recall
Yahoo BART	0.352804	0.200777	0.230642	0.193925
MNLI BART	0.334112	0.177900	0.268956	0.168224
DistilBART	0.250000	0.114641	0.360935	0.130841
DistilBERT	0.231308	0.101750	0.194078	0.128505

Supervised Intent Mining. The test data (ground truth) obtained from the field study described in Sect. 5.1 were compared against the machine predicted labels. The results for intent classification are presented in Table 4. The MNLI BART model performs better than other models, in contrast to the clustering results. However, the Yahoo BART is the highest performing in terms of accuracy scores. The results are consistent with the unsupervised clustering results observed in Table 3. This indicates that Yahoo BART is especially suitable for analyzing email intents. The results obtained from the supervised evaluation metrics are encouraging as previous experiments have achieved similar accuracy metrics in other domains [23, 24]. Moreover, the Yahoo BART model whose fine tuning set contained business and finance specific examples outperformed all other models. This makes us hopeful that fine-tuning using specific business email intent examples can improve the accuracy and precision scores.

Supervised Activity Mining. The results for activity mapping are presented in Table 5. The top three activities that were mapped by our participants to the ground truth intents were ‘Reply-Other’, ‘Reply-YesNo’, and ‘Investigate’. The same activities were also mapped for the DistilBert model. For MNLI BART, Yahoo BART and DistilBART the top three mapped activities were ‘Reply-Other’, ‘Reply-Ack’, and ‘Reply-YesNo’ with ‘Investigate’ being the fourth most common activity found in the email chains analyzed. This is an interesting result as it shows the consistency with which activities are connected to various intents. On average Chain 2 had lower scores across all models compared to Chain 1, which could be due to the fact that Chain 2 contains more emails compared to Chain 1. Additional explanation could be that there is much more discussion involved in creating a video (topic of Chain 2) than there is for creating a slide deck (topic of Chain 1). We detect an improvement in detecting activities compared to the accuracy scores for the intents (see Table 4). Experiments conducted by Wang et al. [6] showed that emails could contain three or more intents, although, 1–2 intents was more

Table 5. The Accuracy of tasks when compared to the ground truth in two email chains. Chain 1 is correspondence related a slide deck; Chain 2 discusses a video clip.

Email chain	Activity found by	Accuracy
Chain 1	MNLI Bart Intents	0.851852
	Yahoo Bart Intents	0.814815
	DistilBart Intents	0.814815
	DistilBert Intents	0.703704
Chain 2	MNLI Bart Intents	0.742690
	Yahoo Bart Intents	0.801170
	DistilBart Intents	0.725146
	DistilBert Intents	0.631579

common. This, together with our results demonstrate that activity mapping can ‘mask’ the noise that stems from intent classification as there is a potential for many (intents) to one (activity) relationship. In other words, emails with multiple intents can often lead to the performance of similar activities. Although the activities used for the mapping are very general, they are representative of how emails are used in organizations. The specificity of activities could potentially be increased with the help of additional metadata such as software available to the user.

6 Related Work

Email Mining in Business Process Management. The MailofMine framework by DiCiccio et al. [25] was one of early works that presented an end-to-end solution for extracting declarative business processes from email data. With recent advancements in NLP and Deep Learning, Jlalaty et al. [11] proposed the extraction of business activities from emails through use of labelled sentences, process model repositories, cosine similarity, word embeddings and clustering methods. Recently, Chambers et al. [9] extended the aforementioned solutions by using unsupervised machine learning algorithms to extract actions and process models from emails. Elleuch et al. [10] also take an unsupervised approach to discovering activities in emails, but through the use of custom developed algorithms which uses the context of the words in an email to derive the idea of a concept pattern. These works rely on the analysis of data that contains specific process-related jargon and labelled data, while our solution is domain independent and can work without supervision for the intent classification task.

Email Mining in Machine Learning. Early works in the field of machine learning tried to extract features from emails to automate task extraction via feature construction and training an SVM to distinguish between tasks and non-task sentences [2]. Lin et al. [8] and Wang et al. [6] use neural networks to predict recipient and sender intent. Alibadi et al. [12] tested various language models including Word2Vec, ELMo, BERT, Deep Averaging Network Based Sentence Encoder, and Transformer Based Sentence Encoder to

classify email sentences with either a ‘to-do’ intent or a ‘to-read’ intent. A recent study leveraged the use of ‘weak labels’ which are created from email interactions through specific labelling functions [26]. This was done to overcome the problem of limited annotated data, which our paper also works to address. However, unlike our work, the authors of [26] proposed manually curated if-else statements (based on replies in email threads) to create weak labels whereas we take advantage of pre-trained models.

7 Conclusion

In this paper we presented an intent classification and task mapping solution for email data. Our method combines zero-shot intent classification with a rules-based task mapping to form a domain agnostic solution for understanding the sender’s intent and translating it into actionable tasks. To quote Judea Pearl, ‘...strong AIs will certainly need to understand the vocabulary of options and intents...’ [27]. Thus, we believe that the work is a stepping stone to a solution where email threads are translated into useful information that would be used by AI digital assistants at our workplaces. Our experimental results show that the MNLI BART model fine tuned on a Yahoo Answers corpus was the top performing model. These results are encouraging as we see that one of the categories the Yahoo BART model is trained on is ‘Business and Finance’ and it supports the idea that a fine tuning a model using domain specific data can increase accuracy and precision of the model. Our results demonstrate that state-of-the-art machine learning tools enable intent and activity extraction with minimal supervision. Therefore, we believe that in future iterations, one would be able to perform process discovery based on the extracted activities, which can then be extended to support other technologies such as recommendation systems, digital assistants and RPA bots. Another future direction is to integrate a human-in-the-loop component into our pipeline where the user will be prompted to write short test emails per intent label to help the language model tune itself. This can help users determine the next best task for business purposes.


References

1. Dustdar, S., Hoffmann, T., Van der Aalst, W.: Mining of ad-hoc business processes with teamlog. *Data Knowl. Eng.* **55**(2), 129–158 (2005)
2. Corston-Oliver, S., Ringger, E., Gamon, M., Campbell, R.: Task-focused summarization of email. In: *Text Summarization Branches Out*, pp. 43–50 (2004)
3. Stuit, M., Wortmann, H.: Discovery and analysis of e-mail-driven business processes. *Inf. Syst.* **37**(2), 142–168 (2012)
4. Bloom, N.: How working from home works out. *Stanford Institute for Economic Policy Research*, pp. 1–8 (2020)
5. Heavin, C., Power, D.J.: Challenges for digital transformation-towards a conceptual decision support guide for managers. *J. Decis. Syst.* **27**(sup1), 38–45 (2018)
6. Wang, W., Hosseini, S., Awadallah, A.H., Bennett, P.N., Quirk, C.: Context-aware intent identification in email conversations. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 585–594 (2019)
7. van der Aalst, W.M., Nikolov, A.: EmailAnalyzer: an e-mail mining plug-in for the prom framework. *BPM Center Report BPM-07-16*, BPMCenter.org (2007)

8. Lin, C.C., Kang, D., Gamon, M., Pantel, P.: Actionable email intent modeling with reparametrized RNNs. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
9. Chambers, A.J., et al.: Automated business process discovery from unstructured natural-language documents. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) BPM 2020. LNBIP, vol. 397, pp. 232–243. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66498-5_18
10. Elleuch, M., Ismaili, O.A., Laga, N., Gaaloul, W., Benatallah, B.: Discovering activities from emails based on pattern discovery approach. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNBIP, vol. 392, pp. 88–104. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58638-6_6
11. Jlalaty, D., Grigori, D., Belhajjame, K.: On the elicitation and annotation of business activities based on emails. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 101–103 (2019)
12. Alibadi, Z., Du, M., Vidal, J.M.: Using pre-trained embeddings to detect the intent of an email. In: Proceedings of the 7th ACIS International Conference on Applied Computing and Information Technology, pp. 1–7 (2019)
13. Radford, A., et al.: Language models are unsupervised multitask learners. *OpenAI Blog* **1**(8), 9 (2019)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
15. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint [arXiv:1910.13461](https://arxiv.org/abs/1910.13461) (2019)
16. Brown, T.B., et al.: Language models are few-shot learners. arXiv preprint [arXiv:2005.14165](https://arxiv.org/abs/2005.14165) (2020)
17. Romera-Paredes, B., Torr, P.: An embarrassingly simple approach to zero-shot learning. In: International Conference on Machine Learning, pp. 2152–2161. PMLR (2015)
18. Cohen, W., Carvalho, V., Mitchell, T.: Learning to classify email into “speech acts”. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 309–316 (2004)
19. Searle, J.R., Searle, J.R.: *Speech Acts: An Essay in the Philosophy of Language*, vol. 626. Cambridge University Press (1969)
20. Carvalho, V.R., Cohen, W.W.: Learning to extract signature and reply lines from email. In: Proceedings of the Conference on Email and Anti-Spam, vol. 2004 (2004)
21. El Emam, K., Madhavji, N.H.: A field study of requirements engineering practices in information systems development. In: Proceedings of 1995 IEEE International Symposium on Requirements Engineering (RE 1995), pp. 68–80. IEEE (1995)
22. Wang, X., Xu, Y.: An improved index for clustering validation based on silhouette index and Calinski-Harabasz index. In: IOP Conference Series: Materials Science and Engineering, vol. 569, p. 052024. IOP Publishing (2019)
23. Yin, W., Hay, J., Roth, D.: Benchmarking zeroshot text classification: datasets, evaluation and entailment approach. arXiv preprint [arXiv:1909.00161](https://arxiv.org/abs/1909.00161) (2019)
24. Sappadla, P.V., Nam, J., Mencía, E.L., Fürnkranz, J.: Using semantic similarity for multi-label zero-shot classification of text documents. In: ESANN (2016)
25. Di Ciccio, C., Mecella, M.: Mining artful processes from knowledge workers’ emails. *IEEE Internet Comput.* **17**(5), 10–20 (2013)
26. Shu, K., Mukherjee, S., Zheng, G., Awadallah, A.H., Shokouhi, M., Dumais, S.: Learning with weak supervision for email intent detection. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1051–1060 (2020)
27. Pearl, J., Mackenzie, D.: *The Book of Why: The New Science of Cause and Effect*. Basic Books (2018)



Automated Intelligent Assistance with Explainable Decision Models in Knowledge-Intensive Processes

Alexandre Goossens^(✉) , Ulysse Maes, Yves Timmermans,
and Jan Vanthienen 

Leuven Institute for Research on Information Systems (LIRIS), KU Leuven,
Leuven, Belgium
alexandre.goossens@kuleuven.be

Abstract. Predictive monitoring techniques increasingly contain explainability aspects to instill trust in the end-users. However, it currently remains difficult to clearly communicate to the end-user how and why a certain outcome is obtained in a knowledge-intensive process. One improvement is the representation of decisions using executable Decision Model and Notation (DMN) models. These allow to automatically construct an intelligent assistant that reasons directly with any DMN model to provide such explanations. This paper examines the added value of a generic intelligent assistant, based on a DMN model of a decision. A preliminary experiment was conducted using two different explanation sources (text and intelligent assistant) to evaluate the explanation facilities of an automated DMN intelligent assistant. The first findings from this ongoing research provide insights into how organizations could easily provide stakeholders with explainable decisions in processes.

Keywords: DMN · Explainability · Intelligent assistants

1 Introduction

Despite the digital transformation and automation of day-to-day processes, people still have many questions regarding applications, procedures or decisions. Organizations struggle with these requests and redirect these to call centers, FAQs, information websites, and chatbots. Sadly, these costly initiatives do not always meet customer demands with long waiting times and human interventions [6]. This hidden service cost is unfortunate as it is crucial to improve processes. Thankfully, decisions can be represented using Decision Modeling and Notation (DMN) models with decision requirements diagrams and decision tables [12]. Decisions have been identified as an important part of processes, especially of knowledge-intensive processes. Recent research indicates that decisions should be modeled independently according to the principle of separation of concerns [1, 16]. In a context where explainability has been getting more attention by various studies or manifestos [3, 7, 8, 11], decision models are currently not being used

to their full potential. In previous work [5], we used DMN models not just to make a decision in a knowledge-intensive process, but also as a knowledge base to automatically create intelligent assistance for various decision related questions. Using various Artificial Intelligence (AI) forms (Natural Language Processing (NLP), knowledge reasoning and decision modeling) in one intelligent assistant, this paper studies the explanation quality of such an intelligent assistant and if it could support/replace other forms of explanation (e.g. web pages, FAQs).

The paper starts with Sect. 2 which introduces the problem and the research hypotheses. Section 3 provides background regarding DMN. In Sect. 4, an overview of assistance scenarios is given that can take place in a knowledge-intensive business process. Section 5 discusses related work and in Sect. 6, the experiment methodology can be found. In this preliminary experiment, the participants had to answer various explanation questions using either a textual description or an intelligent assistant. For each participant, the answers were evaluated, the time spent with each explanation source was also measured. Lastly, the participants had to indicate their preferred explanation source. The results are presented in Sect. 7. In Sect. 8, the results are discussed as well as the limitations and future work. Section 9 concludes this paper.

2 Problem Statement and Research Hypotheses

The problem is stated with a textual description of an adapted dinner-beverage allocation problem [2] together with a few questions the end-user might have.

Dinner-Beverage Allocation: *If the season is Fall and there are 8 or less guests, then they will eat spareribs. If the season is Winter and there are 8 guests or less, then they will eat roast beef. However if the season is Spring and there are less than 5 guests then they eat Dry Aged Gourmet steak and if there are between 5 or 8 (8 included) guests the served dish is Steak. If there are more than 8 guests and it is not Summer, then they eat Stew. Finally, if it is Summer regardless of the number of guests, they will always eat a light salad and a nice steak. If the guests have children with them, then everyone drinks apple juice. If there no children present, the beverages are adapted to the meal. With spareribs, Aecht Schlenkerla Rauchbier is offered. With Stew, a nice Guinness is provided and roast beef is accompanied with Bordeaux. All the other dishes are provided with Pinot Noir.*

The text in this example can be used to make a decision in a process model. But there is much more to it: A few additional questions that might arise are:

- Why is the decision: Steak?
- What do we eat if I invite 15 instead of 6 people in Summer?
- When can I drink Pinot Noir?

General Problem: The above questions are highly important to the user as they explain how a decision or process is performed. Especially the service industry is confronted with this kind of questions as everyone wants to optimize their outcome. Sadly, in a real process, e.g. a student grant allocation, it is not possible to ask all these questions as the decision is often merely executed and only the final decision is communicated. If more explanation is desired, one could

- re-execute the (lengthy?) process with different values
- dive into the technical bureaucratic details of grant allocations
- call an expensive helpdesk/ employee
- use an intelligent assistant if the process contains DMN decisions

Research Hypotheses: This on-going research investigates if the explanation provided by an intelligent decision assistant is comparable to explanations provided by webpages, FAQs or even human-staffed helpdesks. For this preliminary experiment, textual descriptions are used as a proxy for FAQs, webpages and other textual explanation sources. This preliminary experiment focuses on three variables: **Explanation quality** (correctness of the retrieved information); **Explanation speed** (elapsed time to find an explanation); **User preference** (preferred explanation source indicated by the participant).

These variables provide a good initial view of what a digital assistant can mean to the users. The intelligent decision assistant is not compared to just providing DMN models as these can be complex and are hardly ever provided to the customer anyway. Additionally, the intelligent decision assistant is already an enactment of a DMN model. **The following hypotheses are elicited:**

1. The quality of the explanation provided by the intelligent chatbot or by the textual description does not differ between the two.
2. The explanation retrieval speed is faster for intelligent assistants than for textual descriptions.
3. The end-users prefer to use intelligent assistants over textual descriptions.

3 Background

In 2015, the Object Management Group (OMG) introduced the DMN standard to model, communicate and execute operational decisions [12]. Figure 1 shows the DMN model of the beverages allocation problem. The decision structure is visualized with a Decision Requirements Diagram (DRD) with rectangles being decisions (e.g. Determine Dish) and rounded rectangles as inputs for decisions (e.g. Number of guests). Inputs are linked to decisions with information requirements (solid arrows). Outputs of so-called intermediary decisions can also be used as inputs for other decisions e.g. Determine dish. The decision logic is contained in decision tables with decision rules that are read as if-then statements. This format easily allows for consistency and completeness of the rules [17].

4 Assistance Scenarios

4.1 Execution Scenarios

Execution scenarios determine which process path is taken by deriving an output from certain inputs based on rules. **Execution scenarios 1 and 2** are identified with only **scenario 1** mostly being supported by process execution engines.

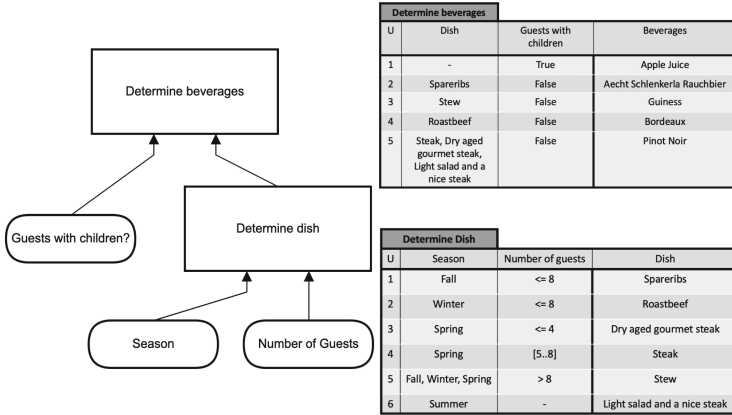


Fig. 1. DMN model to determine beverages

1. **All input information is provided:** *Which beverage will be served if it is spring and there are 6 guests out of which 2 kids?*
2. **Only relevant input information is provided:** *What will we drink in summer and without kids?* Sometimes, a final decision can be reached without needing all the information.

4.2 Explanation Scenarios

Explanation scenarios provide further clarification to how or why a certain decision was made. The explanation is already contained in the decision rules. This section elaborates on **seven explanation scenarios (3–9)**.

A: Reasoning from inputs Scenarios **3** and **4** provide initial explanations using only inputs allowing users to have an initial assessment without even gathering all the information yet.

3. **Input information is partially unknown:** *Which beverage is served if it is spring and we expect 6 guests?* Depending on whether there are children, different decision rules apply. It is important that processes can deal with unknown information so that the potential outcomes are already known.
4. **Partial decision making with incomplete input information:** *What dish is served in Winter with 4 guests?*

Intermediate outputs can be just as insightful as final outputs to show the progress in decisions/processes.

B: Reasoning from outputs When reasoning from outputs, the point is to understand how to achieve that output. This is done by reasoning backwards with the decision rules. **Scenarios 5 and 6 reason with outputs:**

5. **Outcome-driven reasoning** *How can I make sure to drink “Guinness”?* By providing the specific decision rules leading to that outcome, the user can assess for themselves whether it is achievable prior to the process execution.

6. **Optimization** *How can I drink “Guinness” if currently “Apple Juice” will be served?*

C: Reasoning with both inputs and outputs

7. **Output explanation** *Why are we drinking “Apple Juice”?*

To increase trust, a user should be able to know how a certain decision was made, otherwise the process might seem random and unfair towards the user. Providing the concerned decision rule(s) explains the reasoning to the user.

8. **What if** *What beverage will be served if 3 guests canceled (originally 10 guests were invited), no children will be present and the party is in Spring?*

A fast feedback mechanism allows the user to quickly understand what is controllable and what is not, without resubmitting all their information.

9. **Sensitivity** *How many guests can I invite in Summer to keep drinking “Pinot Noir” if right now we are 6 guests?* By knowing which variables can be changed without impacting the outcome, companies do not have to reprocess everything every time information is updated.

In short, explanation scenarios can reason with both inputs and outputs and do not limit themselves to simply deriving an output. Where some scenarios provide useful information before the process execution and others provide useful information afterwards, providing explanations at different points in consultation/execution of the process decision can greatly improve the quality of the process. Figure 2 visualizes the scenarios. The arrows indicate the order in which the assistance scenarios can follow each other. Before any explanation can be provided, it is important that the inputs or outputs are known to the system with scenarios 1,2,3 and 4 (with scenario 4 providing intermediate outputs). With this information, the user can trigger various explanation scenarios if enough information has been provided.

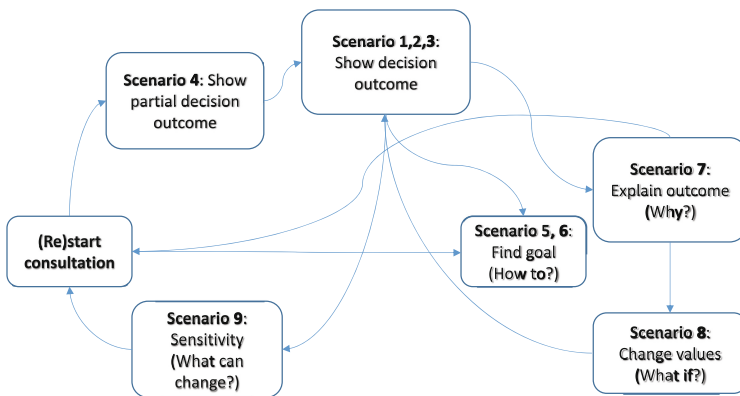


Fig. 2. Visualization of assistance scenarios

5 Related Work

Explainability is increasingly being studied within the context of business processes e.g. [7, 8, 11] and its importance highlighted in [14]. In [13], a user evaluation of explainability techniques concludes that explanation plots are useful but not insightful enough to perform a what-if analysis with enough confidence.

To explain a process it is also possible to directly use the process model itself such as in [9, 10] where a chatbot guides a user through the process steps. In our previous work a preliminary chatbot with a custom-built reasoning mechanism is introduced [5] that reasons with any DMN model. Such a chatbot can have its own reasoning mechanisms or can be connected to other reasoning mechanisms. In [15], the Imperative/Declarative Programming (IDP) reasoning mechanism API is introduced together with a small naive (in the sense of absent AI capabilities e.g. intent interpreter, NLP or voice support, user interface) DMN chatbot prototype whilst in [4], a DMN chatbot is introduced using the Camunda reasoning mechanism. The first two DMN approaches offer more reasoning capabilities compared to the last approach. This first on-going research evaluates an improved DMN chatbot on the correctness of the information retrieved and information retrieval speed.

6 Methodology

6.1 Chatbot Prototype Info

This section presents the extended DMN based chatbot prototype used for the experiment. It is able to answer more questions compared to the chatbot introduced in [5]. Figure 3 visualizes the following components: (1) Dialog Engine; (2) Knowledge Reasoner; (3) DMN model as knowledge source; (4) User Interface.

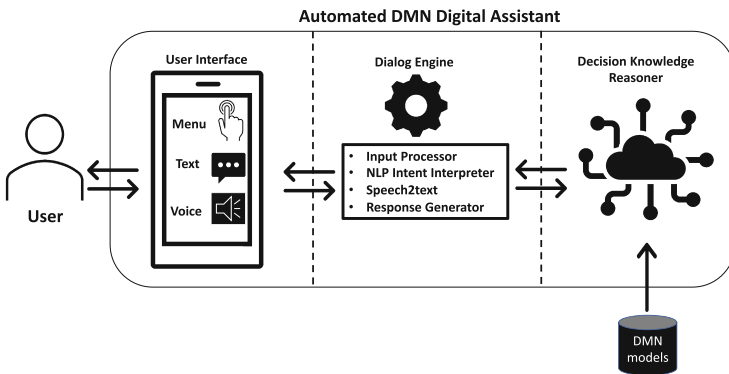


Fig. 3. Automated DMN digital assistant framework

Dialog Engine: The dialog engine guides the conversation. A more refined way is to make use of an AI intent interpreter e.g. Language Understanding (LUIS) from Azure¹ that understands a wider array of sentences describing the same intent. The dialog engine has an internal mapping that links each intent to a certain scenario allowing to communicate the correct scenario to the knowledge reasoner. With the feedback provided by the knowledge reasoner, the decision engine only asks relevant questions concerning the relevant decision rules.

Knowledge Reasoner: The knowledge reasoner interacts with any DMN model. The prototype used in this experiment uses a reasoner based on IDP [15]. The knowledge reasoner is connected to the dialog engine that provides the inputs and the desired execution scenario provided by the user. Once the desired scenario is executed, the relevant answers are sent to the dialog engine. If the reasoner needs more information, it will request the dialog engine to retrieve it from the user. Currently, the reasoner only supports the scenarios in Sect. 4.

DMN Model as Knowledge Source: In the chatbot prototype, DMN models are used as knowledge base to reason and provide an answer. These models have the advantage of being interpretable as well as executable. Therefore, there is no need to hard-code the knowledge elsewhere (in chatbots, FAQs, ...) as only a general reasoner is required to reason with the decision rules of a DMN model.

User Interface: The user interface is the only way the user can interact with the dialog engine. Figure 4 visualizes the user interface. There are three interaction methods with the dialog engine:

1. **Menu:** Even though a menu limits the ways to trigger a certain scenario, it reduces the chances of wrong input or unforeseen user interaction. When only a menu is supported, it can not truly be considered a chatbot as the user can not provide sentences to get information.
2. **Text:** When NLP is supported, the user can interact using written text. To understand what the user wants, the intelligence of the assistant can also be extended with an AI intent interpreter such as LUIS.
3. **Voice:** It is also possible to support speech recognition where the user communicates by speech and a speech2text API e.g. IBM Watson² converts it to text for further processing. It is also possible to provide a spoken answer back to the user using a text2speech API.

6.2 Experiment Information

Experimental Set-Up: Two problems were translated into a textual description and a corresponding DMN model was provided to the chatbot prototype. The dinner problem was introduced in Sect. 2 and its textual description and DMN model have both been used in the experiment. The other problem deals

¹ www.luis.ai/home.

² <https://www.ibm.com/cloud/watson-assistant>.

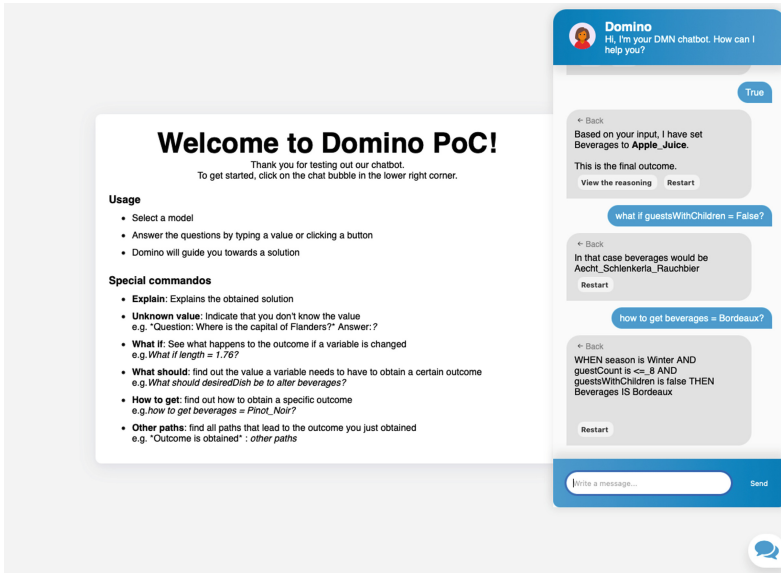


Fig. 4. Chatbot user interface and guidance for the experiment

with a BMI (Body Mass Index) determination problem, its textual description is similar to the dinner problem one. These examples (although rather simple) were chosen as they are easy to understand so that a comparison between explanation sources could be done more efficiently. To make the findings more generalizable, participants with different levels of knowledge about decision modeling and Business Process Modeling (BPM) were selected.

Questionnaire: The following questions (from both examples) were asked with different values for each explanation format (text, chatbot):

1. **Scenario 1:** What is the BMI-level of a female of 65 kg and 1.68 m?
2. **Scenario 3:** Which beverages can be served in Summer with 9 guests?
3. **Scenario 5:** How many guests can be invited if I want to drink Pinot Noir in Spring?
4. **Scenario 8:** What if my son loses weight from 40 kgs to 35 kgs. He is currently 1.45 m high. What will his BMI-level be? (e.g. Underweight)
5. **Scenario 9:** How many guests can I invite to still drink Bordeaux if right now we are 5 guests?
6. **Last question:** Which explanation source did you prefer to get explanations from (text or chatbot)?

The questions would ask the same scenarios but with different values so that the participants had to think with each explanation format without knowing the answer from an earlier format. This was important to compare between explanation sources. Next, the order of explanation sources provided to the participant was randomized to reduce bias towards a source. The intelligent decision assistant prototype was accompanied with a very small manual (left-hand side of

Fig. 4) explaining how the scenarios could be triggered. Lastly, not all assistance scenarios were asked as it does not reflect a real use case. Only scenarios that are variations of those scenarios that were already asked in the questionnaire were left out. The time to answer questions for each explanation format and problem was measured as well. The experiment was conducted using Google forms.

7 Insights from the Experiment

Sample Information: In this preliminary experiment, 25 persons from various backgrounds participated with different levels of knowledge about decision modeling and BPM. The sample size is too small to conduct a statistical analysis for now, but as the research continues this is planned in the future.

Below, we provide the results for each hypothesis introduced in Sect. 2.

Answer to Hypothesis 1: It is important to point out that both the chatbot and the textual descriptions contained the information to provide the correct answers. The graphs below show whether the user was able to extract the correct answer from the text or the chatbot. In Figs. 5 and 6, the number of correct and false answers for scenarios 5 and 8 are shown for both text and chatbot. For each scenario, different values were asked. In both cases, the chatbot performs better for these explanation scenarios (but not perfect, given that the chatbot could have had a better user interface or guidance). In Fig. 7, the scenarios are reported by increasing total percentages of correct answers provided by the user. This means that scenario 3 was the least correctly answered by the users as it has the lowest total percentage of correct answers whilst scenario 1 was the most correctly answered by the users. For *execution scenarios 1 and 3* both chatbot and textual descriptions perform equally well. When looking at the tested *explanation scenarios 5, 9 and 8*, users got more correct answers from the intelligent assistant in scenarios 5 and 8 compared to textual descriptions. For scenario 9, the textual description helped users to get more correct answers.

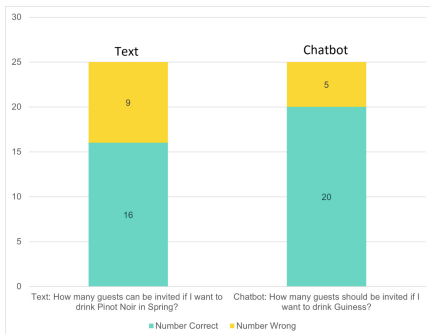


Fig. 5. Answers distribution scenario 5

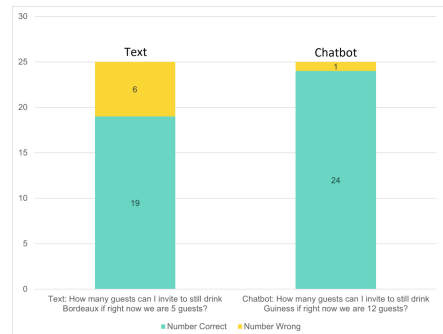


Fig. 6. Answers distribution scenario 8

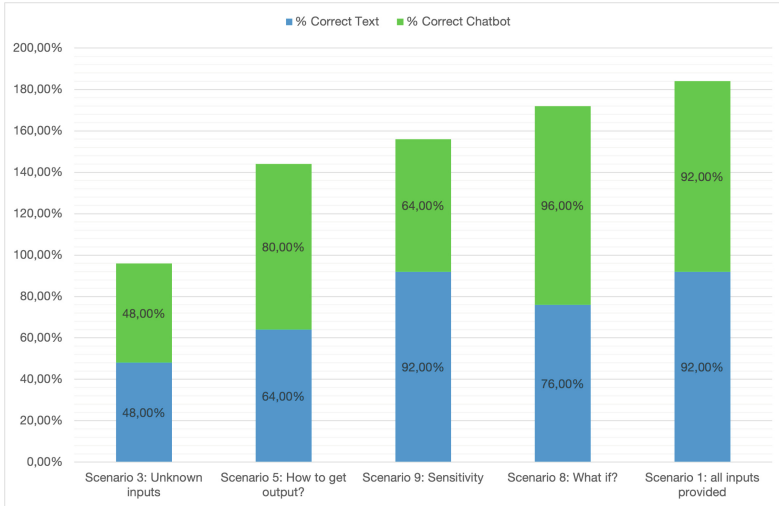


Fig. 7. % of correct answers retrieved from text and intelligent assistant

Answer to Hypothesis 2: The results gathered from the experiment are not conclusive. For the BMI problem, users were faster with the textual description (median of 110 s compared to 120 s). This is of course because the example was very small (just one decision table). But the opposite conclusion was drawn when looking at the beverages problem where using a chatbot was faster on average (median of 222 s compared to 195 s).

Answer to Hypothesis 3: All participants unanimously preferred using the intelligent assistant instead of the textual description.

8 Discussion, Limitations and Future Work

Discussion: From Sect. 7, it is concluded that the intelligent assistant is at least as good for execution scenarios as the textual description. The intelligent assistant also proved itself better compared to textual descriptions for the explanation scenarios, except for scenario 9. The most common feedback for the chatbot prototype was that it was not very user friendly, making it complicated to know how to trigger certain scenarios. This made it difficult for the user to retrieve the correct information as they did not know how to ask that question to the chatbot. It is probable that a better explanation of how to use the chatbot would have allowed participants to better answer scenario 9. We also believe that a more user-friendly chatbot would have impacted the information recovery speed in favor of the digital assistant. The initial findings seem to indicate that digital assistants provide at least equally good explanations compared to textual descriptions if a DMN model is available. Secondly, all users indicated they preferred using the digital assistants. As such, if companies have already adopted

DMN models it seems that they could shift away from updating FAQs, simply upload a DMN model and provide it to the digital assistant. More research is needed to strongly conclude that organizations would benefit from adopting an intelligent DMN assistant.

Limitations and Future Work: Participants of this study were mainly from Belgium, therefore any generalization of the findings to other countries should be done with care. Of course, the quality of the provided explanation depends on the correctness of the DMN model and its decision rules (e.g. overlapping or incomplete rules), but the same applies to the correctness of a webpage, FAQ or a helpdesk. However, in this experiment, the authors have made sure this threat of validity is minimized as much as possible by providing complete and consistent DMN models and textual descriptions containing the exact same information as the DMN models. In the near future, a follow-up study is planned with much more participants where more variables will be investigated such as trust, usefulness, tech savviness, ease of use. This follow-up study will also allow to perform meaningful statistical tests that allow to draw strong conclusions. Further research will also study whether different conclusions could be drawn for different types of participants i.e. expert, knowledgeable and novice with decision modeling.

9 Conclusion

This paper starts with the observation that organizations mainly provide explanations to customers using textual descriptions or expensive help desks. A preliminary experiment was conducted where the participants answered different explanation questions using only a textual description or an intelligent DMN assistant. The main advantage of such an assistant is that it is automatically updated if the rules change in the DMN model which is not the case for FAQs or web pages. Next, it can reason with any DMN model so that organizations can easily adopt it. From the initial findings, we conclude that a generic intelligent assistant is better at providing explanations compared to textual descriptions and that a digital assistant was unanimously preferred over textual descriptions.

References

1. Biard, T., Le Mauff, A., Bigand, M., Bourey, J.-P.: Separation of decision modeling from business process modeling using new “Decision Model and Notation” (DMN) for automating operational decision-making. In: Camarinha-Matos, L.M., Bénaben, F., Picard, W. (eds.) PRO-VE 2015. IAICT, vol. 463, pp. 489–496. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24141-8_45
2. Camunda: DMN Tutorial (2017). <https://camunda.com/dmn/>. Accessed 25 May 2022
3. Dumas, M., et al.: Augmented business process management systems: a research manifesto. arXiv preprint [arXiv:2201.12855](https://arxiv.org/abs/2201.12855) (2022)

4. Estrada-Torres, B., del Río-Ortega, A., Resinas, M.: DemaBot: a tool to automatically generate decision-support chatbots. 2021 Best Dissertation Award, Doctoral Consortium, and Demonstration and Resources Track at BPM, BPM-D 2021, pp. 141–145 (2021)
5. Etikala, V., Goossens, A., Van Veldhoven, Z., Vanthienen, J.: Automatic generation of intelligent chatbots from DMN decision models. In: Moschoyiannis, S., Peñaloza, R., Vanthienen, J., Soyly, A., Roman, D. (eds.) RuleML+RR 2021. LNCS, vol. 12851, pp. 142–157. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91167-6_10
6. Figl, K., Mendling, J., Tokdemir, G., Vanthienen, J.: What we know and what we do not know about DMN. *Enterp. Model. Inf. Syst. Architect. (EMISAJ)* **13**, 2-1 (2018)
7. Galanti, R., Coma-Puig, B., de Leoni, M., Carmona, J., Navarin, N.: Explainable predictive process monitoring. In: 2020 2nd International Conference on Process Mining (ICPM), pp. 1–8. IEEE (2020)
8. Harl, M., Weinzierl, S., Stierle, M., Matzner, M.: Explainable predictive business process monitoring using gated graph neural networks. *J. Decis. Syst.* **29**(sup1), 312–327 (2020)
9. Lins, L.F., Melo, G., Oliveira, T., Alencar, P., Cowan, D.: PACAs: process-aware conversational agents. In: Marrella, A., Weber, B. (eds.) BPM 2021. LNBIP, vol. 436, pp. 312–318. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94343-1_24
10. López, A., Sánchez-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
11. Mehdiyev, N., Fettke, P.: Local post-hoc explanations for predictive process monitoring in manufacturing. arXiv preprint [arXiv:2009.10513](https://arxiv.org/abs/2009.10513) (2020)
12. OMG: OMG: Decision model and notation 1.0 (2015). <https://www.omg.org/spec/DMN/1.0/>. Accessed 08 Jan 2022
13. Rizzi, W., et al.: Explainable predictive process monitoring: a user evaluation. arXiv preprint [arXiv:2202.07760](https://arxiv.org/abs/2202.07760) (2022)
14. Shin, D.: The effects of explainability and causability on perception, trust, and acceptance: implications for explainable AI. *Int. J. Hum. Comput. Stud.* **146**, 102551 (2021)
15. Vandevelde, S., Etikala, V., Vanthienen, J., Vennekens, J.: Leveraging the power of IDP with the flexibility of DMN: a multifunctional API. In: Moschoyiannis, S., Peñaloza, R., Vanthienen, J., Soyly, A., Roman, D. (eds.) RuleML+RR 2021. LNCS, vol. 12851, pp. 250–263. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91167-6_17
16. Vanthienen, J.: Decisions, advice and explanation: an overview and research agenda. In: A Research Agenda for Knowledge Management and Analytics, pp. 149–169. Edward Elgar Publishing (2021)
17. Vanthienen, J., Mues, C., Aerts, A.: An illustration of verification and validation in the modelling phase of KBS development. *Data Knowl. Eng.* **27**(3), 337–352 (1998)



The Label Ambiguity Problem in Process Prediction

Peter Pfeiffer^(✉), Johannes Lahann, and Peter Fettke

German Research Center for Artificial Intelligence (DFKI), Saarland University,
Saarbrücken, Germany

{peter.pfeiffer,johannes.lahann,peter.fettke}@dfki.de

Abstract. Predictive process analytics enables proactive situational awareness by predicting the future of ongoing process instances. To provide a fair comparison between different approaches developed for process prediction, they have been evaluated on publicly available event logs using the next step prediction task. This paper aims to raise awareness of the label ambiguity problem in the context of process prediction by investigating how uncertainty in the ground truth labels affects next step prediction. Label ambiguity arises from cases in the event log that have different continuation options. We argue that the uncertainty created thereby negatively affects evaluation results. To this end, we present a synthetic example that illustrates the problem of label ambiguity in process prediction and quantify the occurrence of ambiguous ground truth labels in common benchmark datasets. Finally, we discuss implications and present ideas that aim to initiate a discussion on how to deal with label ambiguity in process prediction.

Keywords: Process prediction · Machine learning · Next step prediction · Evaluation

1 Introduction

Making predictions on running process instances using machine learning methods has recently gained much attention [3, 6]. The development of new neural network architectures and encoding methods led to significant research progress and set new state-of-the-art results in multiple process prediction tasks. Such predictive models are not only applicable for predicting the future state of process instances. Instead, after being trained and evaluated on the next step prediction task, they are also used for other tasks such as anomaly detection [7], or clustering [2]. Extensive benchmarks have been conducted to determine which prediction method works best for the task of process prediction [11]. In order to quantify and compare the performance, different approaches are typically assessed by the next step prediction task, i.e., comparing the prediction with the information found in the event log. If the prediction and the next occurring event in a case match it is rewarded with a high prediction score such as precision, recall, F1, or accuracy, and if it doesn't, it is penalized.

© Springer Nature Switzerland AG 2023

C. Cabanillas et al. (Eds.): BPM 2022, LNBIP 460, pp. 37–44, 2023.

https://doi.org/10.1007/978-3-031-25383-6_4

However, the occurred behavior in one case might not be the only acceptable answer for the next step since multiple options on how a process instance continues can be found in the event log. On the other side, the fact that multiple continuation options exist in the log does not mean that one particular process instance can continue with all of them. This paper will discuss how this affects the ground truth used to evaluate process prediction models by considering the label ambiguity problem. The term label ambiguity describes the uncertainty about the ground truth labels in predictive tasks [4]; particularly if different ground truth options for one training instance exist. For example, in image segmentation, it might be hard to label examples accurately, which can create uncertainties leading to label ambiguity, as different experts might label one example a little differently. In the scope of process prediction, we refer to label ambiguity as the phenomenon that the ground truth label(s) derived from the event log are uncertain due to typical characteristics of business processes which is different from other types of uncertainty such as model or event data uncertainty. Model uncertainty [10], which can be learned from the data or expressed by the predictive model, describes the uncertainty in its prediction, while event data uncertainty [8] is tied to the impreciseness of the data recorded. In this work, we focus on its impact on the evaluation of predictive approaches from a conceptual perspective.

Little attention has been paid to this problem so far. We argue that it affects the theoretically optimal performance a predictor can achieve in the next step prediction task but also the performance measures obtained during evaluation used for comparison. Systematically underestimating the precision of a predictive model can lead to problems. From a scientific perspective, it prevents an objective estimation of how accurate the developed approaches actually are and how much improvement is possible. From an operational perspective, it might lead to unjustified conclusions when comparing different approaches.

In this paper, we want to raise attention to this problem and motivate a discussion about the use and evaluation of next step prediction from the perspective of label ambiguity.

2 Label Ambiguity in Process Prediction

Process prediction approaches work on data gathered from the execution of business processes stored in event logs. An event log consists of cases that are represented as sequences of events. Each event has attributes like the activity, timestamp, and resource that describe what action in which context has been performed. The task of next step prediction in process prediction is to predict which activity will be performed next in a running process instance. Thus, the sequence of executed events is used to make a forecast about which activity will come next. Training and assessment of process prediction approaches are performed on event logs containing a large number of cases. Using the cases in the event log, training examples, i.e., pairs (x, y) of inputs x and targets y are created. Each input x is a sequence of events, while y is the expected activity to

Table 1. Prefix $\langle A, B, C \rangle$ with 3 different options found in a event log how it can continue.

Prefix	$\langle A, B, C \rangle$		
Sequences	$\langle A, B, C, D, E \rangle$	$\langle A, B, C, L, M, K \rangle$	$\langle A, B, C, M, L, K \rangle$

be predicted. Thus, y serves as the ground truth for the next step in x . During training and assessment, x is given to the predictor f , the prediction $\hat{y} := f(x)$ is compared with y and based on that comparison, performance measures like precision, recall, and accuracy are calculated. To demonstrate the ambiguity problem, we will first focus on the situation where the sequences of events consist of activities only.

Consider three sequences $\sigma_1 = \langle A, B, C, D, E \rangle$, $\sigma_2 = \langle A, B, C, L, M, K \rangle$ and $\sigma_3 = \langle A, B, C, M, L, K \rangle$ shown in Table 1. All start with the same sub-sequence, also denoted as prefix or head [1, p. 134], $\sigma_{pref} = hd^3(\sigma_1) = hd^3(\sigma_2) = hd^3(\sigma_3) = \langle A, B, C \rangle$ but continue with a different activity. For σ_1 it is D , for σ_2 it is L while for σ_3 it is M . Since there are multiple valid next steps $\{D, L, M\}$ for σ_{pref} rather than a single option, one can argue that next step prediction should be formalized as a multi-label classification problem that predicts all continuation options for a running process instance. However, in the literature, next step prediction is typically understood as a single-label prediction problem, where exactly one activity is predicted. In accordance with this, the training samples generated contain only one subsequent activity that is to be predicted.

Thus, for the sequences σ_1 , σ_2 and σ_3 having the same prefix σ_{pref} , three training examples (x, y) with the same input $x = \sigma_{pref}$ would be created. One with target $y = D$, one with target $y = L$ and one with target $y = M$. Three training examples with the same input x but different y create uncertainty about the ground truth labels. We refer to such training examples having the same sequence of activities but different next steps as *ambiguity candidates*. Regardless of which option f predicts, the assessment evaluates to correct for one sample and to false for the others. Thus, having ambiguous ground truth values negatively affects performance measures.

One can argue that this effect only occurs in the synthetic sample but may not occur in real-live event logs. In particular, real-live logs might contain dependencies in contextual information in the events that resolve the label ambiguity. For instance, if the requested amount of money or the resource that executed the previous activity would make the next step clear. Hence, we checked for commonly used next step prediction datasets (Helpdesk¹, BPIC 2012², BPIC 2017³, BPIC 2013⁴) how many samples are affected by label ambiguity. Note that we now consider sequences of events, i.e., $\sigma = \langle e_1, \dots, e_n \rangle$ where each event consists

¹ <https://doi.org/10.17632/39bp3vv62t.1>.

² <https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b>.

³ <https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f>.

⁴ <https://doi.org/10.4121/uuid:a7ce5c55-03a7-4583-b855-98b86e1a2b07>.

Table 2. Characteristics of commonly used event logs for process prediction.

Dataset	Helpdesk	BPIC 2012	BPIC 2017	BPIC 2013
#Cases	3,804	13,087	31,512	7,554
#Training examples	9,007	249,113	1,170,758	8,945
#Ambiguity candidates	5,006	103,326	497,099	1,296
Percentage	55.58%	40.67%	42.46%	14.49%

of different attributes rather than the activity only. We constructed training examples (x, y) following a common procedure [7, 9] by first building prefixes $pref(\sigma) = \{hd^k(\sigma) \mid 2 \leq k \leq n\}$ [1, p. 134] for all cases in the event logs and splitting the prefixes into x and y . For each $\sigma_{pref} \in pref(\sigma)$ of length n we defined $x := hd^{n-1}(\sigma_{pref})$ and $y := \#activity(e_n)$. Given the training examples, we tried to answer the following two questions:

1. How many samples (x, y) are *ambiguity candidates*, i.e., share the same sequence of activities $\{\#activity(e) \mid e \in x\}$ but have different y ?
2. Are there indicators in the context attributes of $e \in x$ that resolve the uncertainty in y for the *ambiguity candidates*?

Table 2 lists the number of cases in the event log, training examples generated from the cases, and *ambiguity candidates* together with the percentage of how much they make up of all training examples. For all analyzed event logs, the number of *ambiguity candidates* makes up significant shares of the total training examples.

Next, we checked if context attributes help to resolve the ambiguity. Due to space constraints, we only demonstrate that for the BPIC 2012 dataset. The prefix with activity sequence $\langle A_SUBMITTED, A_PARTLYSUBMITTED \rangle$ from the BPIC 2012 dataset shown in Fig. 1 accounts with 5.25% for the largest share of *ambiguity candidates* and has 4 options in y how to continue. In all those training examples, regardless of the next activity y , the attribute *resource* always has the value *112* in x . In contrast, the mean and standard deviation of attribute *AMOUNT_REQ* in x for all options do not differ significantly. Similarly, for the variant with the second most training samples in the log, there is also no pattern in the context attributes that clearly indicate which activity to choose next.

Other variants are very likely also affected by this problem. Similar observations have been made for the Helpdesk dataset, where more than half of the cases have multiple continuation options, and only the timestamp is included additionally. For the other two datasets, we were not able to verify if context resolves the ambiguity for the *ambiguity candidates* due to the high number of context attributes which increases the complexity of possible patterns and the difficulty of proving that there are no dependencies which resolve the ambiguity.

We conclude that commonly used datasets have a high number of samples that are possibly affected by label ambiguity. Additionally, the context information was not sufficient to resolve the uncertainty for the two most frequent

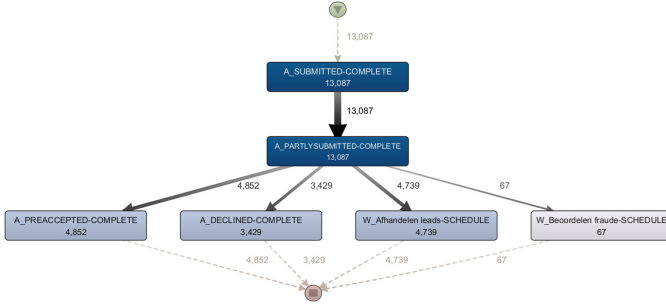


Fig. 1. The prefix $\langle A_SUBMITTED, A_PARTLYSUBMITTED \rangle$ has 4 options how to continue (except of the BPIC 2012 event log).

variants affected in the BPIC 2012 event log. Thus, the performance measures reached on those datasets should be interpreted having label ambiguity in mind.

3 Dealing with Label Ambiguity

There are different options on how to deal with label ambiguity depending on the actual problem to solve. Sometimes, next step prediction is used as a proxy task for training a neural network on event log data before using it to solve the actual target task, e.g., anomaly detection [7]. In such situations, the idea is similar to next word prediction which is used to train vector space representations of sequences of words to be used for language modeling applications. Applied to event log data, this translates to learning a representation of sequences of activities which are helpful in solving target task. In these use cases, ambiguity is less of a problem as the performance of such approaches is evaluated on the target task (sometimes the performance in next step/word prediction is also specified). However, different to next word prediction, predicting the next step in a running process instance is actually a business problem used to give operational support. Thus, we will focus on situations where next step prediction is the actual problem, i.e., the target task to solve.

Similar to the label ambiguity problem in image classification [4], instead of predicting one label, a probability distribution over labels could be predicted. If label predictions are required, single or, multiple labels can be drawn from the distribution using a threshold. Such approaches exist, and, in principle, all neural-network-based approaches using a soft-max function in their last layer actually predict such a probability distribution. In most cases, the label with the highest probability is chosen as \hat{y} . However, multiple labels can be drawn from the distribution generated from the soft-max function.

Instead of predicting probabilities, the assessment method can be adapted to return probabilistic results [5]. Instead of using a binary assessment ($y == \hat{y}$), the assessment takes the uncertainty of y into account for calculating precision, recall, and F1.

Another option is to formalize next step prediction as a multi-label classification problem instead of a single-label. For this, all next steps that possibly can happen have to be predicted.

One major obstacle for all three options is that correct ground truth labels are required to assess how accurate the predictions are. In the case of predicting distributions, ground truth label distributions must be created for each prefix x , indicating the probabilities of y . While such a distribution is straightforward to compute when considering the sequence of activities only, it becomes more complicated the more attributes one considers, as there might be complex dependencies in the attributes that affect the probabilities for the resolution of uncertainty. For example, imagine a situation where the requested amount of money influences which activity comes next. Depending on the amount, different distributions need to be constructed. It is not sufficient to take all the next options as correct ground truth values as there might be only one. If more dependencies between attributes exist, distributions become even more complicated to compute.

Apart from considering the data, the situation in Fig. 1 shows what can go wrong when constructing such distributions without considering the semantics of the process. After $\langle A_SUBMITTED, A_PARTLYSUBMITTED \rangle$ activities $A_PREACCEPTED_COMPELTE$ and $A_DECLINED_COMPLETE$ can follow (among others). The semantics of those activities is the opposite. Constructing a ground truth probability distribution without considering this fact would mean that both of them are valid continuations with a certain percentage. However, this is obviously wrong as only one can be the correct one, and predicting the other or both would be completely wrong.

We conclude that constructing ground truth label distributions from the event log is complicated and error-prone for the mentioned reasons.

4 Conclusion

In this paper, we have discussed the label ambiguity problem in process prediction and found that a large number of cases in real-life event logs are affected. As label ambiguity seems to be ubiquitous in process prediction, it is a phenomenon we have to deal with.

While we found that many training examples are possibly affected by the label ambiguity problem, a more extensive analysis is required to what extent this actually influences the performance measures obtained during evaluation. It would also be interesting to find out the exact number of ambiguous cases when considering all context information to calculate an upper bound on the maximal performance one can archive by disregarding the ambiguous cases. This would help to shed light on the absolute performance of current process prediction approaches. Furthermore, other datasets should be checked if and to what extent the label ambiguity problem exists there as well.

We hope to raise awareness of this problem and conclude that evaluation of process prediction approaches remains challenging. It may require domain

knowledge to obtain correct ground truth values to assess the predictions. We encourage a discussion about the impact of this phenomenon on how we perform and evaluate process prediction in the future. In our opinion, label ambiguity is a challenge for next step prediction, which should be considered. We believe that process prediction could be improved with a different conception of the task, appropriate evaluation methods, and more elaborated labeling techniques. The ideas proposed in this paper might give a starting point. However, we recognize the necessity of further research to address label ambiguity and to obtain unaffected performance measures.

Acknowledgement. Part of this work has been done funded by the project *Smart Vigilance* (FKZ: 01IS20028C) with financial support by the Federal Ministry of Education and Research (BMBF).

References

1. van der Aalst, W.: *Process Mining: Data Science in Action*, 2nd edn. Springer, Heidelberg (2016)
2. De Koninck, P., vanden Broucke, S., De Weerd, J.: act2vec, trace2vec, log2vec, and model2vec: representation learning for business processes. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *BPM 2018*. LNCS, vol. 11080, pp. 305–321. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_18
3. Evermann, J., Rehse, J.-R., Fettke, P.: A deep learning approach for predicting process behaviour at runtime. In: Dumas, M., Fantinato, M. (eds.) *BPM 2016*. LNBP, vol. 281, pp. 327–338. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58457-7_24
4. Gao, B.B., Xing, C., Xie, C.W., Wu, J., Geng, X.: Deep label distribution learning with label ambiguity. *IEEE Trans. Image Process.* **26**(6), 2825–2838 (2017)
5. Kuss, E., Leopold, H., van der Aa, H., Stuckenschmidt, H., Reijers, H.A.: A probabilistic evaluation procedure for process model matching techniques. *Data Knowl. Eng.* **117**, 393–406 (2018)
6. 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 (2021). <https://doi.org/10.1007/s10462-021-09960-8>
7. Nolle, T., Seeliger, A., Mühlhäuser, M.: BINet: multivariate business process anomaly detection using deep learning. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *BPM 2018*. LNCS, vol. 11080, pp. 271–287. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_16
8. Pegoraro, M.: Probabilistic and non-deterministic event data in process mining: embedding uncertainty in process analysis techniques. In: *Proceedings of the Doctoral Consortium Papers Presented at the 34th International Conference on Advanced Information Systems Engineering (CAiSE 2022)* (2022)
9. Pfeiffer, P., Lahann, J., Fettke, P.: Multivariate business process representation learning utilizing Gramian angular fields and convolutional neural networks. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) *BPM 2021*. LNCS, vol. 12875, pp. 327–344. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_21

10. Portolani, P., Brusaferrri, A., Ballarino, A., Matteucci, M.: Uncertainty in predictive process monitoring. In: Ciucci, D., et al. (eds.) IPMU 2022. CCIS, vol. 1602, pp. 547–559. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08974-9_44
11. Rama-Maneiro, E., Vidal, J., Lama, M.: Deep learning for predictive business process monitoring: Review and benchmark. *IEEE Trans. Serv. Comput.* (2021)



Situation-Aware eXplainability for Business Processes Enabled by Complex Events

Guy Amit, Fabiana Fournier, Lior Limonad^(✉), and Inna Skarbovsky

IBM Research – Haifa, Haifa, Israel
guy.amit@ibm.com, {fabiana,liorli,inna}@il.ibm.com

Abstract. Traditionally, the organizational IT landscape is split between Business Process engines that are developed to handle process execution workloads and Complex Event Processing engines designed to search for event correlations in real time to derive actionable insights. For the benefit of process trustworthiness, this work focuses on combining the two engines, resulting in an enriched form of a process log that serves as an input to recently developed eXplainable Artificial Intelligence frameworks, yielding more adequate explanations for process execution outcomes. A designated methodology and a test scheme were created to systematically implement and evaluate our overall approach and its effectiveness in gaining situation-aware explainability. Specifically, we demonstrate our approach using a dataset populated for an illustrative process example, replaying its trace log against the PROTON CEP engine and feeding the result as an input for the SHAP explainer.

Keywords: eXplainable Artificial Intelligence · Complex Event Processing · Business Processes Management · Situation-Aware eXplainability

1 Introduction and Motivation

Augmented Business Process Management Systems (ABPMSs) [5] are a new generation of business process management systems. The goal of ABPMSs is to empower the execution of business processes with novel AI-based capabilities. One of the main characteristics of ABPMSs is their enhanced trustworthiness, shaped by an ability to explain and reason about processes executions. Finding an adequate explanation is not easy, because it requires understanding the situational conditions in which specific decisions were made during process enactments. Frequently, explanations cannot be derived from “local” inference (i.e., current undergoing task or decision in a business process) but require reasoning about situation-wide contextual conditions relevant to the current step as derived from some actions in the past. The recent manifesto [5] calls for “Situation-Aware eXplainability (SAX)” as one of the most prominent research challenges. SAX entails ongoing tracking of reasoning assumptions and inferential associations between subsequent enactments, as a basis for providing trustful explanations.

Complex Event Processing (CEP) enables situation awareness by applying temporal and contextual reasoning on incoming events to produce higher-level of insights (‘complex events’ or ‘situations’). In this paper, we combine situations derived by a CEP engine with traces of a business process executions and show that the resulting “enriched” event log can produce better situation-aware explanations. Our contribution in this work is the augmentation of the conventional use of BPM and XAI with CEP to achieve more adequate explanations of process execution outcomes. Respectively, our solution enables to hypothesize about any plausible causal situation to be examined for its possible effect on process execution outcomes, both in real-time and in retrospect. Our method is generic and can be applied with any CEP and XAI tools. We henceforth elaborate on the underlying fundamental concepts.

2 Background

A business process (BP) is a collection of tasks that execute in a specific sequence to achieve some business goal [18]. The digital footprint that depicts a single execution of a process as a concrete sequence of activities or events is termed a ‘trace’ [1]. A multi-set of traces is usually referred to as a trace-log or event-log. We hereforth describe some basic concepts related to CEP, XAI, and replaying. ***Event Stream Processing (ESP)*** or CEP is computing that is performed on streaming data (sequence of events) for the purpose of stream analytics or stream data integration. ESP is typically applied to data as it arrives (data “in motion”). It enables situation awareness and near-real-time responses to threats and opportunities as they emerge, or it stores data streams for use in subsequent applications [4]. The results of ESP computation are complex events. A complex event may be derived from just a few or from millions of base (input) events from one or more event streams. Stream analytic applications provide continuous intelligence to enhance situation awareness, enable sense-and-respond behavior or just inform real-time decisions. Organizations are doing more stream processing because of the need for continuous intelligence and better situation awareness, as well as faster, more precise business decisions [15].

In our work, we use the PROactive Technology ONline¹ (PROTON) tool as our CEP engine. PROTON follows the terminology and semantics presented in [6]. Its programming model is based on the notion of an Event Processing Network (EPN). An EPN comprises a collection of event processing agents (EPAs), event producers, events, and event consumers. The network describes the flow of events originating at event producers and flowing through various event processing agents to eventually reach event consumers. An EPA is a software module that processes input events and looks for matches between these events, using an event processing pattern or some other kind of matching criterion. An event pattern is a template specifying one or more combinations of events. Given any collection of events, if it’s possible to find one or more subsets of those events that match a particular pattern, it is said as satisfying the pattern. We denote

¹ PROTON open source (Apache v2 licence): <https://github.com/ishkin/Proton>.

situations as the complex events emitted by a CEP engine. A PROTON CEP application consists of a JSON file that defines the EPN that is matched against a streaming of events in real-time to derive the defined situations.

eXplainable AI - Recent advancements in Machine-Learning (ML) [2, 11, 17] have been achieved with increase in the complexity of models that require external explanation frameworks, namely XAI. Such frameworks are predominately developed for post-hoc interpretations of ML models [2, 11]. Context-wise, they can be divided into global, local, and hybrid explanations [2, 8, 13]. Global explanations attempt to explain the ML model’s internal logic, local explanations try to explain the ML model’s prediction for a single input instance, and hybrid approaches vary (e.g., explaining the ML model’s internal logic for a subspace of the input space). This paper adds to a series of recent efforts [3, 16] that focus on exploiting XAI frameworks that are compatible with tabular data for the interpretation of BP execution results. We use process logs as the main data input and train surrogate ML models with this data to represent real-world business processes. As such, ML model faithfulness to the real BP may be lacking, capturing only parts of the holistic situations in which decisions were made. We show how with the use of CEP, the data input for the explainer could be augmented with situation relevant enrichments that result with more adequate explanations. The most commonly used ML-model-agnostic post-hoc local XAI frameworks, compatible with tabular data, are LIME [14] and SHAP [10]. Both rely on sampling data points by way of feature perturbations for derivation of feature importance around the examined sample. To assess the effectiveness of our method, we used SHAP, mostly due to its ability to accommodate inter-feature dependencies. The approach for process explainability is based on the training of a decision-tree to associate process execution variables with process outcomes. Such training uses historical process execution logs that are enriched by the CEP. An XAI tool is then employed to explain individual process execution cases by ranking the importance of the process variables with respect to specific outcomes of interest.

Replaying Process Logs - The terms ‘play in’, ‘play out’ and ‘replay’ were originally used in the work of Harel [9], and later adopted by van der Aalst [1]. ‘Play out’ typically relates to conventional use of a BPM engine, and ‘play in’ refers to the core notion of process mining. Most relevant to our work, the notion of process ‘replay’ combines the process event-log and the model as input for purposes such as checking for execution conformance, predictions, and diagnostics. In the context of process explainability, replay describes best the operation in which historical traces are replayed for the elicitation of richer, situation relevant events that may have previously promoted to internal process decisions but have not been originally persisted by the BPM engine. The replay module enables simulating the process execution as input to the CEP engine as if events were occurring in real-time. More concisely, the events in the process log are streamlined according to the original BP model sequencing as an input to the CEP engine, having the latter derive situations that enrich the original event log as an input to the explainer. This could be employed either in real-time or in a simulated mode, for better explainability derived in retrospect.

3 Methodology

3.1 Types of Events

We identify three types of situations as a function of the source of the events:

- **Events source is internal to the BP.** The situations are a combination of process execution events along with the CEP defined patterns. These situations are tightly coupled with the BP execution.
- **Events source is external to the BP.** These situations are not coupled with the BP execution. We note that although the event is external to the BP it might influence its execution and the outcomes.
- **Events source is a combination of internal and external circumstances.** These situations are bi-coupled to external as well as internal to the BP and are typically characterized by transient/ad-hoc events. The BP model does not change but the (temporal) alteration of its flow affects the possible explanation given to a specific execution instance.

3.2 Approach

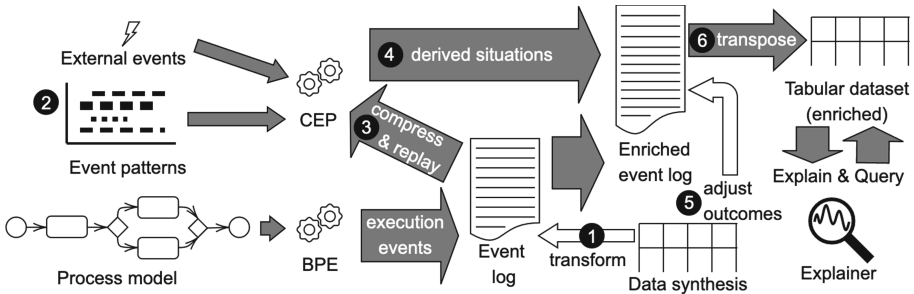


Fig. 1. Sequence of our approach: (1) transform, (2) implement, (3) compress & replay, (4) run CEP, (5) adjust, and (6) transform.

Our methodology includes the following steps (Fig. 1):

1. **Transformation of the tabular dataset into an event log** - The tabular dataset contains rows indicating different states in the BP for each case, with task execution and completion times denoted respectively. Given these times, each row is transformed into a series of timestamped events, where each event is assigned the original case-id and time corresponding to the original row.
2. **Implementation of the CEP application** - This stage unfolds the definition of the event patterns and situations to be detected by the CEP engine. In the case of PROTON, this CEP application specification is implemented as a JSON file consisting of the EPN for the specific application.
3. **Compression of time windows for expedited replay of the event log** - CEP applications are meant to run and trigger situations in real-time. However, when testing a CEP application, we cannot “wait” for events to

happen at their original occurrence times and therefore aim at replaying the events in shorter time windows. For this, we apply PROTON’s simulation tool called Proton EventT Injection and Time comprESSION (see footnote 1) (PETITE) that compresses the original times into shorter intervals. The result is a compressed event log that serves as input file for the CEP application and a new JSON file with “compressed” times.

4. **Running of the CEP application** - Replay of the event log resulted from step 3 as input to the CEP engine. The outcome of this execution is an “enriched” event log, containing the original events interwoven with the situations detected by the CEP engine (timed events).
5. **Adjustment for situation effects** - According to each situation, decision variables (e.g., acceptance) are modified and post-situation events are trimmed from the enriched file.
6. **Transformation of the enriched log into an enriched tabular input for the explainer** - The enriched log is transformed back to the original table format, where additional columns are added that represent features of the new events and situations discovered.

Steps 1–6 apply when replaying the log in retrospect. In the case of running in real-time, only steps 2, 4, and 6 are required. Steps 1 and 5 are strictly associated with the case of data synthesis in which data is generated for testing purposes.

4 Illustrative Example

Figure 2 depicts a loan application model using BPMN [7] notation. Each loan application goes through a set of predefined set of activities (e.g., verify amount and credit check) and decisions junctions (e.g., amount \geq 1000) resulting in either acceptance or rejection of the loan application. For simplicity, we use a process example ending with a binary decision. However, our approach is applicable for any multi-class decision outcome explainability, occurring at any point during process execution. Instantiation of the process can be encoded in a tabular form as depicted in Table 1. Each process instance includes a column for its initiation time, for each decision variable (e.g., amount), and for each task, encoded in a Boolean column to depict whether it was executed or not and a second column denoting its completion time in minutes.

Table 1. An example of an instance in the tabular dataset

case ID	credit_score	risk	done_receive_loan	done_verify_ammount	done_credit_check	done_risk_assessment	done_skilled_agent	done_novice_agent	done.accept
TVQR	1173.08	397.78	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE
...									
arrival_time	post_received_time	post_verify_amount_time	post_credit_check_time	post_risk_assessment_time	post_skilled_agent_review_time	post_novice_agent_review_time	post_decision_time		
48	54	59	71	NaN	87	NaN	92		

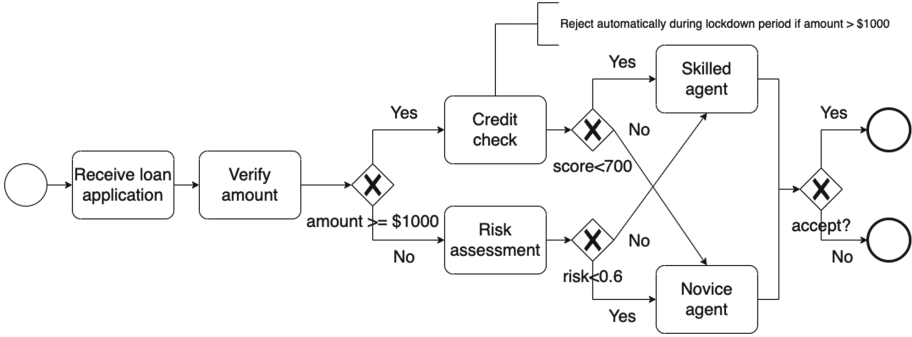


Fig. 2. Loan application BP model in BPMN, with the potential deviation during lockdown informally denoted by an annotation.

4.1 Applying the Methodology in Our Example

Henceforth, we exemplify the above methodology in our illustrative example:

1. **Transformation of the tabular dataset into an event log.** For example, for a case ID=TVQR (Table 1) with `done_credit_check = TRUE` and `post_credit_check = 71` (mins), an event `CreditCheck` with “case ID = TVQR, ... , OccurrenceTime = timestamp of streaming initiation + 71 mins” was added to the event log. Similarly, events with the same case ID were added for: `Arrival`, `Received`, `VerifyAmount`, `SkilledAgentReview`, and `Decision`.
2. **Implementation of the CEP application** - Let’s assume we would like to derive the following complex events or situations:
 - Derive a situation *AgentOverflow* that causes the rejection of new loan applications when more than 4 applications pile-up on an agent’s table. The rationale is that when there is a large workload, the tendency is to reject to speed up the process while “being on the safe side”.
 - Derive a situation *DecisionTimeMoreThanTwoDays* that determines an application as rejected if it stays in the system more than 2 days. Again, the rationale is that want to eliminate bottlenecks while not taking the risk of accepting a somewhat risky application.
 - Derive a situation *LockDownNewGuideline* that concludes an application as rejected if it was submitted during Covid-19 lock-down period with an amount greater than \$1000.

We also have two input events that are part of the *LockDownNewGuideline* situation: *LockDownInitiator* and *LockDownTerminator* that are employed by the CEP engine to denote the beginning and end of a lockdown period. The *AgentOverflow* and *DecisionTimeMoreThanTwoDays* situations are examples of situations in which the event source is the BP in hand as they are derived only as a result of BP instance’s state. On the other hand, the *LockDownNewGuideline* situation is an example of a situation in which the event sources are a combination of internal (e.g., the amount of the requested loan)

and external to the BP (e.g., the lockdown). An EPN with the event pattern definitions in a JSON file specifying these three situations was created.

3. **Compression of time windows for expedited replay of the event log.** The event log produced in step 1 spans four days, the same as the time horizon of the original dataset. By applying the PETITE simulator, we compressed the 4 days elapsed time into 5 min.
4. **Running of the CEP application** - PROTON was applied using the JSON file and the event log resulting from step 3. The output is an enriched log that includes the input events as well as the situations derived by the CEP engine.
5. **Adjustment for situation effects** - We tweaked the log from ‘acceptance’ to ‘rejection’ where the situation was affecting the process result. Trimming was not required here, since all situations related to final process outcomes.
6. **Transformation of the enriched log to tabular input for the explainer** - The additional columns contain new features with time in min from first instance initiation. For example, in the case of the detection of the DecisionTimeMoreThanTwoDays situation, a non-null value representing the occurrence time is added to the TwoDaysWithoutDecision column. Concretely, the value of these new features is either NaN to denote the non-occurrence of the situation, or a value depicting the time from process initiation. In our case, we further decoded the enrichment into a Boolean, translating a NaN value to ‘false’ and a time value to ‘True’.

5 Evaluation

Conforming to the tabular form in Table 1, we populated a dataset with 1000 instances with decision variables drawn at random from Gaussian distributions and completion times drawn from Poisson distributions. Task execution columns were computed based on the values of the decision variables as entailed from process flow logic. As an input for training of the explainer ML model (i.e., SKLearn’s [12] decision tree classifier), we split the data at random into training (800 records) and test (200 records) sets. Our set of test cases for each of the three situations included the following variations, where for each situation 4–5 instances were instantiated:

- T1 Loan acceptance toggled into rejection - a set of test instances in which prior to the enrichment, the original process execution decision was loan ‘acceptance’, and where the decision was altered into loan ‘rejection’ as a result of the newly incurred situation. For any of these test instances, our expectation was to have the explainer identify the corresponding enriched situation manifest itself as a top importance feature in the explanation.
- T2 Rejection overridden by a ‘new’ rejection reason - a set of test instances in which the original process execution decision was loan ‘rejection’, and for which the decision remained a ‘rejection’ as a result of the newly incurred situation, and where the new situation becomes the prominent reason for the rejection instead of the original one. For any of these test instances, our expectation was to have the explainer identify the corresponding situation manifest itself as a top importance feature in the explanation.

- T3 Rejection that persists itself - specific to the *LockDown* situation, some of its corresponding test instances reflected process execution scenarios in which the original decision to reject the loan was not affected by the occurrence of the *Lockdown* situation. For these particular instances, although the result was a ‘rejection’, we did not expect the explainer to identify the corresponding situation manifest itself as a top importance feature in the explanation.
- T4 Contrasting set - with respect to each type of enriched situation and our overall dataset, we identified the subset of instances in which the situation did not apply. Our examination here was to ensure that none of these instances happens to be incorrectly explained by an enriched situation.

The loan rejection toggled into acceptance and loan acceptance that remains acceptance variations were dropped since they are symmetric to the above.

We present here the global model explanation of the enrichment, followed by detailed examination of the local model explanations with respect to the test cases, applied to corresponding situation instances. Our code is available at: <https://github.com/IBM/SAX/tree/main/AI4BPM>.

6 Results

Global distribution of feature importance is illustrated in Table 2. The newly added features play a role in about 5% of the explanations, reflective of the proportion of instances that were tweaked to test for the effects of the newly introduced situations. Furthermore, model accuracy in predicting eventual process outcomes has also been fully mitigated by the enrichment.

Table 2. Global explainability with and without situational enrichment

Without enrichment	With enrichment
Model accuracy (DT): 0.97	Model accuracy (DT): 1.00
0. amount : 0.4826136779898476	0. risk : 0.7714530282992648
1. risk : 0.3426102769869956	1. credit_score : 0.15177506322470122
2. credit_score : 0.17477604502315672	2. AgentOverflow : 0.04241708399719542
	3. LockDownTerminated : 0.019477216063815347
	4. TwoDaysWithoutDecision : 0.014877608415023355

For each of the enriched situations, we elaborate in a corresponding table example cases that highlight test scenario results. Each table includes local feature importance in a descending order for test cases T1–T3 without and with the enrichment, and a collective force-plot with all non incurred situation instances that are graphically stacked horizontally corresponding to test case T4.

AgentOverflow situation: A sample of local test results for this situation is shown in Table 3. With respect to [T1], case id ‘FSAN’ was manually tweaked from originally being accepted into being rejected due to overflow. As listed, prior to the

enrichment, the original explanation incorrectly included ‘risk’ as the top importance feature in the explanation for the rejection, while after the enrichment, the most important feature was correctly recognized as ‘agent-overflow’.

With respect to [T2], case id ‘BROG’, which originally concluded with a loan rejection decision, was modified in the dataset to still be rejected, but in its modified form was tweaked to ensure the reason for the rejection was an overflow situation. As listed, the original incorrect explanation with ‘amount’ as the top feature has been properly replaced by ‘agent-overflow’ as the top importance feature in the enriched explanation.

[T3] doesn’t apply in the case of an agent overflow situation. This is since whenever an overflow event occurs, it is inevitable for the decision to entail an immediate rejection, overriding whatever was the original reason for the rejection, as already covered by [T2]. An exception for a situation that is relevant for test case [T3] is elaborated in the context of the Lockdown Guidelines situation.

With respect to [T4], we examined all contrasting instances associated with the non occurrence of an overflow situation to ensure none happens to incorrectly include in its explanation the ‘agent-overflow’ feature. Horizontal stacking with all relevant instances was plotted as illustrated. As desired, no instance in this set was identified to include the ‘agent-overflow’ feature in its explanation.

Table 3. AgentOverflow situation - test results

Test case	Case ID	Non enriched	Enriched
T1: ‘accept’ to ‘reject’	FSAN	risk-, credit-score-, amount+	✓ agent-overflow-, risk+, credit-score-
T2: overridden ‘reject’	BROG	amount-, credit-score-, risk+	✓ agent-overflow-, credit-score-, risk-
T3: persisted ‘reject’	na		
T4: contrasting set			
		✓ An example plot we created to verify no instance in the contrasting set includes ‘agent-overflow’ in its explanation.	

Decision Time More Than Two Days Situation: As with the previous agent-overflow situation, local test results for the situation of decision time greater than two days are shown in Table 4. Case id ‘YMOJ’ was tweaked to match the scenario of [T1]. Correspondingly, the top importance feature correctly changed from ‘risk’ to ‘two-days-without-decision’ as a result of the enrichment. Adjustment of case id ‘MJKQ’ for test [T2] resulted with proper alteration from ‘credit-score’ to ‘two-days-without-decision’ as desired. [T3] was skipped for the same reason as above. [T4] presented no undesired side effects in the explanation of any of the contrasting set instances for the situation inspected.

Table 4. Decision time > 2 days situation - test results

Test case	Case ID	Non enriched	Enriched
T1: ‘accept’ to ‘reject’	YMOJ	risk-, credit-score-, amount+	✓ two-days-without-decision- , risk+, credit-score-, agent-overflow+
T2: overridden ‘reject’	MJKQ	credit-score-, amount-, risk+	✓ two-days-without-decision- , is-credit-, credit-score-, risk+, agent-overflow+
T3: persisted ‘reject’	na		
T4: contrasting set	✓ No instance resulted with ‘two-days-without-decision’ in its explanation		

Lockdown Guideline Situation: We repeated the same tests for the Lockdown guideline situation as shown in Table 5. As with the two other situations, [T1] demonstrated a remedy in the explanation due to the enrichment for case id ‘YMOJ’. With respect to [T2], the complexity of the situation lets us examine two event encoding nuances: implicit and explicit encoding of the guideline as an interaction between the external occurrence of a lockdown and loan amount being greater than \$1000. For the former implicit case, only ‘LockdownInitiated’ and ‘LockdownTerminated’ events were included in the enrichment examination. For technical simplicity, both events were transformed in the dataset from their original timestamped form into a single Boolean variable that was either ‘true’ in any instance that occurred between the two timestamps, or otherwise was ‘false’ (named ‘lockdown-terminated’). Our aim was to test the ability of the explainer to reveal interactions among features. In our case, between the process external occurrence of a lockdown and the process internal amount being greater than \$1000. First run of [T2] failed to recognize ‘amount’ and ‘lockdown-terminated’ as mutual top-importance features in the explanation. We realized that the explainer might be incapable of handling an interaction between a numeric variable (i.e., ‘amount’) and a boolean variable (‘lockdown-terminated’), and particularly discovering the conditional split of amount around \$1000. To test this, we added another Boolean variable ‘amount>1K’ to explicitly denote this split and re run [T2]. As illustrated in Table 5, with the additional variable, the SHAP explainer was able to correctly recognize the interaction as the top two features.

For the latter explicit case, a ‘LockdownAndLargeAmount’ event was derived by the CEP engine and was explicitly added as a Boolean variable in the dataset to mark corresponding instances. As with the previous two situations, SHAP was able to recognize such an explicit encoding as a top importance feature.

The uniqueness of the Lockdown guideline also allowed us to instantiate the scenario of [T3] in which the external occurring of a lockdown may not force a rejection i.e., when the amount is smaller than \$1000, reflecting a situation where the original reason for the loan rejection should persist. As demonstrated by case id ‘LJLP’ in the results, in such a case, the ‘lockdown-terminated’ feature was recognized as part of the explanation, but not as a top importance feature, and also ‘amount>1K’ was detected as affecting the model towards loan acceptance.

Table 5. Lockdown guideline situation - test results

Test case	Case ID	Non enriched	Enriched
T1: 'accept' to 'reject'	YVDN	amount-, risk+, credit-score-	✓ lockdown-and-large-amount- , risk+, credit-score-, agent-overflow+
T2: overridden 'reject'	NDZY	credit-score-, amount-, risk+	✓ amount>1K- , lockdown-terminated- , credit-score-, risk+, agent-overflow+
T3: persisted 'reject'	LJLP	risk-, amount+, credit-score-	risk-, ✓ amount>1K+ , lockdown-terminated- , credit-score-, agent-overflow+
T4: contrasting set	✓ No instance resulted with 'lockdown-and-large-amount' in its explanation		

Collective Summary of Results: A handful of cases was adjusted to repeat the tests in each situation. An overall summary of all test results is listed in Table 6. Our results conclusively show that in all three situations, the enrichment of the dataset promoted to a perfect feature correctness in the explanations, without any loss in the correctness of all other instances that were not affected by the newly incurred situations. For the first two situations, the enrichment also promoted the ML model accuracy in predicting process decisions.

Table 6. Summary of test cases: # - num of test cases; T-EXP - % of feature correctness in explanation; ACC - % of correct process result predictions; F-EXP - % of affected cases with false explanation. Noteworthy improvements marked in bold.

Test:	T1: 'accept' to 'reject'					T2: 'reject' overridden					T3: 'reject' persisted					T4: affected
	Not enriched		Enriched			Not enriched		Enriched			Not enriched		Enriched			Enriched
Situation	#	T-EXP	ACC	T-EXP	ACC	#	T-EXP	ACC	T-EXP	ACC	#	T-EXP	ACC	T-EXP	ACC	F-EXP
Agent- overflow	5	0%	60%	100%	100%	5	0%	100%	100%	100%	na	na	na	na	na	0%
Decision > 2 days	4	0%	50%	100%	100%	4	0%	100%	100%	100%	na	na	na	na	na	0%
Lockdown guideline	4	0%	100%	100%	100%	4	0%	100%	100%	100%	6	100%	100%	100%	100%	0%

7 Conclusions and Future Research

The enrichment of process events logs with situations derived by a CEP engine, demonstrates that temporal contextual information can be leveraged to improve the adequacy of explanations given for process execution instances. As also demonstrated, with some tweaking of process outcomes, such enrichment can also be employed in retrospect. The effect of the enrichment manifests itself not just in properly adjusting the importance of factors that correctly correspond to the outcome, but also promotes the accuracy of the surrogate ML model.

Our contribution is not only in showing the effect of situational enrichment in attaining better explainability, but also in providing a core taxonomy for the different types of situational events, an overall methodological approach on how to realize the enrichment, and a corresponding test scheme. All these elements have been instantiated with respect to the illustrative example using PROTON and

SHAP, highlighting some concrete caveats, such as the need to include Boolean features for meaningful partitioning of quantitative ones to benefit from SHAP’s capacity to identify feature inter-dependencies. Tasks duration as originally captured in timestamps may also unveil impactful interpretations that we haven’t considered in this work, but could be an interesting direction to explore next.

The transposing of a process log into a tabular representation in which all process execution attributes are flattened into a single immutable row may be criticized for having a naive view of a process as a single processing step. We acknowledge that, in reality, a process can better be seen as a sequence of subsequent real-time choices with casual relationships among its steps, some derived from an “in-flight” incomplete view about overall process state. Our approach can also be applied with respect to any sub-partitioning over the set of process variables and with respect to any intermediary execution result as the target variable for the explanation.

We foresee two fundamental directions that remain open for future work. First is designing for an even tighter integration between the XAI framework and the BP engine, one in which insights inferred by the former serve as feedback for the conditional unfolding of the process execution in real-time. This may be attained by extending process flow logic with execution decisions that rely on the evolving ‘explainable state’ of the process’ instance.

A second and probably the most challenging direction is the development of new aids to infer concrete situational enrichment that may be missing in a given process. Reduced ML model accuracy may denote that some situational condition is still missing. However, to date, it is mainly the responsibility of a domain expert to specify such situations. Future work may also attend to the effort of automatic identification of which concrete situations may be missing.

References

1. van der Aalst, W.: *Process Mining: Data Science in Action*. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
2. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* **6**, 52138–52160 (2018)
3. Amit, G., Fournier, F., Gur, S., Limonad, L.: Model-informed LIME extension for business process explainability. In: *PMAI@IJCAI*. Vienna (2022)
4. Benoit, L., et al.: Hype cycle for the internet of things (2021)
5. Dumas, M., et al.: Augmented business process management systems: a research manifesto. *arXiv preprint arXiv:2201.12855* (2022)
6. Etzion, O., Niblett, P.: *Event Processing in Action*. Manning (2010)
7. Grosskopf, A., Decker, G., Weske, M.: *The Process: Business Process Modeling Using BPMN*. Meghan-Kiffer Press (2009)
8. Guidotti, R., et al.: A survey of methods for explaining black box models. *ACM Comput. Surv.* **51**(5), 1–42 (2018)
9. Harel, D., Marelly, R.: *Come, Let’s Play: Scenario-Based Programming Using LSCs and the Play-Engine*. Springer, Heidelberg (2003). <https://doi.org/10.1007/978-3-642-19029-2>

10. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **30** (2017)
11. Meske, C., Bunde, E., Schneider, J., Gersch, M.: Explainable artificial intelligence: objectives, stakeholders, and future research opportunities. *Inf. Syst. Manag.* **39**(1), 53–63 (2022)
12. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
13. Rehse, Jana-Rebecca., Mehdiyev, Nijat, Fettke, Peter: Towards explainable process predictions for Industry 4.0 in the DFKI-smart-Lego-factory. *KI - Künstliche Intelligenz* **33**(2), 181–187 (2019). <https://doi.org/10.1007/s13218-019-00586-1>
14. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why should i trust you?” Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144 (2016)
15. Schulte, W.R., et al.: Market guide for event stream processing (2022)
16. Upadhyay, S., Isahagian, V., Muthusamy, V., Rizk, Y.: Extending LIME for business process automation. arXiv preprint [arXiv:2108.04371](https://arxiv.org/abs/2108.04371) (2021)
17. Verma, S., Lahiri, A., Dickerson, J.P., Lee, S.I.: Pitfalls of explainable ML: an industry perspective. arXiv preprint [arXiv:2106.07758](https://arxiv.org/abs/2106.07758) (2021)
18. Weske, M.: *Business process management architectures*. In: *Business Process Management*. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-59432-2_8

**6th International Workshop on Business
Processes Meet Internet-of-Things
(BP-Meet-IoT 2022)**

6th International Workshop on Business Processes Meet Internet-of-Things (BP-Meet-IoT 2022)

The Business Process Management (BPM) discipline, as it is known today, emerged as the result of significant advances experienced since the mid-1990s in business methods, tools, standards, and technology. Since then, this discipline has significantly evolved but mainly focused on the business domain with the objective of helping organizations to achieve their goals. However, the arrival of the Internet of Things (IoT) has put into play a huge amount of interconnected and embedded computing devices with sensing and actuating capabilities that are revolutionizing our way of living. The incorporation of this technology into the BPM field has the potential to make business processes (BPs) aware and adaptive to their execution environment and its changes. In addition, the proper combination of these two fields (IoT and BPM) can foster the development of innovative solutions not only in the business domain where BPM emerged, but also in many different application areas in which IoT can be applied (e.g., smart cities, smart agriculture, smart factories, e-health).

Whereas the integration of IoT technology and BPM opens up plenty of opportunities, it also imposes a set of challenges that need to be addressed. In particular, research is needed for addressing questions such as:

- What is the impact of introducing IoT technology into the BPM lifecycle?
- How the top-down and bottom-up paradigms in which BPM and IoT rely on respectively can coexist and benefit each other when merged?
- How to bridge the gap between the low and the high-level in which IoT and BPM operate respectively?
- How BPM will deal with the changing nature imposed by IoT technology?
- How real-time communication and collaboration required in IoT systems will be supported by BPM?
- How to consider privacy aspects in data captured by IoT devices and analyzed with BPM?
- How BPM can be used to model behavior in IoT systems?

The objective of this workshop was therefore to attract novel research that tackles these challenges as well as creating a space for discussion and interactions between the research communities dealing with the integration between IoT and BPM fields.

The 6th edition of this workshop attracted seven international submissions, each of them was reviewed by three members of the program committee. The following four submissions were finally accepted and selected for presentation:

- “Method to Identify Process Activities by Visualizing Sensor Events”, authored by Flemming Weyers, Ronny Seiger and Barbara Weber. In this paper, the authors presented a step-by-step method in order to (manually) identify activities from sensor events. This method relied on useful visualization techniques that could be used to show and process sensor data over time in a meaningful way.

- “A Holistic Framework for IoT-Aware Business Processes”, authored by Yusuf Kirikkayis, Florian Gallik and Manfred Reichert. This paper presented an extension of BPMN 2.0 for modeling, executing, and monitoring IoT-aware business processes.
- “Assessing the Suitability of Traditional Event Log Standards for IoT-Enhanced Event Logs”, authored by Yannis Bertrand, Jochen De Weerd and Estefanía Serral. This paper identified important challenges for the integration of IoT data in event logs, deriving requirements for a suitable IoT-enhanced event log model. Using these requirements, limitations of the main two event log standards, namely XES and OCEL, were shown.
- “vAMoS: eVent Abstraction via Motifs Search”, authored by Gemma Di Federico and Andrea Burattin. This paper presented a trace-based approach for event abstraction that focused on the identification of motifs on the traces. The objective of the approach was to recognize the series of activities that are common in most traces (i.e., motifs) to determine the higher-level activities that could be recognized in the log.

Before these papers were presented, we had a very insightful keynote by Prof. Juan Carlos Augusto, head of the research group on the Development of Intelligent Environments at Middlesex University. The keynote was titled: “Contexts and Context-awareness for BPM”. During this keynote, it was made clear the importance of properly understanding and taking into account the context that is relevant for the modelling and execution of a business process.

The final sessions of the workshop consisted of a very lively discussion around the topic of event log formats for IoT process mining. The discussion started with a presentation by the chairs on the limitations of current event log standards for tackling the task at hand, proposing different possible options to address these limitations. A very interesting and up to date presentation of the current developments of the OCED standard proposal by Prof. Claudio Di Ciccio and Prof. Pnina Soffer followed. Afterwards, the participants were divided into groups following a World Cafe approach to discuss several important challenges and needs on event log formats to enable IoT process mining.

The workshop was celebrated on September 12, 2022, and attracted about 40 participants who actively interacted during the workshop presentations and the discussions. The organizers of this event would like to specially thank the authors of the submitted papers as well as the keynote and active participants in the fruitful discussions. We would also like to note the valuable input from the PC members and conference organizers who facilitated the workshop. We hope that the reader finds the final selection of papers interesting and useful to get a better insight into the integration of IoT and BPM from both theoretical and practical points of view.

Organization

BP-Meet-IoT 2022 Workshop Chairs

Francesco Leotta	Sapienza Università di Roma, Italy
Massimo Mecella	Sapienza Università di Roma, Italy
Estefanía Serral	KU Leuven, Belgium
Victoria Torres	Universitat Politècnica de València, Spain

BP-Meet-IoT 2022 Program Committee

Andrea Delgado	Universidad de la República, Uruguay
Felix Mannhardt	Eindhoven University of Technology, The Netherlands
Andreas Oberweis	Karlsruhe Institute of Technology, Germany
Claudio Di Ciccio	Sapienza University of Rome, Italy
Pnina Soffer	University of Haifa, Israel
Jianwen Su	University of California at Santa Barbara, USA
Juan Manuel Murillo Rodríguez	University of Extremadura, Spain
Udo Kannengiesser	Johannes Kepler University Linz, Austria
Sylvain Cherrier	Université Paris Est Marne la Vallée, France
Adrian Mos	Naver LABS, France
Mathias Weske	University of Potsdam, Germany
Faruk Hasić	KU Leuven, Belgium
Vicente Pelechano	Universitat Politècnica de València, Spain
Jan Mendling	Humboldt-Universität zu Berlin, Germany
Zakaria Maamar	Zayed University, UAE



Assessing the Suitability of Traditional Event Log Standards for IoT-Enhanced Event Logs

Yannis Bertrand^(✉), Jochen De Weerd^t, and Estefanía Serral^d

Research Centre for Information Systems Engineering (LIRIS), KU Leuven,
Warmoesberg 26, 1000 Brussels, Belgium
{yannis.bertrand,jochen.deweerd,estefania.serralasensio}@kuleuven.be

Abstract. Since IoT devices supporting business processes (BPs) in sectors like manufacturing, logistics or healthcare can collect data on the execution of the processes, there is a growing awareness of the opportunity to use IoT data for process mining (PM). However, several challenges need to be addressed to enable IoT-enhanced PM, among which the need for a standard IoT-enhanced event log model. In this paper, we identify four main challenges for the integration of IoT data in event logs, from which 10 requirements for an IoT-enhanced event log model are derived. We then analyse the two main current event log models, the extensible event stream (XES) and the object-centric event log (OCEL), and confront them with the requirements, showing that both present important limitations to storing IoT-enhanced event logs. Based on this analysis, we conclude that a comprehensive data model is required to develop more advanced IoT-enhanced PM techniques.

Keywords: Process mining · IoT · IoT-enhanced event log

1 Introduction

As IoT devices, i.e., sensors and actuators, are more widely used to support the execution of business processes (BPs), there is a growing awareness of the opportunity to use the data collected by these devices for process mining (PM). Currently available PM methods capable of incorporating IoT data are consistent in terms of strategy: IoT data are preprocessed in such a way that they can be integrated in a classical event log format. This necessitates abstraction techniques, either data- or expert-driven. Although this is a good first step in including IoT data in PM, given the fact that it allows for application of the existing set of control-flow and data-aware methods, this approach does not use IoT data to their full potential. Most of the time, the derived high-level event log omits context information (i.e., properties that can influence the execution

This research was supported by Flanders Research Fund (FWO) in the scope of project number G0B6922N.

of the process, see [29]) that could be derived from the IoT data, or is limited in the extent to which context information can be incorporated. Moreover, by separating the abstraction phase from the discovery/analysis phase, the real potential of advanced algorithms to jointly optimise abstraction and model discovery, cannot be exploited. For instance, developing an IoT-enhanced decision mining algorithm would require direct access to lower-level IoT data in order to learn the most relevant features directly on the source data, instead of an error-prone and lossy event abstraction step.

This observation relates strongly to limitations of the most commonly used event log standards, i.e. the extensible event stream (XES, see [13]) and the object-centric event log (OCEL, see [12]). Accordingly, in this paper, we present four concrete contributions to further research on IoT-enhanced process mining. First, we motivate the problem by describing possible new techniques that would use IoT data to their full potential and demonstrate the impossibility to apply them to a typical event log. Second, we put forward four specific challenges that pertain to the integration of IoT data in event logs for PM, and illustrate these challenges with a running example. Third, based on these challenges, we present 10 key requirements that a suitable model for IoT-enhanced event logs should fulfil. Finally, the fourth contribution of this work consists of an assessment of XES and OCEL in function of the identified requirements, to determine their suitability and shortcomings.

Thus, this paper is structured as follows. Section 2 provides the motivation. Section 3 discusses related work, before the running example is introduced in Sect. 4. The challenges and requirements are put forward in Sects. 5 and 6, respectively. Section 6.1 analyses XES and OCEL before concluding the paper in Sect. 7.

2 Motivation

Table 1 provides a high-level overview of methods and techniques in the PM field. The table is structured vertically along the three main process mining task types. In the horizontal dimension, the evolution from a classical control-flow perspective to data-aware and IoT-enhanced approaches is depicted. The second and third column cover the lion's share of process mining research so far. However, an important gap is still present in terms of developing IoT-enhanced process mining techniques. The most prominent existing stream of research relates to activity mining, i.e. approaches to derive higher level process events from low-level IoT data [10, 17, 21, 28, 34, 36]. Nonetheless, as illustrated by an only evocative set of so far unaddressed research opportunities, denoted in italics in Table 1, the widespread and grounded development of IoT-enhanced process mining techniques, covering all process mining tasks and possible applications, is far from realised yet. This is mainly due to the fact that the very source of the opportunities of IoT data is their fine granularity, which enables a deeper understanding of the process and the detection of finer patterns. As illustrated by a non-exhaustive set of examples in the table, there exist many potential new techniques that could be developed to take advantage of IoT data. For example:

- *IoT context-aware trace clustering*: cluster traces based on the values of IoT-derived context parameters.
- *IoT-driven root cause analysis*: delve into IoT measurements to discover the root cause of process deviations.
- *IoT-enhanced decision mining*: incorporate IoT-derived context parameters in decision mining.
- *IoT-enhanced resource mining*: use IoT data (e.g., location, tags) to mine the fine-grained behaviour of resources in a process.
- *IoT-enhanced predictive process monitoring*: use IoT sensor data measurements to facilitate real-time predictive process monitoring.

A common denominator of these envisioned IoT-enhanced PM techniques is the fact that simply abstracting fine grained IoT data into a classical XES or OCEL event log considerably limits analysis possibilities. Accordingly, this paper provides a foundation for the development of a new standard adapted to IoT-enhanced event logs, which is intended to give enough flexibility to develop new techniques taking advantage of the possibilities of IoT data.

Table 1. Positioning of IoT-enhanced techniques within the PM field, extending control-flow and data-aware approaches for all three main PM task types.

		Control-flow	Data-aware	IoT-enhanced
PM task types	Discovery	<ul style="list-style-type: none"> • Control-flow discovery [1,19,37] • Trace clustering [6,31] 	<ul style="list-style-type: none"> • Data-aware process discovery [20,23] • Multi-perspective trace clustering [14] • Multi-perspective process discovery [22] 	<ul style="list-style-type: none"> • Activity mining [10,17,21,27,28,34,36] • <i>IoT context-aware trace clustering</i> • IoT-enhanced data-aware process discovery [2]
	Conformance checking	<ul style="list-style-type: none"> • Control-flow conformance checking [9] 	<ul style="list-style-type: none"> • Data-aware conformance checking [38] 	<ul style="list-style-type: none"> • IoT-enhanced deviation detection [26] • <i>IoT-driven root cause analysis</i>
	Extension		<ul style="list-style-type: none"> • Decision mining [3] • Resource mining [24] • Predictive process monitoring [8] • Performance analysis [5] 	<ul style="list-style-type: none"> • <i>IoT-enhanced decision mining</i> • <i>IoT-enhanced resource mining</i> • <i>IoT-enhanced predictive process monitoring</i> • <i>Real-time predictive process monitoring</i>

To further substantiate our motivation, let us consider a lifelike example to illustrate the main limitations of current standards. This example consists in a smart distribution centre (DC), which receives products from warehouses, assesses their quality and distributes them to supermarkets (adapted from [35]). In this process, several instances (e.g. shipments, crates) are processed in parallel, several decisions are made at different points, involving data from multiple sensors and following more complex rules than simple thresholds.

The process begins with the arrival of a container loaded with crates of products at the smart DC. The first activity is a manual check of the quality and freshness (based on, e.g., firmness, colour, damages) of a sample of the products. The results are entered in the system by the worker performing the check. After this, information about the product (e.g., name, harvest or production date) is retrieved by scanning the label (i.e., the QR code) of the shipment, together with data on the transport conditions recorded by sensors (e.g. temperature, humidity, shocks). Based on the worker's assessment and the evaluation of the overall transport conditions, each crate of products is judged proper for consumption or not. If it is, the crate is registered and stored in a suitably acclimatised refrigerated area to maximise product conservation. Otherwise, the crate is rejected and discarded.

After the first product quality check, a second one is performed over a sample of the products at the laboratory. If bacteria are detected, an alarm is triggered and the crate is discarded. Otherwise, the crate can be shipped. Depending on the quality of the products (as determined by the worker, the transport and storage conditions and the lab analysis), the crate will either be moved to the non-priority shipment area (if the products are excellent), or the crate will be set for priority shipment to be sold as fast as possible, at a discount (if the quality is not excellent). Finally, pallets are registered as shipped when they have left the DC to the supermarket.

It is known that the human evaluation of the freshness of the products is highly correlated with the transport conditions tracked by sensor data. We would like to discover a threshold value from the data (i.e., using IoT-enhanced decision mining), but without more domain knowledge, it is necessary to store the whole time series sensor data in the process event log to look for relationships (correlations) between temperature and humidity values and the decision. However, it is not possible to store both data types in a typical event log. Some sort of aggregation or abstraction needs to be applied to the time series sensor data first, which requires knowing the relationship between sensor data and the process flow beforehand; but that is not the case here, as this relationship is what we would like to mine.

3 Related Work

3.1 Existing Standards for Event Logs

XES [13], the current standard event log model, is an XML-based model that mainly consists of the notions of event, case, and log. It proposes standard

attribute types to contextualise the events, e.g. the resource executing an activity, the cost of an activity, etc. A standard activity lifecycle is defined together with XES, based on which the status of an activity could be mapped with events relating to this activity. XES also allows the definition of new data attribute types through the notion of extensions, thereby increasing the flexibility of the model. Several implementations coexist, the main one being OpenXES¹, which is used by many event logs described in the literature.

Recently, the uptake of new technologies and the gain in maturity of the PM field have increased the urge to create alternative models. Multiple propositions that relax some assumptions of XES and allow for more flexibility in event data storage have been presented, e.g., in [12,25]. Among them, the OCEL [12] was designed to be more suitable for storing event logs extracted from relational databases and is widely considered as the main challenger of XES today. It replaces the strict notion of case with the concept of object, which generalises it by allowing one event to be linked with multiple objects instead of a single case. This removes the necessity to “flatten” the event log by picking one case notion from the several potential case notions that often coexist in real-life processes. A second noticeable difference with XES is the explicit inclusion of the concept of activity in OCEL, which is absent in XES.

3.2 Process Mining Using IoT Data

As mentioned in Sect. 2, the vast majority of the process mining literature involving IoT data has focused on mining high-level events of the process from low-level IoT data to create XES event logs (so-called *activity mining* in Table 1). Traditional process mining techniques can then be applied to these event logs to, e.g., discover control-flow models of the processes.

Trzcionkowska and Brzywczy describe a framework to create an event log from industrial IoT data in four steps: data preprocessing, clustering low-level data, classification to derive events from clusters and creation of the final event log [34]. The event log obtained is in XES format and contains no data attribute. Also focusing on industrial applications, Seiger et al. propose to transform raw IoT data into an XES event log using complex event processing (CEP) and event detection and refinement techniques [28]. They apply this approach to a smart manufacturing case to mine a production process in a follow-up paper [27]. In Valencia-Parra et al., a domain-specific language is developed to extract different XES event logs from IoT data by specifying the case and activity identifiers [36]. Koschmider et al. [18] propose a model to go from low-level events captured by sensors to instances of the process, by aggregating low-level events into high-level events using methods like CEP. In a further work, the authors showed a more systematic and complete framework [17], which assumes a sensor event log as input, and consists of three event log creation steps: (1) event correlation, (2) activity discovery, (3) event abstraction. This approach generates an XES event log, from which a process model can be mined, and was applied to a smart home scenario by Janssen et al. [16].

¹ <https://www.xes-standard.org/openxes/start>.

Although most of the existing literature is in activity mining, some of the other possible techniques have also been investigated. Banham et al. [2] proposes to perform data-aware process discovery with IoT-based attributes. A data Petri Net is discovered from two real-life event logs, and rules behind some decisions are mined based on IoT-derived attributes. The framework proposed requires abstracting the IoT data to integrate them in an XES event log. A second work is proposed by Rodriguez-Fernandez et al. [26], who present an approach for IoT-enhanced deviation detection. In their paper, they argue that traditional conformance checking cannot take into account data that can change over time independently of the events of the process (i.e., time series data). They propose a method to detect patterns in the time series data directly (in a so-called *time-series log*). Remark that these papers bumped into the limitations of traditional event logs: Banham et al. [2] had to abstract the IoT data, which implied making important assumptions, and Rodriguez-Fernandez et al. [26] only used the time series data as input to their technique because they could not integrate them in an event log.

4 Challenges

Using this running example as illustration, we pinpoint four key challenges of IoT data that make it difficult to integrate them in a traditional event log without having prior knowledge about the process or making non-trivial assumptions. These challenges are established based on an in-depth analysis of the literature (see Sect. 3) and based on the authors' experience in a number of case studies [33]. Observe that the literature also mentions other challenges than the ones listed here, e.g., data quality or data volume. However, we focused on the challenges that 1) have a direct influence on the mining and 2) pertain to the format of the event log.

C1: Granularity. The granularity of the sensor can be considered a topmost challenge [4, 10, 17, 28, 30, 39]. Process decisions are usually not made based on the raw sensor values, but rather depend on aggregations of the value of a sensor over a certain time or on combinations of the values of several sensors. E.g., the first decision is based on the humidity and temperature inside the refrigerated truck over the whole transport. Granularity is usually dealt with in the event abstraction step, i.e. it is considered as preprocessing. However, without prior knowledge, there is no way to know which aggregation or combination of sensor data should be applied to retrieve the relevant high-level context parameter that determines the decision, and the event abstraction is a mining step, which should be done based on the event log.

C2: Perspective convolution. IoT data can relate to events of the control-flow or to the context of the process, or to both [2, 15, 30]. E.g., accelerometer data could tell that crates of a product are being shaken during transport (context) or that the crates are being loaded in the truck (control-flow), a spike in fridge

temperature can indicate that the fridge is opened to put in or take out crates (control-flow), etc. However, in existing event log standards, it is required to choose between using the IoT data as process events or attributes of a case or an event.

C3: Scope of relevance. The relevance of the IoT data, especially when it relates to the context of a process, may not always be limited to an event or a case, but can relate to several events or several cases [10, 15, 17]. E.g., the temperature in the truck does not necessarily impact only one crate of products, but usually many crates of different types of products transported in the same truck. And logging the value of temperature (and other variables) separately for each event of each product or crate potentially leads to duplicating huge chunks of data.

C4: Dynamicity. IoT data are inherently dynamic and not always synchronised with the process [26]. The dynamicity of context parameters is usually coped with by allowing data attributes to be updated in process events. However, and as also shown with the A-B-C process example, IoT data are often loosely coupled with the process. In the motivating example, the temperature is sensed at regular intervals during the transport of the products, and not only when events happen in the process. Although there is no guarantee that the sensor data will have relevant values when process events are logged, this is usually the only point when context parameters can be updated. But without previous knowledge, it is not possible to know which sensor measurement(s) is or are relevant. E.g., the observation of temperature that impacts the decision to keep a crate of fresh products or not can be made at any moment before the decision point, and there is no way to log it without an event. But this event 1) would not be linked with a process activity and 2) this requires prior knowledge on the impact of temperature on the decision, which we assume we do not have, as it is the sort of effects that we would like to discover with PM.

5 Requirements for an IoT-Enhanced Event Log Model

Based on the challenges identified previously, we put forward 10 requirements that a suitable data model for IoT-enhanced event logs should meet.

- R1: Store high-level events
- R2: Store low-level events
- R3: Store intermediary/mixed-level events
- R4: Enable traceability between high-level and low-level events
- R5: Represent context at event level
- R6: Represent context at activity level
- R7: Represent context at case/object level
- R8: Represent context at process level
- R9: Update context parameters independently from process events
- R10: Update context parameters at a higher frequency

Second, R9-10 stipulate that context parameters should be able to evolve with time at their own rhythm (C4), independently of control-flow events (C2). This aims at catering for situations such as the one depicted by the A-B-C process example, where sensor data that are relevant for a decision can be observed at different moments during the execution of the process, i.e., typically at any point within a certain time frame, and not only when events happen.

6 Assessment of Current Event Log Standards

6.1 Comparison of Existing Models

In this section, we confront existing event log models (XES and OCEL) with the requirements listed in Sect. 5 to point out their limitations.

Table 3. Comparison of XES and OCEL with respect to the Requirements.

		XES	OCEL	Source
Requirements	R1: Store high-level events	✓	✓	C1
	R2: Store low-level events			C1
	R3: Store intermediary/mixed-level events			C1
	R4: Enable traceability between high-level and low-level events			C1
	R5: Represent context at event level	✓	✓	C3
	R6: Represent context at activity level			C3
	R7: Represent context at case/object level	✓	✓	C3
	R8: Represent context at process level	✓		C3
	R9: Update context parameters independently from process events			C2
	R10: Update context parameters at a higher frequency			C4

XES. So far, nearly all event logs used in IoT-based PM were XES logs. However, these event logs were already abstracted and most of the time IoT data were only used to retrieve process events and not to add context information.

XES typically stores data at a high level of abstraction, as events or attributes (fulfilling R1), which means that IoT data need to be abstracted into events or attributes to be recorded in a XES log, potentially losing important bits of information to the abstraction step. Moreover, abstracting the IoT data considerably restricts the spectrum of analyses that can be performed with the IoT data. This entails that XES cannot meet R2-4, which stipulate that it should be possible to store data at different granularity levels in a same event log and allow for traceability between them.

The elements that store all the information are the attributes. However, attributes in XES are defined over an event, a trace or the log (R5,7,8; R6 is not fulfilled). This makes it difficult for an IoT context variable, which can often impact several cases at the same time, to be related to all the cases it applies to (R9). Moreover, this means that new or updated context information cannot be recorded outside of a process event, while R10 stipulates that context parameters (especially physical phenomena tracked by IoT devices) sometimes have interesting values at arbitrary moments.

OCEL. In line with XES, OCEL typically stores data at a high level of granularity (R1), which means that IoT data need to be abstracted into events or attributes to be saved in an OCEL log too, incurring the same risk of losing important information and restricting the possibilities of analysis of the IoT-enhanced log. This contradicts R2-4.

Then, like in XES, all context information is represented by means of attributes of events or objects (R5,7; R6 and R8 not fulfilled). Moreover, the only way to log a new value of an attribute is with an event, which goes against R9-10.

7 Conclusion

In this paper, we pleaded for a wider look at the possible uses of IoT data in process mining. While the existing literature largely tackles the issue of extracting process events from IoT data (so-called *activity mining*), other potential uses of IoT data listed in Table 1 remain almost entirely unexplored. One of the main hurdles complicating the development of new techniques is the restrictions that are imposed by the current main standards for event logs, namely XES and OCEL. All the data have to be stored as high-level events or attributes of high-level events or cases (objects in the case of OCEL), leaving context as a second-class citizen. This makes it necessary to abstract the low-level IoT data to store them in event logs, thereby losing information and making important assumptions on the influence of the phenomena tracked by IoT devices on the process. Moreover, to make these assumptions, an extensive knowledge of the working of the process is required, which may not be available when doing process mining, as acquiring this deeper knowledge of the process is often the goal of process mining itself.

To solve this issue, we listed requirements for a suitable model for IoT-enhanced event logs. Then, we confronted XES and OCEL, the two main standards for event logs at the moment, with these requirements, and showed that they are both currently unsuitable for the storage of IoT-enhanced event logs. We therefore claim that a dedicated comprehensive event log model is needed to develop more advanced IoT-enhanced PM techniques.

Note that some of the challenges are not only encountered when dealing with IoT data. The representation of the context in XES and OCEL can be a limitation to more traditional process mining as well, e.g., some non-IoT context parameters can change outside of process events and a blurry definition of the context can lead to complex process models (see [7]). Moreover, the issue of granularity is found in many other processes and is discussed outside of the literature on PM with IoT data (e.g., [11,32]).

In future works, we plan to define and formally implement a new model for IoT-enhanced event logs. We also intend to investigate additional real-life examples and use cases to show the use of a new standard. At a later stage, we would like to develop algorithms that can be applied to such event logs to realise the IoT-enhanced PM techniques evoked in Sect. 2. This may also

require to examine the suitability of current modelling languages to represent such IoT-enhanced processes, and propose new ones, e.g., combining BPMN with ontologies to give semantics to the context of the process.

References

1. van der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **16**(9), 1128–1142 (2004)
2. Banham, A., Wynn, M.T.: xPM: a framework for process mining with exogenous data, p. 12 (2021)
3. Bazhenova, E., Buelow, S., Weske, M.: Discovering decision models from event logs. In: Abramowicz, W., Alt, R., Franczyk, B. (eds.) *BIS 2016. LNBIP*, vol. 255, pp. 237–251. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39426-8_19
4. Bertrand, Y., De Weerd, J., Serral Asensio, E.: A bridging model for process mining and IoT. In: *ICPM Workshops Proceedings* (2021)
5. Bozkaya, M., Gabriels, J.: Process diagnostics: a method based on process mining, p. 7 (2009)
6. De Weerd, J., Vanden Broucke, S., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. *IEEE Trans. Knowl. Data Eng.* **25**(12), 2708–2720 (2013). <https://doi.org/10.1109/TKDE.2013.64>
7. Dees, M., Hompes, B., van der Aalst, W.M.: Events put into context (EPiC). In: *ICPM*, pp. 65–72 (2020)
8. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Milani, F.: Predictive process monitoring methods: which one suits me best? In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *BPM 2018. LNCS*, vol. 11080, pp. 462–479. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_27
9. Dunzer, S., Stierle, M., Matzner, M., Baier, S.: Conformance checking: a state-of-the-art literature review. In: *S-BPM ONE 2019 Proceedings*, pp. 1–10 (2019)
10. van Eck, M.L., Sidorova, N., van der Aalst, W.M.P.: Enabling process mining on sensor data from smart products. In: *RCIS*, pp. 1–12 (2016)
11. Fazzinga, B., Flesca, S., Furfaro, F., Pontieri, L.: Process discovery from low-level event logs. In: Krogstie, J., Reijers, H.A. (eds.) *CAiSE 2018. LNCS*, vol. 10816, pp. 257–273. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91563-0_16
12. Ghahfarokhi, A.F., Park, G., Berti, A., van der Aalst, W.M.P.: OCEL: a standard for object-centric event logs. In: Bellatreche, L., et al. (eds.) *ADBIS 2021. CCIS*, vol. 1450, pp. 169–175. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85082-1_16
13. Günther, C.W., Verbeek, H.M.W.: XES standard definition (2014)
14. Jablonski, S., Röglner, M., Schöning, S., Wyrski, K.M.: Multi-perspective clustering of process execution traces. *EMISAJ* **14**, 1–22 (2019)
15. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A.E.A.: The internet-of-things meets business process management: a manifesto. *IEEE Syst. Man Cybern. Mag.* **6**(4), 34–44 (2020)
16. 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
17. Koschmider, A., Janssen, D., Mannhardt, F.: Framework for process discovery from sensor data, p. 8 (2020)

18. 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
19. 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
20. de Leoni, M., van der Aalst, W.M.P.: Data-aware process mining: discovering decisions in processes using alignments. In: SAC 2013, pp. 1454–1461 (2013)
21. Mannhardt, F., Bovo, R., Oliveira, M.F., Julier, S.: A taxonomy for combining activity recognition and process discovery in industrial environments. In: Yin, H., Camacho, D., Novais, P., Tallón-Ballesteros, A.J. (eds.) IDEAL 2018. LNCS, vol. 11315, pp. 84–93. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03496-2_10
22. Mannhardt, F.: Multi-perspective process mining. Ph.D. thesis (2018)
23. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Data-driven process discovery - revealing conditional infrequent behavior from event logs. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 545–560. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_34
24. Nakatumba, J., van der Aalst, W.M.P.: Analyzing resource behavior using process mining. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 69–80. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_8
25. Popova, V., Fahland, D., Dumas, M.: Artifact lifecycle discovery. *Int. J. Coop. Inf. Syst.* **24**(01), 1550001 (2015)
26. Rodriguez-Fernandez, V., Trzcionkowska, A., Gonzalez-Pardo, A., Brzywczy, E., Nalepa, G.J., Camacho, D.: Conformance checking for time-series-aware processes. *IEEE Trans. Industr. Inform.* **17**(2), 871–881 (2021)
27. Seiger, R., Malburg, L., Weber, B., Bergmann, R.: Integrating process management and event processing in smart factories: a systems architecture and use cases. *J. Manuf. Syst.* **63**, 575–592 (2022)
28. Seiger, R., Zerbato, F., Burattin, A., Garcia-Banuelos, L., Weber, B.: Towards IoT-driven process event log generation for conformance checking in smart factories. In: EDOCW, pp. 20–26 (2020)
29. Serral, E., De Smedt, J., Vanthienen, J.: Making business environments smarter: a context-adaptive petri net approach. In: UIC 2014, pp. 343–348 (2014)
30. Soffer, P., et al.: From event streams to process models and back: challenges and opportunities. *Inf. Syst.* **81**, 181–200 (2019)
31. Song, M., Günther, C.W., van der Aalst, W.M.P.: Trace clustering in process mining. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008. LNBIP, vol. 17, pp. 109–120. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00328-8_11
32. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Event abstraction for process mining using supervised learning techniques. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2016. LNNS, vol. 15, pp. 251–269. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-56994-9_18
33. Torres, V., Serral, E., Valderas, P., Pelechano, V., Grefen, P.: Modeling of IoT devices in business processes: a systematic mapping study. In: CBI 2020, vol. 1, pp. 221–230 (2020)

34. Trzcionkowska, A., Brzywczy, E.: Practical aspects of event logs creation for industrial process modelling. *Multidiscip. Asp. Prod. Eng.* **1**(1), 77–83 (2018)
35. Valderas, P., Torres, V., Serral, E.: Modelling and executing IoT-enhanced business processes through BPMN and microservices. *J. Syst. Softw.* **184**, 111139 (2022)
36. Valencia-Parra, A., Ramos-Gutierrez, B., Varela-Vaca, A.J., Gomez-Lopez, M.T., Bernal, A.G.: Enabling process mining in aircraft manufactures: extracting event logs and discovering processes from complex data, p. 12 (2019)
37. Weijters, A.J.M.M., van Der Aalst, W.M., De Medeiros, A.A.: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP **166**(July 2017), 1–34 (2006)
38. van der Werf, J.M.E.M., Verbeek, H.M.W., van der Aalst, W.M.P.: Context-aware compliance checking. In: Barros, A., Gal, A., Kindler, E. (eds.) *BPM 2012. LNCS*, vol. 7481, pp. 98–113. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32885-5_7
39. Zerbato, F., Seiger, R., Di Federico, G., Burattin, A., Weber, B.: Granularity in process mining: can we fix it? In: *CEUR Workshop Proceedings*, vol. 2938, pp. 40–44 (2021)



Method to Identify Process Activities by Visualizing Sensor Events

Flemming Weyers, Ronny Seiger^(✉), and Barbara Weber

Institute of Computer Science, University of St. Gallen, St. Gallen, Switzerland
ronny.seiger@unisg.ch

Abstract. With the onset of the Internet of Things (IoT) everyday objects suddenly become data sources equipped with sensors measuring the object’s properties and surroundings. However, the lack of process-awareness in IoT environments (e.g., smart factories) prevents the adoption of more sophisticated process analysis and optimization. One hurdle is the differing abstraction level of low-level sensor data and process-level activities. We propose a method to identify activities step-by-step from raw IoT data using visualizations. By relying on minimal process knowledge, we discover process activities from sensor events. These activities are represented by specific sequences of sensor events—*Activity Signatures*—that serve as a basis for finding similar activities. We demonstrate the method’s validity with a proof of concept in a smart factory.

Keywords: Business Process Management · Cyber-physical systems · Internet of Things · Activity detection · Sensors

1 Introduction

With the onset of the Internet of Things (IoT), more and more domains are pervaded with sensors and actuators controlled by software [22]. Physical objects suddenly produce data about their state, surroundings and the processes they are involved in [10]. However, process-awareness in the sense of Business Process Management (BPM) is still very-low in IoT as there usually is no workflow management system (WfMS) available to orchestrate or monitor processes [15]. The *BPM-IoT Manifesto* discusses various challenges and benefits of bringing both domains together [6]. With this work, we investigate how process activity executions in IoT can be linked to sensor data from the respective IoT devices and vice versa. The goal is to discover activities from raw IoT sensor data, thereby addressing the challenge of “Bridging the Gap Between Event-Based and Process-Based Systems” [6]. Existing approaches like supervised machine learning (ML) rely on inputs from the WfMS (e.g., activity labels). Unsupervised ML struggles with feature selection, especially with thousands of sensors as inputs. We propose a novel visualization-based method to enable the identification of activities from raw sensor data that serves as basis for future automation of sensor data analysis. The following research questions guide our investigations:

- RQ1** What are reasonable assumptions and steps to analyze sensor data for process activities in IoT?
- RQ2** How can sensor data be associated with activity executions to potentially automate the detection of activities from sensor data?

The paper is structured as follows: Sect. 2 introduces relevant background. Section 3 discusses related work. Section 4 introduces the new method for activity detection from sensor data. Section 5 demonstrates and discusses the method’s validity. Section 6 concludes this paper and gives an outlook on future work.

2 Basic Concepts and Context

This work is closely related to research at the intersection of BPM and IoT/CPS (Cyber-physical Systems). We propose to adapt the “Ingredients of a business process” [3] combined with the “UML representation of the IoT Domain Model” [1] as basic conceptual model for bringing both fields together (cf. Fig. 1).

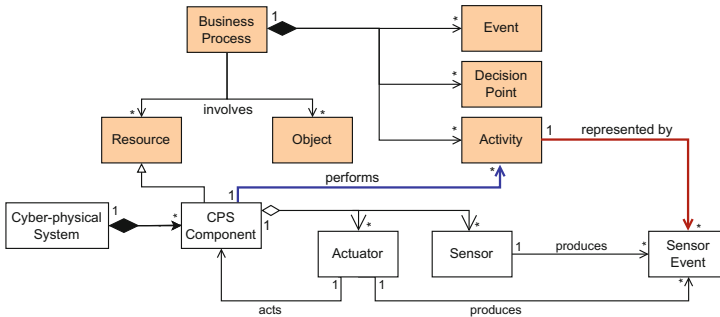


Fig. 1. Meta-model with basic concepts from BPM and CPS used in this work.

2.1 Basic Concepts

Business Process Management: BPM is “[...] a well-established discipline that deals with the identification, discovery, analysis, (re)design, implementation, execution, monitoring, and evolution of organizational procedures” [3]. *Business Processes* are chains of events, activities and decisions to achieve a desired outcome [3]. *Activities* can be both fine-grained or coarse-grained units of work [3]. *Resource* in the context of BPM is “[...] a generic term to refer to anyone or anything involved in the performance of a process activity” [3].

Cyber-Physical Systems and Internet of Things: CPS integrate computation and physical processes where both the digital and physical systems affect each other [11]. In IoT, everyday objects are interconnected and work together to

accomplish an objective [10]. While IoT focuses on interoperability and communication among devices, CPS put emphasis on control and automation relying on IoT for connectivity. As this fits well with our research, we use the term *Cyber-physical System (CPS)* throughout this paper. In *Smart Factories*, hardware/software components that are composed of sensors, actuators and controllers work together in manufacturing cells to achieve a production outcome [15]. We call these self-contained cells *CPS Components* of a smart factory representing CPS. We treat CPS components as resources that execute activities of a production-related business process [12]. Our investigations focus on the relation between sensor data and activity executions by the CPS components (cf. Fig. 1).

Sensors and Actuators: *Sensors* monitor a physical entity and provide information about its physical and virtual properties [10]. They can be attached to, or embedded in the entity’s structure or be placed in its environment [10]. *Actuators* modify the entity’s physical state or influence other entities’ functionalities (e.g., via motors or valves in the CPS components) [10]. They also produce data (e.g., regarding their states) that we treat as sensor data. Thus, we refer to both, data from sensors and actuators, as *Sensor Data/Sensor Events* in this paper.

2.2 Context: Fischertechnik Factory Model

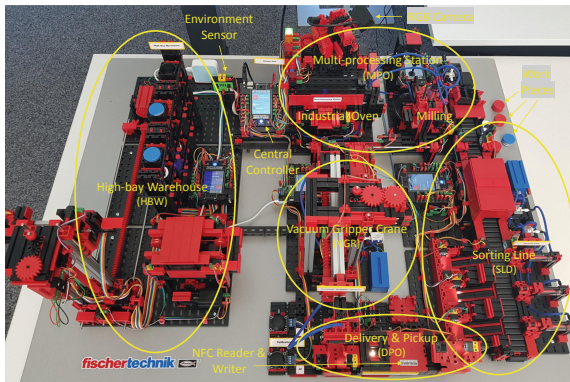


Fig. 2. Smart factory simulation model as a CPS representative [15].

Figure 2 shows the Fischertechnik smart factory model that represents CPS in our work [15]. Each highlighted station is one CPS component (e.g., HBW, VGR, SLD). The factory features realistic discrete manufacturing processes and CPS components, each equipped with a multitude of sensors (e.g., light-barriers, switches) and actuators (e.g., motors, valves) [12]. We rely on the software stack presented in [14] for controlling the smart factory.

```

1  {"UUID": "91c4fd59-27d7-477b-ae18-2ef8b6f04cb5",
2   "timestamp": "2022-02-23 10:34:23.72",
3   "i_1": 1, "...", "i_n": 0,
4   "o_1": 0, "...", "o_n": 0,
5   "m_1_speed": 0, "...", "m_n_speed": 5.12,
6   "target_pos_x": 0, "target_pos_y": 0, "target_pos_z": 0}

```

Fig. 3. Exemplary payload in JSON for one message in the *VGR* topic.

Sensor and Actuator Event Streams: An MQTT (Message Queuing Telemetry Transport [19]) broker streams sensor events from the smart factory during its operation. We use one *Topic* per CPS component, which is a message channel for clients to subscribe to and receive messages. Figure 3 shows the payload of an MQTT message including multiple attributes that represent individual sensor and actuator events: unique identifier of the message (Line 1); timestamp when the message is generated (Line 2); $i_1 \dots i_n$ for input values from sensors (Line 3); $o_1 \dots o_n$ for states of output devices, e.g., valves or compressors (Line 4); $m_1 \dots m_n$ for motor speeds (Line 5); additional component-related parameters, e.g., target positions (Line 6). We subscribe to all topics (= CPS components from Fig. 2) and record the messages for offline analysis.

2.3 Process Awareness in Sensor Data

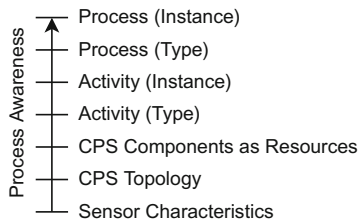


Fig. 4. Bottom-up process awareness associated with sensor data.

Process execution knowledge (called *Process Awareness*) that can be associated with sensor data exists on a spectrum (cf. Fig. 4) [2]. It ranges from only knowing the characteristics of individual sensors (as discussed, e.g., in [8]) to being able to associate a concrete activity and process instance with a given sensor event (as discussed, e.g., in [13]). Our basic assumption is that a WfMS does not always exist to coordinate and monitor process executions in CPS [15]. Our goal is

to gradually increase process awareness in sensor data following a bottom-up approach.

3 Related Work

CPS introduce new approaches, e.g., for condition monitoring or predictive maintenance based on recorded sensor data [5]. Massive amounts of data, different data formats, sampling rates, and data quality are among the challenges that come with using sensor data [9]. Existing works use different types of data for the discovery of events and activities at different levels [2]. Koschmider et al. provide a framework to discover processes from sensor data. Accordingly, we focus on “Activity Discovery”, and “Event Abstraction”, where we relate events to the start or completion of process activities [8].

Going from sensor data to data suitable for process mining poses challenges regarding event extraction, abstraction, and event correlation [2,6]. Identifying relevant data for process mining and extracting it from different sources is part of *Event Extraction* [2]. This data often resides in traditional databases and information systems in the form of *Event Logs*. Existing approaches identify this data using, e.g., database schemas, process documents, domain models, event models, and/or domain knowledge [2,7]. In our work, we use sensors and actuators in CPS as data sources. Event abstraction in the context of BPM focuses on the *abstraction gap* between the granularity at which the data is recorded and at which it is analyzed [2,6,25]. When considering sensor data, the challenge of mapping fine-grained sensor data to more abstract process activities becomes more pronounced [6,24]. In literature [24], various approaches exist to bridge this abstraction gap using, e.g., Complex Event Processing (CEP) [15,18] or machine learning (ML) [4,8,20].

Most works assume rather high levels of process knowledge when discovering activities and processes from sensor data (cf. Fig. 4), i.e., existing activity labels only have to be connected to the raw events [2,13]. With almost no process knowledge and limited CPS topology knowledge (lower end of the spectrum, cf. Fig. 4) most approaches are not applicable. Supervised ML needs activity labels to learn. Unsupervised methods like clustering could find activities, yet the amount of features (thousands of sensor) make it hard to gain valuable insights. We exploit existing CPS topology knowledge and assume CPS components to act as resources to then use data visualizations for building a *Knowledge Base* (KB) of CPS-based activities which can be labeled by domain experts and be used for further identification and labeling of activities in unknown datasets.

4 Method to Identify Activities from Sensor Events

The method to identify activities from sensor data is comprised of multiple steps to visualize data, filter by CPS component, and incrementally refine activities (cf. Fig. 5). We base the method on the *Visual Information Seeking Mantra*: “Overview first, zoom and filter, then details-on-demand” [17]. In the following, we explain each step from a conceptual point and illustrate it with an example.

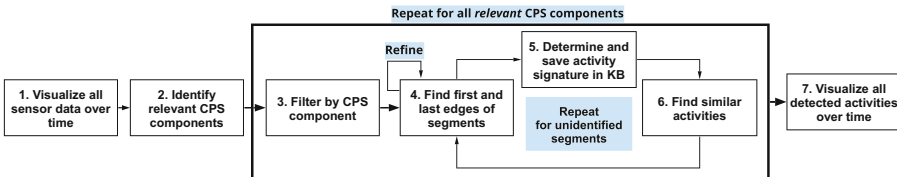


Fig. 5. Method to identify activities by visualizing sensor events.

4.1 Visualize All Sensor Data Over Time

We plot all sensor data from a given dataset to provide an overview for the analyst (*Overview First* [17]). The y-axis shows the concrete values of all sensors and the x-axis shows the associated timestamps.

Example: Figure 6 shows all sensor data from our smart factory over a recorded timeframe. Each graph represents one sensor from a CPS component, e.g., *VGR_i1* refers to sensor *i1* from *VGR* (cf. Sect. 2.2). Even with this small CPS setting, there is a plethora of graphs for all sensors that populate data (cf. Fig. 3).

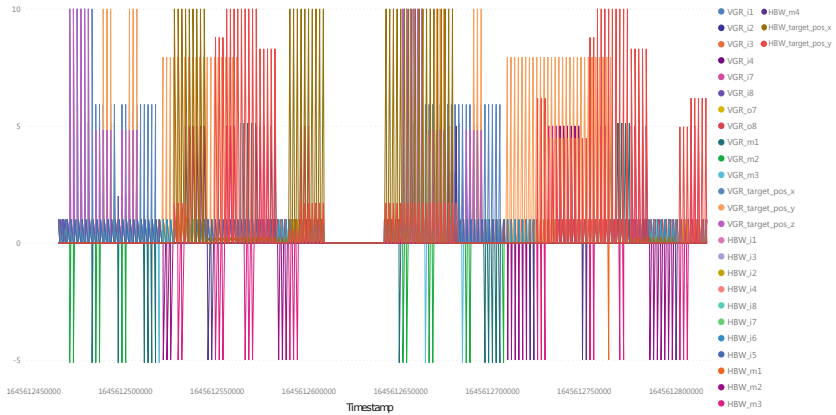


Fig. 6. Recorded sensor data from smart factory model.

4.2 Identify Relevant CPS Components

The *relevant* CPS components are identified. We assume that components relevant for the activity identification show changes within their associated sensor data when executing activities. However, not all CPS components that populate sensor data are also executing activities. This step cannot be fully automated as the relevance should be confirmed by the analyst.

Example: We identify the vacuum gripper crane (“VGR”) as relevant CPS component since the values of its associated sensors change over time (cf. Fig. 7). An example for an “irrelevant” component is an environment sensor constantly measuring temperature that may not be associated with a specific activity.

4.3 Filter by CPS Component

We *filter* the sensor data by the first relevant CPS component to only visualize data related to one CPS component. Here we assume that one activity is executed by one CPS component (cf. Fig. 1), i.e., only one CPS component shows

changes in its sensor values that are relevant for activity detection. This is step is **repeated** for **all relevant** CPS components.

Example: Figure 7 shows the sensor data for component “VGR” which we identified as a relevant component in step 2. Still, the entire recorded timeframe is shown.

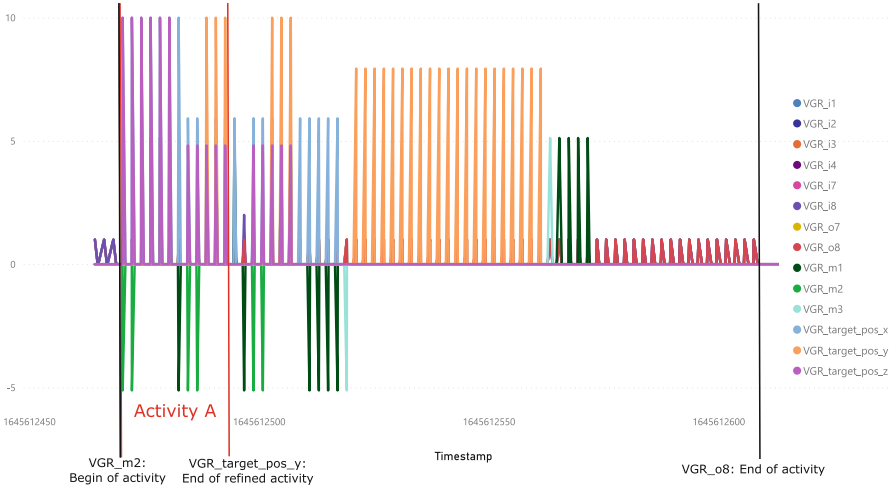


Fig. 7. Identifying and refining activity boundaries for the VGR component based on the first and last edges of segments.

4.4 Identify First and Last Edges of Segments

We search for first and last edges of segments that can be indications of activities. We assume that one CPS component executes only one activity at a time (i.e., there is no batch processing, but only discrete manufacturing steps [21]). By definition an actuator performs work. Thus, an actuator becoming “active” or the occurrence of a “start pattern” (i.e., a combination/sequence of multiple sensors/actuators becoming active) are good indications for an activity’s start. The same holds for the end when an actuator stops or an “end pattern” occurs. Times where CPS components are inactive can be short “breaks” that are part of the execution of one activity or they can be an indicator for an activity’s end. Additionally, we can factor in *context* to differentiate activities, e.g., the switch to another CPS component is an indicator for an activity’s end rather than a break. Since we may not be able to distinguish between the execution of two activities by the same CPS component in sequence, we have to *zoom in* to the identified activity and **repeat** this step to **refine** an activity. An indication for these refinements could be that the set of the CPS component’s involved sensors/actuators or the pattern of sensor data changes significantly. This strongly relies on the visualization and analyst’s knowledge.

Example: In Fig. 7 we see the first raising and last falling edges of an activity block. With “VGR_m2”, an actuator (here: motor) of VGR is switched on which indicates that an activity starts. The last edge shows “VGR_o8” (here: a compressor) switching off followed by a break which indicates the end of the activity. Figure 8 shows this identified activity (black borders). This figure also shows the result of refining the activity (red borders): the VGR executes transport activities and the sensor “VGR_target_pos_y” provides context as to where the crane moves. Since this value changes in the next segment we assume that one transport activity stops and another starts. We now have identified a single activity “Activity A” that does not need to be refined further. As stated before, these refinement steps depend heavily on the knowledge of the analyst.

4.5 Determine and Save Activity Signature in Knowledge Base (KB)

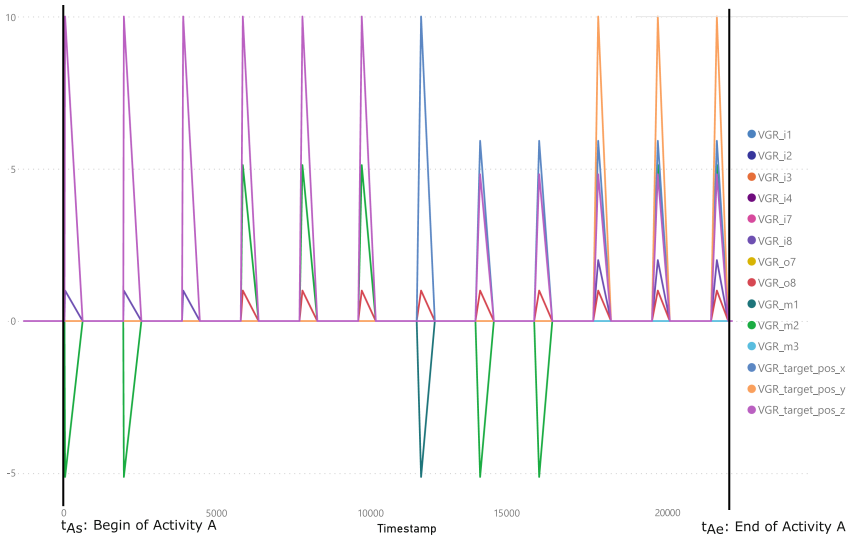


Fig. 8. Activity signature of “Activity A” executed by CPS component “VGR”.

The *Activity Signature* of the identified activity is determined (*details-on-demand* [17]). This signature refers to the distinct sequence of all sensor data for the specified CPS component within the identified activity boundaries, i.e., its start time t_{As} and its end time t_{Ae} . The determined activity signature is saved in a KB with a label (e.g., “Activity A”) and the respective sensor data as multivariate time series for all timestamps t_n : $t_{As} \leq t_n \leq t_{Ae}$. Thereby, the absolute event timestamps are replaced with relative timestamps starting at $t_{As} = 0$ for the start of the activity until the end of the activity t_{Ae} to search for similar activity signatures within the given dataset in the next step.

Example: Figure 8 shows the *Activity Signature* for “Activity A” executed by the vacuum gripper crane (“VGR”). More specific activity labels have to be provided by the analyst. This signature is stored in a time series database.

4.6 Find Similar Activities

We look for activities based on similarity with the determined signature. This can be achieved by zooming out and visually finding the activity signature for the specific CPS component within the dataset. Visual pattern matching quickly becomes infeasible for determining similarity here since activity signatures can easily grow in complexity, i.e., they may consist of a multitude of sensors and actuators, and patterns within the sensor data over a longer period of time. Moreover, our data shows that the same activities do not necessarily have identical activity signatures, e.g., due to minor variations in sensor and actuator behavior or different process parameters. Thus we need a way to approximate the similarity of the time-series data where the analyst can define a threshold for when an activity is accepted as similar [16]. We are currently investigating the use of *Matrix Profiles* in multi-variate time series as a novel way of calculating these approximations to determine the similarity of signatures [23].

Example: Figure 9 shows multiple identified activities along the entire timeline. The activity signatures of activities “A”, “B”, “C” and “D” were determined in the previous steps. We zoomed out and found similar segments in the dataset that can also be marked as “A”, “B”, “C” or “D”.

4.7 Visualize All Detected Activities over Time

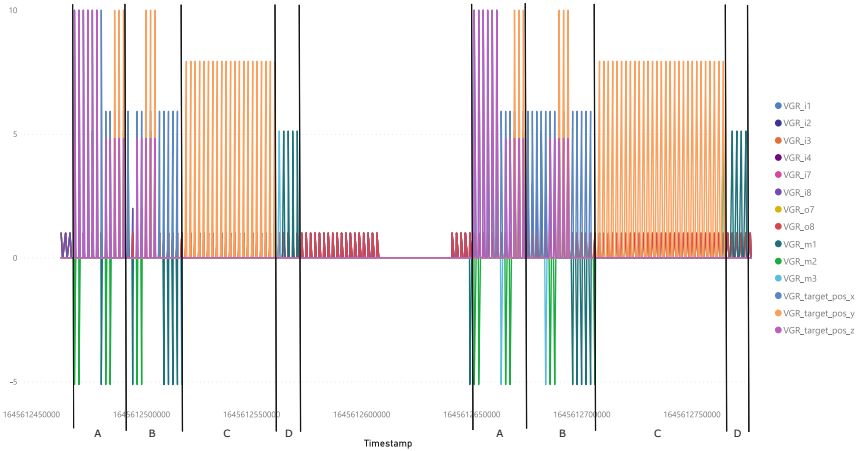


Fig. 9. All detected activities for the CPS component VGR over entire timeline.

After repeating Steps 3–6 in Fig. 5 for relevant CPS components and repeating Steps 4–6 identify new activities within unidentified segments, all detected activities are visualized for all CPS components over the entire timeline to identify process instances. Recurring patterns of activity sequences might be an indication for the execution of different process instances of the same process. However, we cannot fully say if these repeated sequences could also be part of the same instance. Moreover, we limit our approach based on the assumption that only one process instance can be executed at a time, which is reasonable for many discrete manufacturing settings [21].

Example: Figure 9 shows recurring sequences of activities “A”, “B”, “C” and “D” for the VGR in chronological order. This might indicate that two instances of the same process were executed in sequence. Unlabeled segments are either parts that we were not able to identify or classified as noise or inactivity by the analyst.

5 Proof of Concept and Discussion

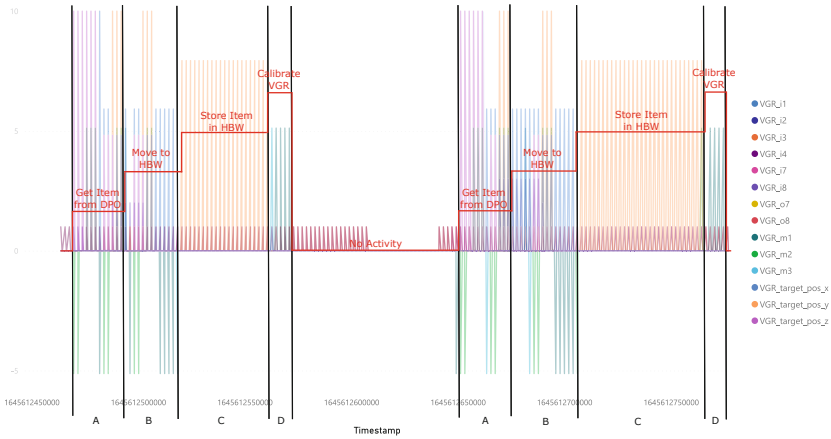


Fig. 10. Overlay of identified activities with event log data.

The software stack used for data recording also features a WfMS for process execution and generating an *Event Log* [14]. When developing the method (cf. Sect. 4), we assumed that the analyst does not have access to this event log. To provide a proof of concept showing the validity of our proposed method in this early development stage, we overlaid the event log data with the activities we identified from the sensor data (cf. Fig. 10). The red graph shows the mapped activities with their actual labels as executed by the WfMS. Apart from minor temporal delays resulting from different timestamp resolutions, we can see an

almost exact match between the logged activities and the identified activities. The example also nicely shows that activity signatures for the same type of activity (cf. “Store Item in HBW”) may differ as explained in Sect. 4.6.

Considering the research questions, we were able to answer RQ1 by proposing a visualization-based method for analyzing sensor data to identify activity executions while assuming minimal process knowledge and making reasonable assumptions for the domain of discrete manufacturing. Regarding RQ2, we are able to associate sensor data with identified activities based on the novel concept of *Activity Signatures* that can be used to automate the detection of similar activities. However, not all steps of the method can be completely automated. Especially the identification of relevant CPS components and the refinement of identified activities rely on the expertise of the analyst. Our proposed method follows a bottom-up approach to increase process awareness step-by-step (cf. Sect. 2.3). It is suited to identify activities as part of the *control flow* perspective as well as the process resources (i.e., CPS components) that performed them. We can only provide abstract activity labels without relying on further domain knowledge. Although it was not an explicit goal, we are able to make statements about the correlation of detected activities with process instances based on the assumption that no batch processing or parallel process execution is performed.

6 Conclusion and Future Work

In this work, we proposed a method to identify activities from sensor data following a bottom-up approach. Assuming a low degree of process awareness and limited knowledge about CPS, we apply various steps of *overview*, *filter and zoom*, and *details on demand* [17] to visually identify activity executions from sensor data. We also provide first approaches towards automating steps within the method (e.g., based on the new concept of *activity signatures*). A proof of concept evaluation with actual event log data from a WfMS has shown promising results regarding the applicability of our method in smart manufacturing.

In future work, we will relax some of the assumptions made for the initial version of the method (e.g., to also allow for parallel process executions). A larger case study with data from our smart factory will be next. With this, we will investigate the feasibility of the method when working with larger datasets. In this context, we will further develop the concept of *Activity Signatures* to automatically detect activities in unknown datasets based on similarity with known activities [16, 23]. We will also investigate if we are able to generate stream processing applications from activity signatures to enable online activity detection [15].

References

1. Bauer, M., et al.: IoT reference model. In: Bassi, A., et al. (eds) Enabling Things to Talk, pp. 113–162. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40403-0_7

2. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.* **10**(3), e1346 (2020)
3. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*, vol. 1, p. 2. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-33143-5>
4. Folino, F., Guarascio, M., Pontieri, L.: Mining predictive process models out of low-level multidimensional logs. In: Jarke, M., et al. (eds.) *CAiSE 2014. LNCS*, vol. 8484, pp. 533–547. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07881-6_36
5. Hoppenstedt, B., et al.: Techniques and emerging trends for state of the art equipment maintenance systems—a bibliometric analysis. *Appl. Sci.* **8**(6), 916 (2018)
6. Janiesch, C., et al.: The internet of things meets business process management: a manifesto. *IEEE Syst. Man Cybern. Mag.* **6**(4), 34–44 (2020)
7. Jans, M., Soffer, P., Jouck, T.: Building a valuable event log for process mining: an experimental exploration of a guided process. *Ent. Inf. Syst.* **13**(5), 601–630 (2019)
8. 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
9. Kammerer, K., Pryss, R., Hoppenstedt, B., Sommer, K., Reichert, M.: Process-driven and flow-based processing of industrial sensor data. *Sensors* **20**(18), 5245 (2020)
10. Kramp, T., van Kranenburg, R., Lange, S.: Introduction to the internet of things. In: Bassi, A., et al. (eds) *Enabling Things to Talk*, pp 1–10. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40403-0_1
11. Lee, E.A.: Cyber physical systems: design challenges. In: 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), pp. 363–369 (2008)
12. Malburg, L., Seiger, R., Bergmann, R., Weber, B.: Using physical factory simulation models for business process management research. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) *BPM 2020. LNBIP*, vol. 397, pp. 95–107. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66498-5_8
13. Mangler, J., Pauker, F., Rinderle-Ma, S., Ehrendorfer, M.: centurio.work—Industry 4.0 integration assessment and evolution. In: 17th BPM Conf., pp. 106–117 (2019)
14. Seiger, R., Malburg, L., Weber, B., Bergmann, R.: Integrating process management and event processing in smart factories: a systems architecture and use cases. *J. Manuf. Syst.* **63**, 575–592 (2022)
15. Seiger, R., Zerbato, F., Burattin, A., García-Bañuelos, L., Weber, B.: Towards IoT-driven process event log generation for conformance checking in smart factories. In: 24th Intern. EDOC Workshop, pp. 20–26. IEEE (2020)
16. Serrà, J., Arcos, J.L.: An empirical evaluation of similarity measures for time series classification. *Knowl.-Based Syst.* **67**, 305–314 (2014)
17. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: *The Craft of Information Visualization*, pp. 364–371. Interactive Technologies, Morgan Kaufmann (2003)
18. Soffer, P., et al.: From event streams to process models and back: challenges and opportunities. *Inf. Sys.* **81**, 181–200 (2019)
19. Standard, O.: MQTT version 5.0 (2019)

20. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Event abstraction for process mining using supervised learning techniques. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) *IntelliSys 2016*. LNNS, vol. 15, pp. 251–269. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-56994-9_18
21. Traganos, K., Grefen, P., Vanderfeesten, I., Erasmus, J., Boultradakis, G., Bouklis, P.: The HORSE framework: a reference architecture for cyber-physical systems in hybrid smart manufacturing. *J. Manuf. Syst.* **61**, 461–494 (2021)
22. Whitmore, A., Agarwal, A., Da Xu, L.: The internet of things—a survey of topics and trends. *Inf. Syst. Front.* **17**(2), 261–274 (2014). <https://doi.org/10.1007/s10796-014-9489-2>
23. Yeh, C.C.M., et al.: Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 1317–1322 (2016)
24. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. *Granul. Comput.* **6**(3), 719–736 (2020). <https://doi.org/10.1007/s41066-020-00226-2>
25. Zerbato, F., Seiger, R., Di Federico, G., Burattin, A., Weber, B.: Granularity in process mining: can we fix it? In: *CEUR WS Proc.*, vol. 2938, pp. 40–44 (2021)



A Holistic Framework for IoT-Aware Business Processes

Yusuf Kirikkayis^(✉), Florian Gallik, and Manfred Reichert

Institute of Databases and Information Systems, Ulm University, Ulm, Germany
{yusuf.kirikkayis,florian-1.gallik,manfred.reichert}@uni-ulm.de

Abstract. The Internet of Things (IoT) enables a variety of smart applications, including smart home, smart factory, and smart health. As Business Process Management (BPM) can also benefit from IoT technologies, the combined use of BPM and IoT has attracted considerable research works. Providing integrated lifecycle support for modeling, executing, and monitoring IoT-aware business processes constitutes a challenge. Existing process modeling and execution languages such as BPMN 2.0 are unable to fully meet the requirements of IoT-aware processes. In this paper, we present an extension of BPMN 2.0 for modeling, executing, and monitoring IoT-aware business processes. We introduce specific artifacts and events that enable IoT awareness during the execution and monitoring of IoT-driven business processes. The resulting framework is illustrated along a real-world scenario.

Keywords: BPMN · IoT · IoT-aware BPM · Execution engine · BPMS

1 Introduction

The interest and relevance of the Internet of Things (IoT) has been increasing continuously during the recent years and IoT has become one of the most relevant technologies to realize digital twins of the physical world [1]. The electronic components of IoT devices are becoming smaller, cheaper, and more powerful. As a result, IoT technology is experiencing an upswing [2]. IoT devices are equipped with sensors, actuators, software, protocols, and various network interfaces. This enables IoT devices to capture, collect, and exchange data as well as to physically respond to events [3]. While sensors are used to collect and capture data about the physical world (e.g., humidity, air quality, and temperature), actuators are used to control the latter (e.g., watering systems, light control, air conditioner, and security systems) [4]. While IoT enables the collection and exchange of data about the physical world, BPM enables modeling, implementing, executing, monitoring, and analyzing business processes [5]. Incorporating IoT capabilities into BPM systems, therefore, offers promising perspectives for process automation including automated decision making that takes the state of the physical world into account as well. Moreover, IoT devices can be used to automate various

types of physical tasks (e.g. opening a window) or digital tasks (e.g. transferring data) [5]. To be able to provide lifecycle support for IoT-aware processes their modeling requires specific elements that allow capturing the physical process context appropriately. Amongst others modeling IoT-aware processes shall foster the understanding of how these processes operate as well as enable the detection and avoidance of problems. Moreover, it should be possible to detect and handle errors and exceptions (e.g., faulty sensors) during process enactment. Existing standards such as BPMN 2.0 do not provide sufficient expressiveness for modeling IoT-aware processes [3].

In this paper, we enhance BPMN 2.0 with IoT-specific artifacts and events, which enable the modeling, execution, and monitoring of IoT-aware processes. The functions and benefits of these artifacts and events are illustrated along a real-world manufacturing process in a smart factory. The remainder of this paper is structured as follows. In Sect. 2, we summarize the problems that emerge when modeling IoT-aware processes with the standard BPMN 2.0 language. Section 3 describes the proposed extension, i.e., specific artifacts and events. In Sect. 4 we present our approach for modeling, executing, and monitoring IoT-aware business processes, which is then applied in the context of a case study in Sect. 5. Section 6 discusses related work. Finally, Sect. 7 summarizes the approach and provides an outlook on future work.

2 Problem Statement

Though BPMN 2.0 does not provide explicit support for capturing and modeling IoT capabilities, it offers various ways to represent IoT aspects such as tasks, events, and resources [6]. However, following such a straightforward approach, no distinction between IoT-related process model elements and regular elements can be made [14]. Consequently, it does not become apparent whether or not a process includes IoT aspects. Instead, the process model needs to be read and understood based on the chosen element (e.g. task) labels.

Figure 1 illustrates a process with IoT aspects modeled in terms of BPMN 2.0. Along this example, we want to demonstrate characteristic problems. The depicted production process involves multiple actuators as well as sensors. The process starts every ten seconds and then checks whether the High-Bay Warehouse (HBW) light barrier is interrupted. If the latter applies, the Vacuum Gripper Robot (VGR) starts moving, otherwise the process terminates. However, the VGR is moved until reaching the pick-up station whose light barrier check is embedded within a loop. After the VGR has reached the pick-up station, a QR code is generated. Subsequently, the status of the HBW is checked. If the status is *OK*, the workpiece is stored in HBW, otherwise it is transported to HBW 2. Finally, the process terminates.

When using BPMN 2.0 for modeling the IoT-related tasks (cf. Fig. 1), it is unclear, which tasks involve IoT devices and which do not. It is further unclear that business rule tasks shall represent sensors and service tasks shall represent actuators. Instead, the process model reader needs to interpret the task labels

correctly to properly understand the process model. Moreover, no visual distinction can be made between an IoT-related service task (cf. Tasks 2, 6, and 7 in Fig. 1) and a service task not involving IoT devices (cf. Task 4), or between an IoT-related business rule task (cf. Tasks 1 and 3) and a normal one (cf. Task 5). Note that this aggravates both the readability and the comprehensibility of the process model.

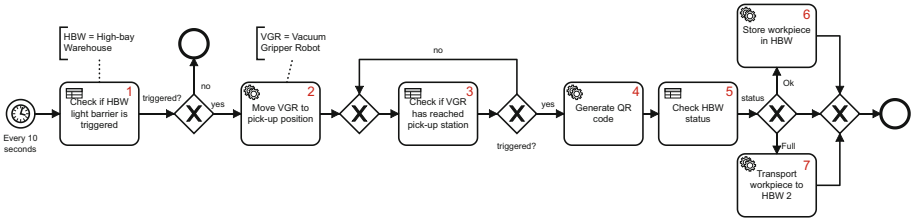


Fig. 1. IoT-aware business process modeled in terms of BPMN 2.0.

3 Solution Proposal

The goal of this paper is to provide a BPMN 2.0 extension that enables the modeling and enactment of IoT-aware processes. Moreover, the behavior of these elements needs to be mapped to a process execution engine, which constitutes the core architecture component of our approach. Taking the problem statement set out in Sect. 2, we extended BPMN with the artifacts and events shown in Fig. 2. In the following, each of these elements is shortly described. Note that all elements are decorated with a WLAN icon and labeled as “IoT”. In addition, the letter in the upper left corner indicates the artifact type (i.e., “A” for actuator and “S” for sensor).

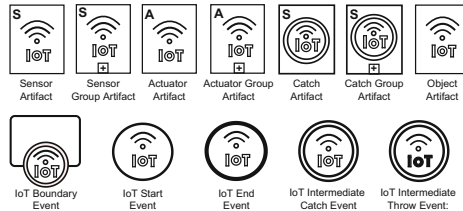


Fig. 2. Extending BPMN 2.0 with IoT-specific elements

Sensor Artifact: A sensor artifact (Fig. 2) can represent various sensors in a business process model (e.g., measuring temperature, speed, or GPS), and enables the collection of data from the physical environment and process context. When connecting a sensor artifact to a task, the corresponding sensor may

be queried by the task during its execution. All necessary information about the sensor is captured by the sensor artifact. The task connected to it can only be successfully completed after having received a positive response from the sensor. Note that the explicit representation of sensors as artifacts allows linking any number of sensors to a task (Fig. 3a), and a sensor artifact may be arbitrarily combined with other artifacts (Fig. 3b). In such a case, the sensors are concurrently queried during task execution. Note that the representation of the artifact is generic allowing for the representation of arbitrary sensor types. *Moreover, text annotations may be used, for example, to designate artifacts and events.*

Sensor Group Artifact: A sensor group artifact is represented by a collapsed sensor artifact (cf. Fig. 2) and shows the same behavior as a sensor artifact. As depicted in Fig. 3a, individual sensor artifacts may be aggregated to a sensor group artifact in order to increase the abstraction level. More precisely, a sensor group artifact combines multiple sensor artifacts ($n \geq 2$, with n being number of sensor artifacts).

Actuator Artifact: An actuator artifact (cf. Fig. 3(b)) allows modeling actuators (e.g., electric motor, relay, light, and microphone). This enables the process to react to situations, e.g., by opening a window as soon as a certain temperature threshold is exceeded. An actuator is controlled by the task associated with the corresponding actuator artifact. All necessary information about the actuator is captured by the actuator artifact. The corresponding task is completed successfully once it has received a positive response from the actuator. Note that the representation of the artifact is generic, allowing for the representation of arbitrary actuator types.

Actuator Group Artifact: An actuator group artifact is represented by a collapsed actuator artifact (cf. Fig. 3b) and shows the same behavior as an actuator artifact. More precisely, an actuator group artifact combines multiple actuator artifacts ($n \geq 2$, with n being number of actuator artifacts). Figure 3b shows an example combining both sensor and actuator artifacts.

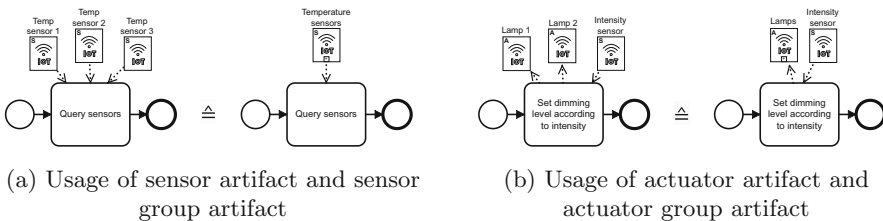


Fig. 3. Using extended IoT elements in BPMN 2.0.

Catch Artifact: A catch artifact (cf. Fig. 4a) allows checking a condition during task processing in combination with a boundary timer event. Immediately after starting the task, the respective condition is continuously checked. All necessary information about the condition is provided by the catch artifact. The task may be completed successfully only when meeting the specified condition. If the condition is not satisfied within the time period specified by the boundary timer event, the sequence flow attached to the latter is executed (cf. Fig. 4a). If other artifacts are attached to the task, their execution and verification run in parallel.

Catch Group Artifact: A catch group artifact is represented by a collapsed catch artifact. It shows the same behavior as a catch artifact (cf. Fig. 2). As depicted in Fig. 4a, the group artifact aggregates multiple catch artifacts to increase the abstraction level ($n \geq 2$, with n being number of catch artifacts).

Object Artifact: An object artifact (cf. Fig. 2) enables the modeling of physical objects (e.g., service robot, machine, or smart factory) of the environment, in which the business process is executed. As illustrated in Fig. 4b An object artifact may contain both sensors and actuators. On one hand, this allows hiding unnecessary information from domain experts. On the other, modeling becomes more accurate when using an object artifact.

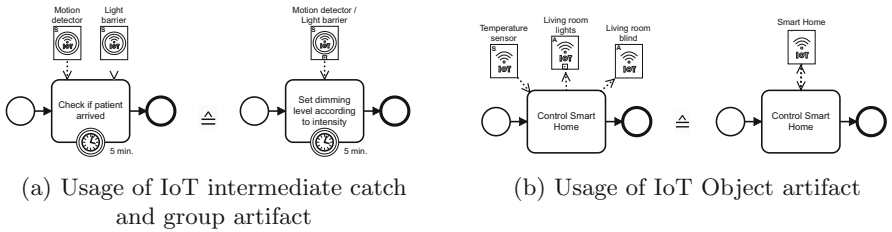


Fig. 4. Using extended IoT elements in BPMN 2.0.

IoT Boundary Event: An IoT boundary event (cf. Fig. 5a) may be used to define a condition and redirect the sequence flow accordingly if this condition becomes fulfilled during task execution. After starting the task, the condition is continuously checked. This condition checking terminates either upon task completion or when meeting the condition. *Note that all events use sequence flow as connection type.*

IoT Start Event: To enable the start of an IoT-aware process based on IoT sensors, the IoT start event (cf. Fig. 5a) can be used. It trigger a process instance when meeting the specified start condition (e.g., *temperature ≥ 20 °C* or *“motion detected”*).

IoT End Event: An IoT end event (cf. Fig. 5a) triggers the execution and/or control of an actuator and terminates the corresponding process instance. Unlike the IoT start event, the end event has a thicker border.

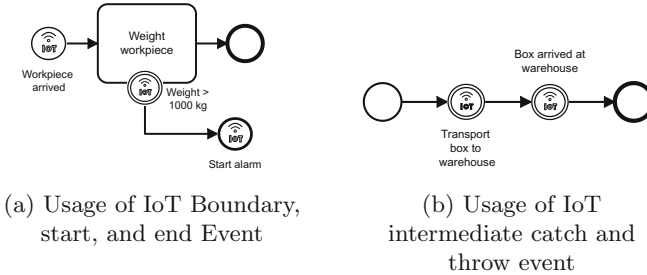


Fig. 5. Using extended IoT elements in BPMN 2.0.

IoT Intermediate Catch Event: An IoT intermediate catch event (cf. Fig. 5b) is linked to an IoT sensor. It enables the process to check a physical condition along the sequence flow (e.g. *volume > 60 decibels*). More precisely, when reaching an IoT intermediate catch event the sequence flow does not continue until its corresponding condition is met. Note that an artifact (e.g., sensor or actuator artifact) must not be linked to an IoT intermediate catch event.

IoT Intermediate Throw Event: To control an actuator along a sequence flow, the IoT intermediate throw event (cf. Fig. 5b) may be used. Semantically, such events corresponds to a task with a linked actuator artifact. However, only one actuator may be controlled at the same time in the context of an IoT intermediate throw event. The latter is successfully completed upon receipt of a positive response from the actuator. Only then, the sequence flow continues.

4 Business Process Management System for IoT

To execute and monitor IoT-aware processes based on the the described BPMN 2.0 extension, an appropriate software architecture is needed that implements the behavior of the various elements. A coarse-grained view on such an architecture is shown in Fig. 6. It consists of three main components, i.e., BPM system, MQTT broker, and IoT services. An *IoT service* controls or queries IoT devices and offers corresponding functions via its interface (e.g. REST API), while at the same time hiding technical peculiarities of the IoT devices from the calling environment (e.g. the IoT-aware process engine). For example, an IoT service may provide an endpoint to allow querying the room temperature, which can then be fetched with a GET request.

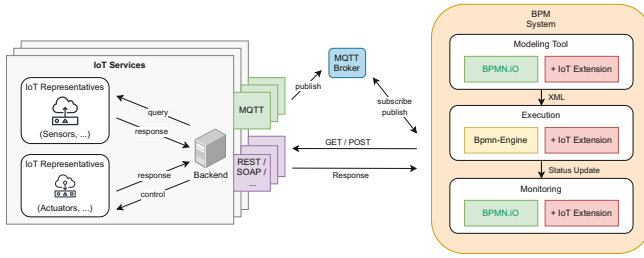


Fig. 6. Architecture of the BPM System.

The *MQTT broker* (cf. Fig. 6) is used to reduce the network load and to react to occurring IoT events. IoT devices publish their data to the broker, which, in turn, distributes these data to all subscribers. Thus, it becomes possible to react to IoT events in real-time during process execution. The *BPM system* consists of three main components, i.e., its *modeling tool*, execution engine, and monitoring system. The modeling tool is based on bpmn.io¹, extended with the elements described in Sect. 3. The latter are therefore available for modeling IoT-aware processes and can be configured for the respective execution context. A complete process model contains all information necessary to execute the IoT-aware process. The corresponding process model information is encoded in an XML file. The corresponding XML format is machine-readable and, thus, executable. The open source javascript workflow engine² serves as the basis for executing the IoT-aware processes. We extend the engine with the elements and implement their behavior (e.g., to react to IoT events by subscribing to MQTT topics, to query sensors, or to control actuators using GET/POST requests) Sect. 3. If a new process instance is created the engine reads the XML file and then starts process execution. The *monitoring component*, in turn, uses bpmn.io with the IoT extension as a basis, just like the modeling tool. However, the process model can only be viewed and not edited. The *execution engine* ensures that the state and the data of the process instance become updated in real time, i.e., users can always view the current state of the running IoT-aware process. Corresponding color markings indicate to them how far the execution of the IoT-aware process has progressed and where errors have occurred.

5 Smart Factory Scenario Process

In the following, we illustrate the modeling, execution, and monitoring of an IoT-aware process along a sophisticated smart factory scenario. Using the scenario we want to investigate whether the framework enables the generic integration of business processes with IoT capabilities. Note that we also applied the framework to IoT-aware processes from other domains such as smart home, smart healthcare, and smart logistics.

¹ <https://bpmn.io>.

² <https://github.com/paed01/bpmn-engine>.

We use physical simulation models developed by *Fischertechnik*[®](FT)³ to emulate the smart factory. These models simulate a complete production line as shown in Fig. 7. The smart factory consists of 5 stations, i.e. high-bay warehouse, vacuum gripper robot, oven, milling machine, and sorting machine.

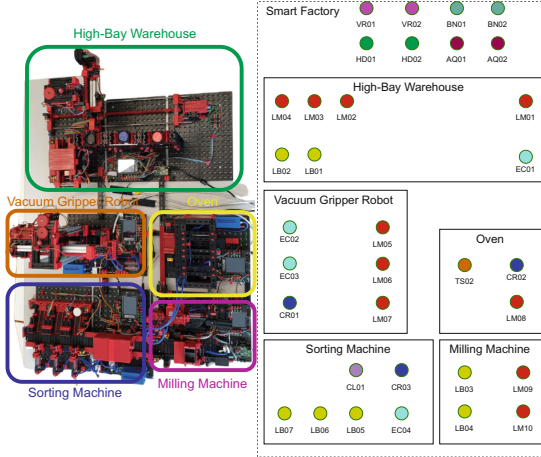


Fig. 7. Overview of sensors in the smart factory.

The smart factory is equipped with six different types of IoT sensors (cf. Fig. 7).

- Limit switch sensors (represented by red circles and labeled as *LM*).
- Light barrier sensors (represented by yellow circles and labeled as *LB*).
- Pressure sensors (represented by blue circles and labeled as *CR*).
- Temperature sensors (represented by orange circles and labeled as *TS*).
- Encoder sensors (represented by cyan circles and labeled as *EC*).
- Color sensors (represented by purple circles and labeled as *CL*).

In addition to the sensors installed in the smart factory, the scenario comprises sensors that measure the environment:

- Vibration sensors (represented by pink circles and labeled as *VR*).
- Brightness sensors (represented by green circles and labeled as *BN*).
- Humidity sensors (represented by neon green circles and labeled as *HD*).
- Air quality sensors (represented by dark red circles and labeled as *AQ*).

In total, the smart factory is equipped with 34 sensors (cf. Fig. 7), i.e. 7 light barrier sensors that detect the interruption of a light beam and display it as an electrical signal, 10 limit switch sensors actuated by the movement of a machine

³ <https://www.fischertechnik.de/en/simulating/industry-4-0>.

part or the presence of an object, 3 pressure sensors that measure the overpressure of the suction, 1 temperature sensor that measures the temperature in the oven, 4 encoder sensors that return the current position of the motors, and 1 color sensor to recognize the workpiece color in the sorting machine. In order to be able to assess the workpiece quality, 2 vibration sensors, 2 brightness sensors, 2 humidity sensors, and 2 air quality sensors are additionally used as well.

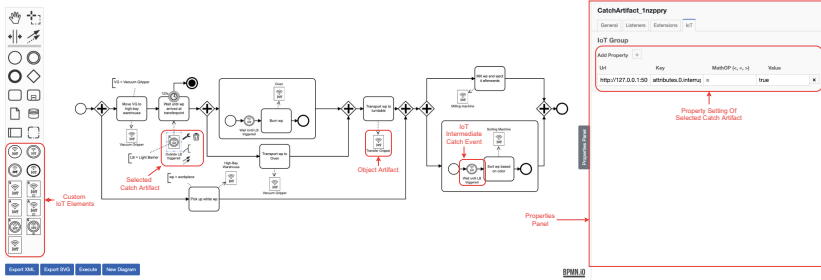


Fig. 8. Modeling and configuring IoT-aware process with the modeling tool.

Figure 8 shows an example of a manufacturing process of the smart factory that we can model, configure, execute, and monitor with our approach. First, a white workpiece (WP) is taken from the high-bay warehouse to the transfer point. In parallel, the vacuum gripper moves to the transfer point and waits there until the workpiece arrives. Waiting is realized with a catch artifact (cf. Sect. 3), which is attached to a task. This artifact checks for the triggering of a light barrier. If the condition is not met within 120s, the process terminates. As soon as the workpiece has arrived at the transfer point, the vacuum gripper transports it to the oven. The transport is realized by an object artifact (cf. Sect. 3). In parallel, the oven waits until the workpiece arrives. Upon arrival, the workpiece is burned and then transported to the turntable where it is milled. Afterwards the workpiece is moved on a conveyor belt to the sorting machine. Once the light barrier is triggered, the color of the workpiece is detected and the workpiece is sorted according to color. Then, the process terminates.

The smart factory is controlled with the Business Process Management System (BPMS) presented in Sect. 4. First of all, the IoT-aware process needs to be modeled, including the configuration of the involved IoT devices. Figure 8 shows the modeling component of the BPMS that provides all standard BPMN 2.0 elements as well as the newly introduced IoT-specific elements (cf. Sect. 4). The process model shown in Fig. 8 comprises an object artifact, an IoT intermediate catch event, and a sensor catch artifact. In the properties panel (cf. Fig. 8), the IoT condition (cf. Sect. 4) is configured using properties. Figure 8 shows the configuration of the selected sensor catch artifact. For all group artifacts as well as for the object artifact, one may use the *plus* symbol of the *Add Property* label to add another condition. The following properties need to be set for the catch

artifact: (i) an endpoint, (ii) an attribute to be accessed from the response, and (iii) the condition that needs to be satisfied in order to continue process execution. Since a boundary timer event is attached to task *Wait until WP arrived at transfer point*, the condition is checked for correctness only as long as the specified timeout has not been triggered yet. If the timeout is reached, however, the corresponding sequence flow is executed and process execution then terminates. After modeling, the process can be exported as a machine-readable XML file. This XML file contains all the information required to execute the IoT-driven process.

The modeled and configured IoT-aware process is then deployed to the execution engine of the BPM system, and new process instances may be created and started. Figure 9 shows the execution of one such instance. All elements colored in green have been successfully executed; their actual execution time is attached as a yellow colored overlay. The elements colored in orange are currently executed. In case of an error (e.g., network error or timeout during execution), the corresponding elements are colored in red. On the right side an execution log is displayed, which shows relevant events (e.g., current progress and error messages).

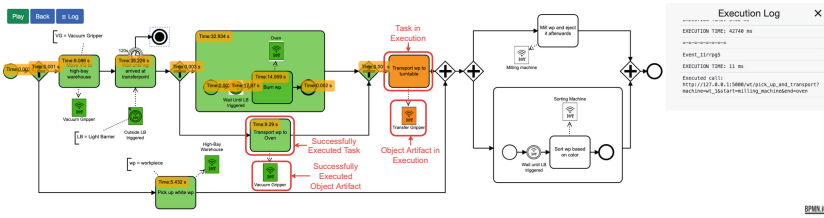


Fig. 9. IoT-aware smart factory process in execution. (Color figure online)

6 Related Work

There exist several works that introduce BPMN notations and extensions for capturing IoT aspects of business processes [12]. [7] enhances BPMN with a sensor task that covers the following aspects: (i) sensor service, (ii) sensor handler, and (iii) sensor device. For representing physical entities (e.g. a bottle of milk) a collapsed pool is used in [8]. In addition, two task types for sensing and actuation are introduced. [9] introduces a wireless sensor network (WSN) task and a WSN pool. The WSN task has an *actionType* (e.g. sensing (?) or actuation (!)). In [10], an Industry 4.0 process modeling language (I4PML) based on BPMN 2.0 is presented. I4PML comprises the following elements: (i) cloud app, (ii) IoT device, (iii) device data, (iv) actuation task, (v) sensing task, (vi) human computer interface, and (vii) mobility aspect. uBPMN [11], in turn, introduces additional elements for camera, collector, sensor, and microphone. Each of these elements is represented by its own task and event types.

The existing approaches enable modeling IoT aspects in BPMN. However, the approaches are either too specific or cannot fully represent the behavior of IoT devices. For example, none of the approaches enables the modeling of an IoT boundary event. Another scenario that cannot be modeled with existing approaches concerns the verification of an IoT condition when processing a task.

None of the described approaches allows modeling IoT-aware processes at different levels of abstraction as our approach does (cf. Sect. 4). Note that different abstraction levels enable different process views for the various stakeholders (e.g., process expert, IoT expert, or domain expert). Moreover, existing approaches do not support the execution and monitoring of the elements they introduced. Table 1 summarizes the approaches (with \times expressing missing support and \checkmark indicating support of the respective feature). As can be easily seen, none of the existing approaches encompasses IoT-enabled processes with the necessary treatment.

Table 1. Comparison of related work.

	Sensor	Actuator	Combining sensor and actuator	Start event	End event	React to IoT within a task	Intermediate event	Condition element	Physical entity	IoT data object	Abstraction level	Multiinstance	Execution engine	Score
Cheng et al. [7]	\checkmark	\times	\times	\times	\times	\times	\times	\times	\times	\times	\times	\times	\times	1/12
Meyer et al. [8]	\checkmark	\checkmark	\times	\times	\times	\times	\times	\times	\checkmark	\times	\times	\times	\times	3/12
Sungur et al. [9]	\checkmark	\checkmark	\times	\times	\times	\times	\times	\times	\checkmark	\times	\times	\times	\times	3/12
I4PML [10]	\checkmark	\checkmark	\times	\times	\times	\times	\times	\times	\checkmark	\checkmark	\times	\times	\times	4/12
uBPMN [11]	\checkmark	\times	\times	\checkmark	\times	\times	\checkmark	\checkmark	\times	\checkmark	\times	\times	\times	5/12
This work	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\times	\checkmark	11/12

7 Conclusions and Outlook

This paper presented a BPMN 2.0 extension to enable the modeling, execution, and monitoring of IoT-aware business processes. Taking the given problem statement as well as the insights we gained from our literature review, we were able to identify fundamental challenges regarding the modeling, execution, and monitoring of IoT-aware business processes. We extended BPMN 2.0 with IoT artifacts and events that address the identified gaps. The added elements allow collecting, capturing, and exchanging data about the physical world over the Internet with the sensor artifact as well as controlling actuators with the actuator artifact. In addition, IoT conditions may be validated during task processing by the IoT intermediate catch artifacts as well as along the sequence flow with the IoT intermediate events. Furthermore, process start may be triggered by an IoT condition associated with an IoT start event and a process end may execute an actuator with the IoT end event. Finally, the IoT boundary event allows redirecting the sequence flow based on an IoT condition. In addition to the BPMN 2.0 IoT-related extensions, we presented an architecture to execute and monitor the modeled IoT-aware processes. The architecture allows for the execution and monitoring of IoT-aware processes in real time. Moreover, we applied our approach to a smart factory case to demonstrate that the framework is beneficial for modeling, executing, and monitoring IoT-aware business processes.

In future work, we want to make our framework multi-instance capable. In addition, we would like to generate IoT-enhanced logs and exploit them for an advanced process support treatment (e.g. to discover deviation between digital processes and their counterparts in the physical world or to improve process analytics). Moreover, we will conduct additional case studies of different domains to validate the domain independence.

References

1. Chang, C., Srirama, S.N., Buyya, R.: Mobile cloud business process management systems for the internet of things: a survey. *ACM Comput. Surv.* **49**, 1–42 (2016)
2. Ashton, K.: That ‘internet of things’ thing. *RFID J.* **22**, 97–114 (2009)
3. Kirikkayis, Y., Gallik, F., Reichert, M.: Towards a comprehensive BPMN extension for modeling IoT-aware processes in business process models. In: Guizzardi, R., Ralyté, J., Franch, X. (eds.) *RCIS 2022. LNBIP*, vol. 446, pp. 711–718. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05760-1_47
4. Valderas, P., Torres, V., Serral, E.: Modelling and executing IoT-enhanced business processes through BPMN and microservices. *J. Syst. Softw.* **184**, 111139 (2022)
5. Janiesch, C., et al.: The internet of things meets business process management: a manifesto. *Syst. Man Cybern. Mag.* **6**, 34–44 (2020)
6. Hasić, F., Serral, E.: Executing IoT processes in BPMN 2.0: current support and remaining challenges. In: *RCIS (2019)*
7. Cheng, Y., et al.: Modeling and deploying iot-aware business process applications in sensor networks (2019)
8. Meyer, S., Ruppe, A., Hilty, L.: The things of the internet of things in BPMN. In: *Conference in Advanced Information Systems Engineering Workshops (2015)*
9. Sungur, C.T., et al.: Extending BPMN for wireless sensor networks. In: *Business Informatics (2013)*
10. Petrasch, R., Hentschke, R.: Process modeling for industry 4.0 applications towards an industry 4.0 process modeling language and method. In: *Computer Science and Software Engineering (2016)*
11. Alaaeddine, et al.: uBPMN: a BPMN extension for modeling ubiquitous business processes. *Inf. Softw. Technol.* **74**, 55–68 (2016)
12. Torres, V., Serral, E., Valderas, P., Pelechano, V., Grefen, V.: Modeling of IoT devices in business processes: a systematic mapping study. In: *CBI (2020)*
13. Marrella, A., Mecella, M., Sardina, S.: SmartPM: an adaptive process management system through situation calculus, IndiGolog, and classical planning. In: *Principles of Knowledge Representation and Reasoning (2014)*
14. Kirikkayis, Y., Gallik, F., Reichert, M.: Modeling, executing and monitoring IoT-driven business rules with BPMN and DMN: current support and challenges. In: Almeida, J.P.A., Karastoyanova, D., Guizzardi, G., Montali, M., Maggi, F.M., Fonseca, C.M. (eds.) *EDOC 2022. LNCS*, vol. 13585, pp. 111–127. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17604-3_7



vAMoS: eVent Abstraction via Motifs Search

Gemma Di Federico^(✉)  and Andrea Burattin 

Technical University of Denmark, Kgs. Lyngby, Denmark
gdf@dtu.dk

Abstract. Process mining analyzes events that are logged during the execution of a business process. The level of abstraction at which a process model is recorded is reflected in the level of granularity of the data in the event log. When process activities are recorded as sensors readings, typically, they are very fined-grained and therefore difficult to interpret. To increase the understandability of the process model, events need to be abstracted into higher-level activities. This paper proposes vAMoS, a trace-based approach for event abstraction, which focuses on the identification of motifs on the traces, allowing some level of flexibility. The objective is the identification of recurring motifs on the traces in the event log. The presented algorithm uses a distance function to deal with the variability in the execution of activities. The result is a set of readable and interpretable motifs.

Keywords: Event abstraction · Motifs search · Sensor data

1 Introduction

Process mining [1] analyzes events that are logged during the execution of a business process. These events contain information on the activity executed, the resources involved, the execution time, etc. Control-flow discovery is the task of generating a process model that describes the process executions as collected in the logged data. The degree of abstraction to which the process model is represented, is reflected in the level of granularity of the recorded data. When data in the event log are recorded in a too fine grained fashion, the process model becomes too complex and no longer interpretable, thus less effective in extracting knowledge regarding of the underlying process.

When Business Process Management is used with data coming from Internet of Things [6], assuming a one-to-one mapping between an event in the log and a sensor reading, typically leads to too-fine grained event logs. Therefore, there is a need to abstract events such that they are able to represent more meaningful higher-level activity that can be related to a process level. To accomplish this task, a certain degree of expertise and knowledge may be required to be able to properly configure abstraction mechanisms.





When the data under investigation refers to human activities, human behavior and processes, additional challenges must be faced. Human related processes, such as daily personal routines, are unstructured and characterized by variability [5]. Consequently, it is difficult to define the set of activities carried out and their execution order. Additionally, when human behavior is collected in form of sensors data, the two challenges coalesce. The result is that, on the one hand, activities are expressed in the form of too fine-grained sensors readings; on the other hand, there is complexity in aggregating the activities carried out and understand what these express.

This problem has received substantial interest in the literature. However, the solutions available in the literature do not address the two sides of the problem together. Our research aims at finding a better solution for this challenging problem, by leveraging solutions devised in different scientific disciplines, i.e., biology. The method we propose is called vAMoS and it aims at recognizing common sequences of activities that show a high-level of similarity among traces. Therefore, the research conducted aims to address the following two questions:

RQ1 Is the technique able to identify meaningful patterns that can be abstracted as higher-level activities?

RQ2 Is the technique able to abstract behaviors from an event log by aggregating patterns?

vAMoS focuses on the identification of patterns of behavior at the trace level, in the form of sub-traces which show high-level of commonalities. The algorithm produces the list of patterns that can be used to abstract the original event log. To exemplify the problem as well as the devised solution better, let's introduce an example referring to the morning routine of a person who lives in an environment equipped with sensors.

Example 1. Every morning the person gets up from the bed area (()), eats some food (()) and drinks a coffee (()). Rarely, after waking up, the person goes directly to the work (()). We can consider the following event log: $t_1 = \langle \langle \text{床}, \text{苹果}, \text{苹果}, \text{咖啡} \rangle \rangle$ $t_2 = \langle \langle \text{床}, \text{苹果}, \text{咖啡}, \text{咖啡} \rangle \rangle$ $t_3 = \langle \langle \text{床}, \text{工作}, \text{工作}, \text{工作}, \text{工作} \rangle \rangle$ The typical behavior explained before (getting up and having breakfast, either as coffee or eating something) is indeed reflected in the traces, even though there is no single sub-sequence capturing it properly.

The objective is therefore to recognize the series of activities that are common in most traces. Each recognized sequence must be expressive enough to be explainable in the form of a higher-level activity.

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

2 Related Work

The problem of event abstraction has been tackled by several papers presenting many techniques, ranging from supervised to unsupervised approaches [15].

The majority of event abstraction algorithms require a knowledge expert to take active role in the analysis. Experts' knowledge is crucial to obtain information on the groups of sensors involved in the execution of a specific activity, to identify the main activities, and for the evaluation and verification of the results. However, there are two drawbacks that limit the expert's participation in the recognition of activities. As previously introduced, human processes are typically flexible and variable, therefore it is challenging for the expert to define exact rules for their recognition, in particular considering the fine granularity of the data collected. Another drawback concerns the set of recognizable activities, limited to the one provided by the expert. Instead, a "bottom-up approach", where possible activities are elicited directly from the data, allows the discovery of activities that were not foreseen beforehand. Consequently, the abstraction identifies a larger set of activities and is more pertaining to the reality observed in the log. All things considered, experts' knowledge should only have a supportive role and should not be considered as the main driver of the abstraction. For these reasons researchers investigated alternative ways to initialize the abstraction. In the literature, techniques for events abstraction can be divided in model- or trace-based. The former aims to derive process models that represent activities, the latter focuses on the identification of activities at a trace-level.

Model-based approaches aim to derive a model for each recognized activity in the event log. Leotta et al. [8] propose a model-based approach that consider the log as a "complex trajectory", and their objective is to identify sub-trajectories that are portions of the log referring to the same action. The algorithm identifies very short paths, hence the granularity of the abstraction remains too fine. Mannhardt et al. [9] proposed a supervised event abstraction method that makes use of behavioral patterns. Given a set of activities pattern, they build an abstracted model. After that, alignment techniques [3] are used to map low-level events in the event log to activities in the abstraction model. The approach is extended by the use of Local Process Model [13] (LPM) to describe the set of patterns: each LPM represents a high-level activity. The set of LPMs is then filtered and used with the abstraction method proposed in the previous version. LPMs can suffer from the over-generalization problem: the same LPM could be matched by potentially infinite different sequences of low-level events. What is more, LPMs describe small patterns in the log, composed by only three to five transitions. Another model-based abstraction solution is proposed by Baier et al. [2], which exploits imperative process model to derive constraints, following the same line of reasoning as the approach above. The proposed technique requires as input a process model describing the behavior as a set of activities, as well as a low-level event log. The objective is to map activities in the model with events in the log. The approach extracts declarative constraints from both the model and the log to build matching constrains to reduce the number of possible mappings. The mapping is also reduced using natural language processing, with a matching based on activities' labels and external knowledge. It follows from the above that a rigid and well-defined idea of the expected model is required,

limiting the recognition to a known set of activities represented in the process model.

Trace-based approaches, on the other hand, focus on the identification of commonalities between traces. In [7], de Leoni et al. propose a technique that aims to divide traces into batch sessions. Each trace is seen as a sequence of sessions of events. The sessions are converted into vectors which abstract the behavior observed inside the sessions, and then are clustered. The centroids are used for naming the activities. Each session is abstracted as one high-level activity execution, and K-Means is one of the proposed techniques for the clustering step. The main drawback lies in the choice of the session threshold parameter used for trimming sessions: this could have potentially impactful effects further in the analysis since activities could be missed or wrongly identified. The use of clustering algorithms, such as K-Means, is common among abstraction approaches proposed in the literature. Van Eck et al. [14] addressed the challenges of mapping sensor measurements to human activities, and grouping activities into process instances. The approach starts with the segmentation of sensor data in windows to be labeled. Relevant features are calculated for each of them, and subsequently the segments are clustered using K-Means. At this point, a domain expert assigns labels to the clusters, and segments can be grouped together to create activities. This technique was applied on a process regarding only one smart product equipped with sensors, so the performances in a real environment have not been evaluated. A limitation of the approaches that make use of K-means concern the impacts caused by the choice of the number of clusters (K). What is more, when number of low-level activities is large, the algorithm generates sparse clustering points which compromises the quality of the clusters.

When broadening the scope of the work to other domains, similarities can be found in bioinformatics and with the problem of protein sequence classification. The authors in [11] apply the sequential K-Means algorithm to sequences of data. Their goal is to find clusters that represent proteins, but considering the accumulation of mutations. To bring their problem into this paper's domain, each sequence can be seen as a trace, and the proteins represent the activities to be recognized (including mutations, i.e. noise). The algorithm provides as output clusters, that are activities identified, in form of sequences of events.

Regardless of the progress in the area, the major problem of event abstraction persists. In particular, there is no predominant trend between model and trace-based approaches. Although several studies have indicated good performances, little attention has been given to the granularity of the data source. In fact, only a few works in the literature demonstrate to perform well with low-level sensors data, e.g. [10]. For this reason, in the rest of this paper we use this work as a means of comparison to our algorithm.

3 Solution

The approach proposed in this paper aims at finding sequences of recurring activities over the event log that, when identified on a low-level log (e.g., where

activities refer to sensor data), can be interpreted as higher-level activities. The contribution of this work is called vAMoS, a motif search algorithm able to deal with recurring and variable patterns, that focuses only on observed sequences. vAMoS is implemented as a Java application, and the code is freely available¹.

The origin of vAMoS can be traced to the qPMS algorithm [12], a quorum Planted Motif Search approach designed to identify patterns (called *motifs*) in biological sequences of proteins, such that these motifs occur in most sequences provided as input. The objective of vAMoS is finding motifs but providing a more complex and accurate mechanism for their identification (e.g., to accept slight variations in how the motif is actually observed). To establish parallelism with the concept of *alignment* between traces [3]: in alignments, given two traces it is possible to calculate their cost (i.e., the extent of their similarity); in vAMoS, given two traces and a maximum cost it is possible to identify common parts of the traces that are similar (up to the cost).

Intuitively, a motif is a recurring sequence of activities that can be recognized in the log. To identify motifs, vAMoS first constructs a set of *candidate* motifs which are then verified on the traces of the event log. The verification procedure checks whether the motif is observed, up to a certain dissimilarity, in a minimum number of traces. The length of the motif is decided by the user. The construction of the set of candidate motifs could involve the construction of all possible sequences of activities but this would be unfeasible, especially since most of the candidates would be completely irrelevant (being very different from any trace on the event log). However, since we are interested in all possible motifs, even those that are not perfectly matched in the log, it is not enough to just consider all possible sub-traces of the log. Therefore, we opted for a two-step approach: first, identify the sub-traces that define the “alphabet” of our candidate motifs, and then combine these sub-traces in order to populate our set of candidate motifs. With this approach, the alphabet would be based on actual observations coming from the log, but the way candidate motifs are constructed allows for previously unseen candidates.

The first step consists of constructing the alphabet of sub-traces. Since the objective is to consider only observed sequences, we have to derive all the sequences, inside the traces, which are of a given length defined by the user. Hence, for each trace in the log, all the sub-traces of the given length are stored (like a sliding window). A sub-trace is called *n-gram*, while the set of all the sub-traces identified is named *n-set*. Considering Example 1 and setting 2 as the n-grams length, given the traces t_1, t_2, t_3 , the n-set is composed by all the pairs observed in the log: $N(E_L, 2) = \{ \langle \text{blue}, \text{red} \rangle, \langle \text{red}, \text{red} \rangle, \langle \text{red}, \text{purple} \rangle, \langle \text{purple}, \text{purple} \rangle, \langle \text{blue}, \text{green} \rangle, \langle \text{green}, \text{green} \rangle \}$.

The second step consists of the construction of the actual set of candidate motifs, meaning that all the n-grams have to be combined in order to reach the desired motif length. To construct the set of candidate motifs, starting from the n-set, we concatenate all the elements in the n-set between each other, up to reach the desired motif length (under approximation, in case the motif length is not multiple of the n-gram length). The obtained set comprises all possible combina-

¹ The source code can be found at <https://dx.doi.org/10.5281/zenodo.6378497>.

tions. Continuing with Example 1, considering an n-gram length $l_n = 2$ and motif length $l_m = 4$, the set of candidate motifs is the product between each element of the n-set, $C(E_L, l_n, l_m) = \{\langle \text{🛏}, \text{🍎} \rangle, \langle \text{🛏}, \text{☕} \rangle, \langle \text{🛏}, \text{🍎}, \text{🍎} \rangle, \langle \text{🛏}, \text{🍎}, \text{☕} \rangle, \dots\}$. It becomes quickly clear that not all candidate motifs are meaningful.

Once the set of candidate motifs is available, the next step is the verification of each of its elements in order to establish whether it is a motif indeed. The verification requires the concepts of *cost*, *distance* and *quorum*. The objective is to check if there are enough traces containing a given motif, within a certain similarity. The concept of similarity, in the context of this paper, establishes whether two activities can be considered equivalent w.r.t. the motif they belong. The cost relation indicates the extent of the interchangeability of two activities. E.g., given two activities a_1, a_2 , if the $cost(a_1, a_2)$ is close to 0, then a_1 and a_2 are interchangeable (i.e., they are equivalent); if the value is close to 1 it means they are not. The *cost* relation is expected to be provided by the user of the approach, since it is heavily domain dependent. For example, if we are considering two completely unrelated activities, and we have enough knowledge to state that they cannot be performed in the same context, therefore they cannot be interchanged, we will define their cost as a value close to 1. On the opposite, if we have two activities that are commutable, we define their cost as zero. Considering the IoT setting, this functionality can be useful to define relations between sensors placed on the same object, or sensors that are always activated together.

Given the cost between pairs of activities, we can define the distance d between two traces as the sum of the costs for each pair of activities of the respective traces. The distance function is useful to quantify the variability of non-interchangeable activities in traces which, in turn, is relevant since the same motif may appear multiple times in the log, but each instantiation can have a slightly different configuration. Considering Example 1, it can be noticed that both t_1 and t_2 follows a similar sequence of activities, i.e., moving from the bed area, to the kitchen for having some food and coffee. However, the traces are not exactly equivalent. In particular, if we consider the cost relation $cost = \{(\text{🍎}, \text{☕}, 0.7), (\text{🍎}, \text{🚰}, 1), (\text{☕}, \text{🚰}, 1)\}$ then $dist(t_1, t_2) = 0.7$, $dist(t_1, t_3[1, 4]) = 3$, $dist(t_2, t_3[1, 4]) = 3$. Employing a maximum distance threshold d , we would deem t_1 and t_2 within distance; whereas t_1 and t_2 against the sub-trace $t_3[1, 4]$ are not.

As discussed, a motif should be very common in the log to be considered relevant. vAMoS leverages the notion of *quorum* which represents the minimum percentage of traces where the candidate motif should appear (within a certain distance) to be considered relevant. Intuitively, from Example 1, considering the candidate motif $m = \langle \text{🛏}, \text{🍎}, \text{☕}, \text{☕} \rangle$, with max distance 1, a quorum 100% would not qualify m as a valid motif; whereas a quorum of 60% would. All things considered, we need the ability to check if a candidate motif is contained, within a certain distance d and *cost*, in trace t . The result of the evaluation is a so called motif or verified motif. To check if a candidate motif is a valid motif, we calculate the set of all traces for which at least one sub-trace of it (with the

same length as the motif) has a distance less than d against the motif. If there are enough traces (i.e., more than the quorum asks for), then the candidate motif is indeed a motif. The set of all verified candidate motifs is called *Set of motifs*. In particular, the procedure for the verification of a candidate motif starts by computing the number of traces where it appears by checking each possible sub-trace (with the same length as the candidate motif) of each trace in the log. If one sub-trace contains the candidate motif (i.e., its distance is less than a parameter), the respective trace counts as valid, and the number of traces that contains the motif under evaluation is incremented. If the number of traces containing the candidate motif satisfies the quorum, then the candidate motif becomes verified and hence it is returned as a motif. It is worth mentioning that while the set of motifs requires a very specific configuration both in terms of the motif and n-gram lengths, the motifs can be iteratively computed for any configuration and then combined together.

Once the motifs are identified, these can be abstracted. Each motif should be labeled with an activity name, and replaced in the event log. In order to replace a verified motif, it is enough to look for all instances of the motif in each trace (up to a given distance, considering a cost relation) and replace each of them with the abstracted activity. Considering one final time Example 1, we can compute the motifs that fulfill the requirements considering $l_n = 2$, $l_m = 4$, $d = 0.7$, $q = 60\%$, and $cost = \{(\langle \text{🍎}, \text{🍷} \rangle, 0.7), (\langle \text{🍎}, \text{🚰} \rangle, 1), (\langle \text{🍷}, \text{🚰} \rangle, 1)\}$. The set of motifs identified is $M = \{(\langle \text{🍷}, \text{🍎}, \text{🍷}, \text{🍷} \rangle), (\langle \text{🍷}, \text{🍎}, \text{🍎}, \text{🍷} \rangle)\}$. These sequences of events could then be abstracted into the higher level activity of “getting ready in the morning”.

The motifs computed by vAMoS are saved in form of traces in an XES file. The abstraction requires the set of motifs identified and the original event log. The approach implemented also offers the possibility to abstract the event log, as introduced above, by replacing the motifs in the log with a label. The abstracted event log is returned as an XES file, suitable to be used in process mining analysis.

4 Evaluation

The objective of the evaluation is to verify whether the algorithm is able to fulfill and answer the research questions presented in Sect. 1. To do so, we conducted tests on real data, for a qualitative assessment. All the evaluations are made by comparing our solution against one other approach in the literature, that is the event abstraction using LPMs [10] (referred to as LPM).

4.1 Evaluation on Real Data

For the analysis of real data, we opted for a qualitative assessment of the two approaches: vAMoS and LPM. We select the CASAS dataset [4] for this task. The dataset contains sensor data collected in a smart apartment, where a single person is living. It consists of 95 000 events (generated starting from 39 sensors)

over 15 cases. The log is labeled with high-level activities, but no information is given on how and when the activities are performed. Therefore, we decided to not consider the actual labeling and just use the real raw data for our experiment. As the resident did not have guidelines to follow, it cannot be assumed that the behaviors were recurrent. To make the dataset homogeneous, we select only cases in which the same set of activities is performed. In addition, the recording interval is reduced keeping only events from 07:00 to 12:00. Sensors not necessary for the recognition of activities (e.g., thermometers and OFF values) were filtered out. The resulting log consists of 7 cases, 10 000 events, and 31 sensors involved.

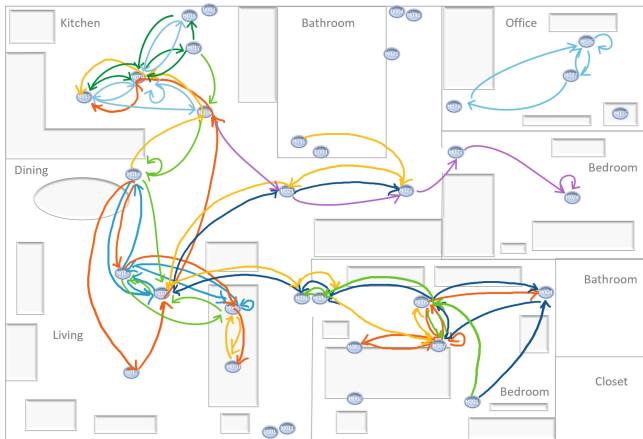
The event log is used as input for the two abstraction algorithms. An iterative approach was followed for vAMoS, starting by searching long motifs and then gradually reducing the length when the algorithm was no longer able to identify verified motifs. A total of 18 motifs are identified. On the other hand, several iterations were necessary to refine the parameter configuration of LPM. To give the algorithm the possibility to discover a wide range of results, LPM was set to produce 30 models, with five transitions and no duplicate activities. All the patterns are represented in form of process models. Two evaluations are conducted, one compares the patterns identified by both the algorithms, while the second evaluation focuses on the analysis of the abstracted event log obtained by the application of vAMoS.

The objective of the first evaluation is to compare the single patterns identified by the two approaches. The assessment is presented in Fig. 1 as a visual comparison between the models obtained by the application of the two approaches, projected on top of the floor map provided by CASAS. Even if the objective of LPM is to represent small patterns, between three and five activity nodes (i.e., transitions), the majority of the identified patterns are represented by the alternation between two sensors (see Fig. 1a), thereby these sequences are not very effective at recognizing meaningful higher level activities. On the contrary, results from vAMoS are shown in Fig. 1b, where the patterns are significantly longer and can be interpreted as activities. For instance, the path in the Office room comprises entering the room, moving steadily around the two sensors and then going out; another example (colored in lilac) illustrates the motif of moving from the Kitchen to the Bedroom.

The second evaluation is a comparison between the original log and the abstracted log obtained by the application of vAMoS. After the identification of the motifs, these were replaced in the original log to obtain a new abstract event log. The scarf-plot in Table 1 shows, for each case, a pair of bars: the bar on top represents the raw data from CASAS, the bottom bar presents the data after the processing with vAMoS. Each bar is composed of a series of colors, which indicate different activities (i.e. sensor triggers on top, sensor triggers and patterns on the bottom). Each sensor is associated with a different color, while the identified motifs are all represented by the yellow regions. The width of each colored bar is given by the duration of the activity. The last two columns in Table 1 refer to the number of events in the case replaced by a motif both in terms of duration and events abstracted. The case C5 is an example of the per-



(a) Patterns identified using LPM



(b) Patterns identified using vAMoS

Fig. 1. Floormap and patterns identified by the two approaches evaluated

formance of the algorithm. As can be noticed from the table, the 66% of the duration of the case is grouped due to a motif. In total, 55% of the duration of the original log is replaced with pattern labels. Concerning events, 45% of the total number of events in the log has been grouped in motifs. Following the same example of C5, 52% of events are abstracted. In case C4 instead, the percentages of the reduction are much lower, meaning that the motifs recognized in this trace are short and with low frequency. It is important to mention that we decided to stop the iterations of vAMoS after obtaining 25 patterns, otherwise we would have a percentage of the replaced log even higher.

Table 1. CASAS dataset before and after abstraction

Case	Scarf-plot	Dur	Evs
C1		55%	55%
C2		63%	44%
C3		60%	47%
C4		24%	21%
C5		66%	52%
C6		52%	51%
C7		60%	62%
Average:		54%	47%

4.2 Discussion and Limitations

The algorithm presented in this paper can be used to abstract activities recorded in fine-grained event logs, for example when events refer to sensors readings. The algorithm was compared to a model-based approach, LPM. The evaluation performed on the real dataset was used to verify whether the algorithm is able to recognize recurrent pattern, that are meaningful, in order to abstract an event log.

The first evaluation conducted has the objective to verify if the sequences identified are self-explainable or can be easily interpreted as activities, i.e., answering **RQ1**. To this end, the experiment showed that vAMoS was capable of identifying much longer motifs compared to LPM and was able to abstract a lot of behavior from low-level activities. The patterns identified by LPM are very short, therefore is challenging to recognize a meaningful behavior. Instead, the motif from vAMoS are clear in their structure, and can be intuitively matched with the performance of a specific activity. What is more, we observed that vAMoS was able to handle the variability. In fact, analyzing the traces in the event log in which every single motif was identified, we noticed that there was a strong similarity despite some discrepancies such as in the alternation of the order of two sensors.

Another key factor to focus on is the resulting event log, that is the abstract log. As indicated in **RQ2** we wanted to investigate if the algorithm was able to recognize behaviors and abstract them. The second evaluation conducted on the CASAS dataset shows the difference between the event log before and after the abstraction. The results of the abstraction are promising, since a high percentage of the event log is grouped into higher-level activities. Therefore, we can conclude that all research questions presented in Sect. 1 (i.e., **RQ1**, **RQ2**) can be positively answered.

Although the compelling results, the method proposed in this paper suffers from certain limitations. As introduced before, the procedures for calculating the set of candidate motifs as well as their verification are computationally intensive. During the application of the approach on a real dataset, the number of candidates (motifs for vAMoS, models for LPM) to be evaluated were remarkably high (i.e., millions of candidates). In this case, we made an assumption for the application of vAMoS, that is to keep the n-gram of the same length of the motif to reduce the number of candidates being verified each time. For this reason, it might be possible to introduce optimizations, yet these are beyond the scope of this work. For example, more optimized ways for constructing the set of candidates could be envisioned where only candidates with more likelihood of becoming verified are considered. It is important to highlight, however, that not only candidates perfectly seen in the log should be considered (though this can be done, by assigning the value of the n-gram length equal to the motif length), otherwise, there is the risk of missing some. Furthermore, the number of recognized motifs may be considerable, due to a non-optimized configuration of the parameters. In order for the algorithm to produce a limited number of motifs, which can be handled by an expert, some fine-tuning is required.

5 Conclusion and Future Work

In this paper, we presented a trace-based approach for event abstraction, named vAMoS. The fundamental idea of vAMoS, which leverages the qPMS algorithm, is based on the identification of recurring motifs that can be observed in a given percentage of the traces of the event log, up to a certain dissimilarity. The novelty introduced in the paper lies in the generalization of the qPMS algorithm (e.g., alphabet, candidates, distance, costs) as well as in the application scenario. The algorithm uses n-grams, as small observed sequences, that are combined for the construction of the set of candidate motifs. The motifs are evaluated based on a distance function that makes use of a cost relation. The algorithm is implemented as a Java application and it has been tested on a real dataset, in comparison with the state of the art.

In future work, we plan to investigate further improvements to the algorithm, in particular connected to the limitations introduced above. For example, having a more efficient identification of candidate motifs, where only viable candidates are considered. The labeling of the abstractions also represents an important future endeavor of our work.

References

1. Van der Aalst, W.M.: Process Mining: Data Science in Action. Springer, Cham (2016)
2. Baier, T., Di Ciccio, C., Mendling, J., Weske, M.: Matching events and activities by integrating behavioral aspects and label analysis. *SoSyM* **17**(2), 573–598 (2018). <https://doi.org/10.1007/s10270-017-0603-z>

3. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: *Conformance Checking*. Springer, Cham (2018)
4. Cook, D.J.: Learning setting-generalized activity models for smart spaces. *IEEE Intell. Syst.* **2010**(99), 1 (2010)
5. Di Federico, G., Burattin, A., Montali, M.: Human behavior as a process model: which language to use? In: *IT-BPM*, pp. 18–25. *CEUR-WS* (2021)
6. Janiesch, C., et al.: The internet of things meets business process management: a manifesto. *IEEE Syst. Man Cybern. Mag.* **6**(4), 34–44 (2020)
7. de Leoni, M., Dündar, S.: Event-log abstraction using batch session identification and clustering. In: *Proceedings of the ACM SAC*, pp. 36–44 (2020)
8. Leotta, F., Mecella, M., Sora, D.: Visual process maps: a visualization tool for discovering habits in smart homes. *J. Ambient Intell. Humanized Comput.* **11**(5), 1997–2025 (2020)
9. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: From low-level events to activities - a pattern-based approach. In: La Rosa, M., Loos, P., Pastor, O. (eds.) *BPM 2016*. LNCS, vol. 9850, pp. 125–141. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_8
10. Mannhardt, F., Tax, N.: Unsupervised event abstraction using pattern abstraction and local process models (2017)
11. Melman, P., Roshan, U.W.: K-means-based feature learning for protein sequence classification. In: *Proceedings of BICOB* (2018)
12. Nicolae, M., Rajasekaran, S.: qPMS9: an efficient algorithm for quorum planted motif search. *Sci. Rep.* **5**(1), 1–8 (2015)
13. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.: Mining local process models. *J. Innov. Digital Ecosyst.* **3**(2), 183–196 (2016)
14. Van Eck, M.L., Sidorova, N., Van der Aalst, W.M.: Enabling process mining on sensor data from smart products. In: *Proceedings of RCIS*, pp. 1–12. *IEEE* (2016)
15. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. *Granul. Comput.* **6**(3), 719–736 (2021). <https://doi.org/10.1007/s41066-020-00226-2>

**18th International Workshop on
Business Process Intelligence (BPI 2022)**

18th International Workshop on Business Process Intelligence (BPI)

Business Process Intelligence (BPI) is a growing area both in industry and academia. BPI refers to the application of data- and process-mining techniques to the field of Business Process Management. In practice, BPI is embodied in tools for managing process execution by offering several features such as analysis, prediction, monitoring, control, and optimization.

The main goal of this workshop is to promote the use and development of new techniques to support the analysis of business processes based on run-time data about the past executions of such processes. The workshop aims at discussing the current state of research and sharing practical experiences, exchanging ideas and setting up future research directions that better respond to real needs. We aim to bring together practitioners and researchers from different communities such as business process management, information systems, business administration, software engineering, artificial intelligence, process mining, and data mining who share an interest in the analysis of business processes and process-aware information systems. In a nutshell, it serves as a forum for shaping the BPI area.

The 18th edition of this workshop attracted 6 international submissions. Each paper was reviewed by at least three members of the Program Committee. From these submissions, the top 2 were accepted as full papers for presentation at the workshop, which was held in Münster, Germany. The workshop program was aligned with the AI4BPM workshop and kicked-off with a shared keynote by Prof. Marco Montali on the topic of “Constraints for Process Framing in Augmented Business Process Management”. In the talk, an interesting perspective was provided on how automated reasoning and declarative process management techniques can drive future innovations in the field. The two regular papers presented at the workshop provided a mix of novel research ideas, as described below.

First, *Aamer, Montali and Van den Bussche* focus on instance-spanning constraints, i.e. constraints that cover multiple execution instances of a business process. In particular, the authors investigate the application of database query processing technology to efficiently monitor constraints of interest. By translating an instance-spanning constraint to an ensemble of four database queries, it is shown how incremental view maintenance can be deployed, in this particular case using the DBToaster software in a proof-of-concept implementation.

Second, with the paper by *Kourani, van Zelst, Di Francescomarino, Ghidini, and van der Aalst*, the workshop also addressed the second main area of BPI, i.e. process discovery, in addition to conformance checking, which was covered by the first paper. In the second paper, a novel approach for detecting and discovering long-term dependencies in hybrid discovered process models is proposed. The technique relies on graph metrics applied to causal graphs. More specifically, the first step of the Hybrid Miner algorithm is refined by including long-term dependency edges to a discovered causal graph, relying on seven specific conditions that should be fulfilled. The

technique was implemented in ProM and tested quantitatively in terms of computational runtime and discovered process model quality.

As with previous editions of the workshop, we hope that the reader will find this selection of papers useful to keep track of the latest advances in the BPI area. We are looking forward to keeping bringing new advances in future editions of the BPI workshop.

October 2022

Organizaiton

Workshop Chairs

Andrea Burattin	Technical University of Denmark, Denmark
Jochen De Weerd	KU Leuven, Belgium
Marwan Hassani	Eindhoven Univ. of Technology, The Netherlands

Program Committee

Ahmed Awad	Cairo University, Egypt
Johannes De Smedt	KU Leuven, Belgium
Benoit Depaire	Universiteit Hasselt, Belgium
Claudio Di Ciccio	Vienna Univ. of Economics and Business, Austria
Chiara Di Francescomarino	Fondazione Bruno Kessler – IRST, Italy
Luciano García-Bañuelos	Tecnologico de Monterrey, Mexico
Gianluigi Greco	University of Calabria, Italy
Gert Janssenswillen	Universiteit Hasselt, Belgium
Anna Kalenkova	University of Melbourne, Australia
Sander Leemans	Queensland University of Technology, Australia
Michael Leyer	University of Rostock, Germany
Fabrizio Maggi	Free University of Bozen/Bolzano, Italy
Jorge Munoz-Gama	Pontificia Universidad Católica de Chile, Chile
Marco Pegoraro	RWTH Aachen University, Germany
Prina Soffer	University of Haifa, Israel
Boudewijn van Dongen	Eindhoven Univ. of Technology, The Netherlands
Seppe vanden Broucke	Ghent University, Belgium
Eric Verbeek	Eindhoven Univ. of Technology, The Netherlands
Matthias Weidlich	Humboldt-Universität zu Berlin, Germany
Hans Weigand	Tilburg University, The Netherlands
Sebastiaan Van Zelst	FIT/ RWTH Aachen University, Germany
Han van der Aa	University of Mannheim, Germany
Wil van der Aalst	RWTH Aachen University, Germany



Mining for Long-Term Dependencies in Causal Graphs

Humam Kourani¹(✉), Chiara Di Francescomarino², Chiara Ghidini²,
Wil van der Aalst³, and Sebastiaan van Zelst¹

¹ Fraunhofer FIT - Data Science and Artificial Intelligence, Sankt Augustin,
Germany

{humam.kourani,sebastiaan.van.zelst}@fit.fraunhofer.de

² Fondazione Bruno Kessler - Process and Data Intelligence, Trento, Italy

{dfmchiara,ghidini}@fbk.eu

³ RWTH Aachen University - Process and Data Science, Aachen, Germany
wvdaalst@pads.rwth-aachen.de

Abstract. Process discovery is one of the most challenging tasks in process mining. Based on event data, a process discovery approach generates a process model that captures the behavior recorded in the data. The hybrid miner is a two-step process discovery approach that creates a balance between the advantages of formal modeling and the necessity of remaining informal for vague structures. In the first discovery step, an informal causal graph is constructed based on direct succession dependencies between activities. In the second discovery step, the hybrid miner tries to convert the discovered dependencies into formal constraints. For vague structures where formal constraints cannot be justified, dependencies are depicted informally. In this paper, we reduce the representational bias of the hybrid miner by exploiting causal graph metrics to mine for long-term dependencies. Our evaluation shows that the proposed approach leads to the discovery of more precise models.

Keywords: Causal graphs · Long-term dependencies · Hybrid miner

1 Introduction

Process mining is a family of techniques that can be applied to analyze and monitor systems based on the events they produce. *Process discovery* is one of the main branches of process mining. Process discovery techniques analyze event data, aiming at discovering a process model capturing the behavior recorded in the data; the resulting process model illustrates how process activities are related to each other [6]. Most existing process discovery techniques produce *formal models* that have executable semantics and are able to classify traces into *fitting* and *non-fitting*. Alternatively, most commercial process mining tools use *informal models* that illustrate causal dependencies between activities without providing executable semantics. Although formal models provide more powerful insights, commercial tools favor representing processes using informal models due

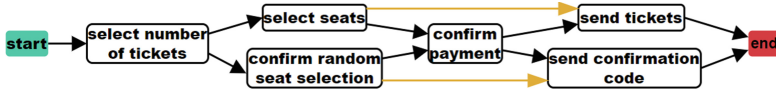


Fig. 1. Causal graph extended to include long-term dependencies. (Color figure online)

to multiple reasons. First of all, attempting to formally model complex structures results in complex models that cannot be easily interpreted by users. Moreover, for most real-life processes there is no clear correct classification of traces into fitting and non-fitting due to noise and infrequent behavior. Trying to precisely model all behavior seen in an event log can lead to overfitting process models that are not able to generalize well on unseen data. Finally, the discovery of formal models is very time-consuming compared to the discovery of informal models. Commercial process mining tools need to handle huge logs and interactively generate and update process models based on them.

The *hybrid miner* [7] combines the best of formal and informal modeling notations by discovering *hybrid Petri nets*. A hybrid Petri net shows some causal dependencies in an informal way similar to the models produced by commercial tools, and at the same time, it contains *places* that provide formal semantics for other parts of the process. The hybrid miner is a two-step discovery approach. In the first discovery step, causal metrics are computed based on direct succession dependencies between activities, and a causal graph is generated based on these metrics. A *causal graph* consists of nodes representing activities and two types of directed edges connecting nodes. *Certain edges* represent strong causal dependencies and *uncertain edges* represent weak dependencies. In the second discovery step, the hybrid miner converts the discovered certain edges into formal places if there is enough evidence in the data justifying adding formal constraints. For vague structures where formal constraints cannot be justified, causal relations are depicted in the final hybrid Petri net as informal edges. Since uncertain edges represent weak dependencies, they are not used for building places; they are added to the final Petri net as informal edges as well.

One of the main limitations of the hybrid miner is that it is not able to detect long-term dependencies. The hybrid miner constructs the causal graph based on direct succession dependencies between activities, preventing the discovery of simple structures with long-term dependencies. For instance, let us consider a simple process of booking concert tickets. After selecting the number of tickets, customers either select their seats (against additional fees) or confirm a random seat allocation. Afterward, customers confirm their order by paying. Finally, based on the earlier decision, customers either directly receive their tickets with assigned seat numbers via email or receive a confirmation code that they should use on the day of the concert to get their tickets. Figure 1 shows a causal graph discovered in the first step of the hybrid miner to model this process extended with two additional edges modeling long-term dependencies (visualized through yellow arcs). The causal graph shows dependencies between activities in an informal manner. The additional long-term dependency

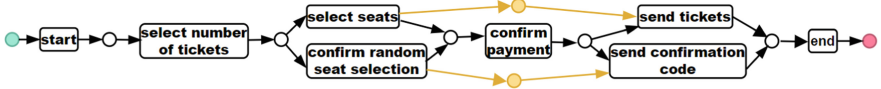


Fig. 2. (Hybrid) Petri net discovered based on Fig. 1. (Color figure online)

edges cannot be discovered by the hybrid miner because certain edges are constructed based on direct succession metrics (*select seats* is never directly followed by *send tickets* and *confirm random seat selection* is never directly followed by *send confirmation code*).

Certain edges are transformed into formal constraints in the second discovery step of the hybrid miner as shown in Fig. 2. The edges (*confirm payment* \rightarrow *send tickets*) and (*confirm payment* \rightarrow *send confirmation code*) are transformed into the place (visualized through a circle) connecting *confirm payment* with *send tickets* and *send confirmation code*. This place formally models a choice between sending tickets and confirmation codes after the payment. Since this choice depends on the earlier decision, two additional places are needed to capture this non-free-choice: the place connecting *select seats* with *send tickets* and the place connecting *confirm random seat selection* with *send confirmation code*. This behavior cannot be modeled precisely by the hybrid miner because these two places are generated based on the additional yellow edges we added to model long-term dependencies. Without these places, the model would allow, for instance, for behavior where customers select their seats and then receive a confirmation code for a random seat assignment.

In this paper, we propose an approach for extending the first discovery step of the hybrid miner to detect long-term dependencies. The proposed approach keeps using direct succession metrics for creating the initial causal graph, and it mines for long-term dependencies as a post-processing step. We define additional metrics based on eventually-follow relations between activities. We use these metrics to detect and filter long-term dependencies, and we add certain edges based on them. These certain edges are used in the second step for generating candidate places to formally capture long-term dependencies. This helps reduce the representational bias of the hybrid miner.

The remainder of the paper is structured as follows. We present some preliminaries in Sect. 2. In Sect. 3, we define an extended version of the causal graph miner that supports the detection of long-term dependencies. We evaluate our approach in Sect. 4, and we discuss related work in Sect. 5. Finally, we provide a short summary of the paper in Sect. 6.

2 Preliminaries

In this section, we address the preliminaries needed to understand the concepts we present in this paper.

2.1 Event Log

In the field of process mining, data is often represented in the form of *events*. The term event refers to the recording of an activity that happened during the execution of a process instance. Each event contains attributes providing information about the executed activity. The term *trace* refers to a sequence of events that represents the execution of a process instance. An *event log* is a collection of events that record the execution of multiple instances of a particular process. For simplicity, we abstract from these notations and define an event log as a multi-set of activity sequences.

Before providing any formal definitions, we introduce basic notations (based on [7]):

- $\mathcal{B}(X)$ denotes the set of all multi-sets over some set X . For example, $M = [x_1^2, x_2] \in \mathcal{B}(X)$ is a multi-set over $X = \{x_1, x_2, x_3\}$ with $|M| = 3$.
- X^* denotes the set of all sequences over some set X .
- For a sequence σ , $\sigma(i)$ denotes the i -th element of the sequence and $|\sigma|$ denotes the length of the sequence. For example, $\langle x_1, x_2, x_1 \rangle \in X^*$ is a sequence over $X = \{x_1, x_2, x_3\}$ with $\sigma(1) = \sigma(3) = x_1$, $\sigma(2) = x_2$, and $|\sigma| = 3$.

Definition 1 (Event Log [7]). Let \mathcal{A} be a set of activities. A trace $\sigma \in \mathcal{A}^*$ is a sequence of activities. An event log $L \in \mathcal{B}(\mathcal{A}^*)$ is a multi-set of traces.

$L_1 = [\langle \text{select number of tickets, select seats, confirm payment, send tickets} \rangle^{70}, \langle \text{select number of tickets, confirm random seat selection, confirm payment, send confirmation code} \rangle^{30}]$ is an example of an event log that contains 100 traces and 400 events.

2.2 Causal Graph

A causal graph is a directed graph with nodes representing activities and edges representing causal relations between activities. There are two types of edges: *certain* and *uncertain*. The edge type is based on the strength of the causal relation between the two activities connected by the edge. Certain edges are used to represent *strong* causal relations; uncertain edges are used to represent *weak* relations. Note that different metrics can be used to determine the strength of causal relations.

Definition 2 (Causal Graph [7]). A causal graph is a triple $G = (\mathcal{A}, R_S, R_W)$ where \mathcal{A} is a set of activities, $R_S \subseteq \mathcal{A} \times \mathcal{A}$ is the set of certain edges, $R_W \subseteq \mathcal{A} \times \mathcal{A}$ is the set of uncertain edges, and $R_S \cap R_W = \emptyset$.

In all examples in this paper, we abstract from uncertain edges because they represent weak dependencies and they are not used for generating candidate places in the second discovery step; We only visualize certain edges. Figure 1 shows an example causal graph.

2.3 Direct Dependency Metrics

There are many possible approaches for constructing a causal graph based on an event log. The hybrid miner uses a variant of the approach used by the heuristic miner [6,15]. In order to construct a causal graph, we define a causality metric ($Caus_\alpha$) for evaluating causal relations between activities and adding edges accordingly. This metric is based on direct succession dependencies between activities while taking concurrency and loops into account as well. In Definition 3, we define the causality metric $Caus_\alpha$.

Definition 3 (Direct Dependency Metrics [7]). Let $L \in \mathcal{B}(\mathcal{A}^*)$ be an event log over a set of activities \mathcal{A} and let $\{a, b\} \subseteq \mathcal{A}$.

- $\#(a, b, L) = \sum_{\sigma \in L} |\{i \in \{1, \dots, |\sigma| - 1\} \mid \sigma(i) = a \wedge \sigma(i+1) = b\}|$ counts the number of times a is directly followed by b in L .
- $\#(a, *, L) = \sum_{\sigma \in L} |\{i \in \{1, \dots, |\sigma| - 1\} \mid \sigma(i) = a\}|$ counts the number of times a is directly followed by any activity in L .
- $\#(*, b, L) = \sum_{\sigma \in L} |\{i \in \{2, \dots, |\sigma|\} \mid \sigma(i) = b\}|$ counts the number of times b is directly preceded by any activity in L .
- $Rel1(a, b, L) = \frac{\#(a, b, L) + \#(a, b, L)}{\#(a, *, L) + \#(*, b, L)}$ evaluates the strength of the causal relation (a, b) relative to the split and join behavior of activities a and b .
- $Rel2(a, b, L) = \begin{cases} \frac{\#(a, b, L) - \#(b, a, L)}{\#(a, b, L) + \#(b, a, L) + 1} & \text{if } \#(a, b, L) - \#(b, a, L) > 0 \\ \frac{\#(a, b, L)}{\#(a, b, L) + 1} & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$ evaluates the strength of the causal relation (a, b) taking into account concurrency and loops.
- $Caus_\alpha(a, b, L) = \alpha \cdot Rel1(a, b, L) + (1 - \alpha) \cdot Rel2(a, b, L)$ is the weighted average of $Rel1(a, b, L)$ and $Rel2(a, b, L)$ where $\alpha \in [0, 1]$.

The metrics $Rel1$, $Rel2$, and $Caus_\alpha$ all produce values between 0 and 1. Low values indicate weak dependencies; high values indicate strong dependencies. The variable $\alpha \in [0, 1]$ sets the weight of the relations $Rel1$ and $Rel2$ in $Caus_\alpha$. Let us consider our example log L_1 and the causal graph shown in Fig. 1. Both long-term dependency edges (yellow arcs) achieve a $Caus_\alpha$ score of 0 regardless of the value of α . Therefore, the hybrid miner is not able to discover these edges.

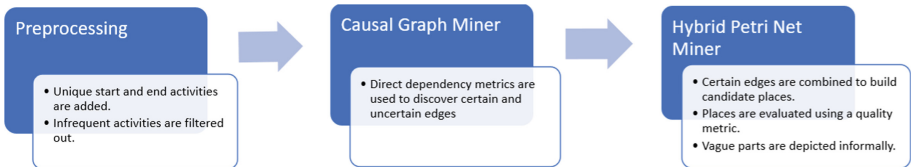


Fig. 3. Three steps of the hybrid Miner.

2.4 Hybrid Miner

The hybrid miner is a two-step process discovery approach that constructs a hybrid Petri net based on an input event log. The hybrid miner works according to the three steps shown in Fig. 3.

First, the hybrid miner preprocesses the log by adding a unique start activity “start” and a unique end activity “end” to all traces in order to help make the discovered models more understandable for users¹. Moreover, the user sets a filtering threshold, and infrequent activities are filtered out based on it. In the first discovery step, a causal graph is discovered based on the causality metric $Caus_\alpha$ defined in Definition 3. A parameter t_{R_S} is used for setting a threshold for certain edges. Similarly, another parameter t_{R_W} is used for uncertain edges². An additional parameter is used to set the value of the weight α . In Sect. 3, we will formally define an extended version of the causal graph miner that detects long-term dependencies.

In the second discovery step, the hybrid miner transforms the discovered causal graph into a hybrid Petri net. Candidate places are generated by combining certain edges, and they are evaluated using a quality metric. A parameter t_{eval} is used for setting a threshold for the quality metric. Based on this evaluation, some places are accepted and added to the hybrid Petri net while other places are rejected. At the end of the second discovery step, uncertain edges as well as certain edges that are not covered by any accepted places are added to the hybrid Petri net as informal edges. As the second discovery step is beyond the scope of this paper, we refer to [7] for more details on hybrid Petri nets and the second discovery step of the hybrid miner.

3 Detection of Long-Term Dependencies

In this section, we introduce an extended version of the causal graph miner; we extend the causal graph miner to detect and filter long-term dependencies.

3.1 Quantifying Long-Term Dependencies

The causal graph miner constructs causal edges based on direct succession relations between activities. In order to detect long-term dependencies, we also consider *eventually-follow relations*. In Definition 4, we define *outgoing and incoming dependency metrics* based on both direct succession relations and eventually-follow relations between activities. For a pair of activities (a, b) , the *Outgoing Direct Dependency* $ODD(a, b, L)$ is an estimation of the probability of executing b directly after executing a . The *Incoming Direct Dependency* $IDD(a, b, L)$ is an

¹ In all examples in this paper, we assume that all traces in any event log start with the activity “start” and end with the activity “end” without explicitly mentioning them.

² For all examples and experiments in this paper, we use $t_{R_W} = 1$ to deactivate the detection of uncertain edges because these edges are out of the scope of this paper.

estimation of the probability of executing a directly before executing b . The *Outgoing Long-term Dependency* $OLD(a, b, L)$ is an estimation of the probability of eventually executing b after executing a . The *Incoming Long-term Dependency* $ILD(a, b, L)$ is an estimation of the probability of eventually executing a before executing b .

Definition 4 (Outgoing and Incoming Dependency Metrics) . Let $L \in \mathcal{B}(\mathcal{A}^*)$ be an event log over a set of activities \mathcal{A} and $\{a, b\} \subseteq \mathcal{A}$.

- $ODD(a, b, L) = \frac{\#(a, b, L)}{\#(a, *, L)}$ is the outgoing direct dependency score of (a, b) in L .
- $IDD(a, b, L) = \frac{\#(a, b, L)}{\#(*, b, L)}$ is the incoming direct dependency score of (a, b) in L .
- $\widehat{\#}(a, L) = |\{\sigma \in L \mid \exists_{1 \leq i \leq |\sigma|} \sigma(i) = a\}|$ counts the number of traces in L where a occurs.
- $\widehat{\#}(a, b, L) = |\{\sigma \in L \mid \exists_{1 \leq i \leq j \leq |\sigma|} \sigma(i) = a \wedge \sigma(j) = b\}|$ counts the number of traces in L where a is eventually followed by b .
- $OLD(a, b, L) = \frac{\widehat{\#}(a, b, L)}{\widehat{\#}(a, L)}$ is the outgoing long-term dependency score of (a, b) in L .
- $ILD(a, b, L) = \frac{\widehat{\#}(a, b, L)}{\widehat{\#}(b, L)}$ is the incoming long-term dependency score of (a, b) in L .

Note that the dependency metrics ODD , IDD , OLD , and ILD all produce values between 0 and 1. Low values indicate weak dependencies; high values indicate strong dependencies. For a weight $\alpha \in [0, 1]$, we define the *Long-Term Dependency* LD_α as the average of OLD and ILD while taking into account concurrency and loops as well. Using higher values for the weight α means placing more emphasis on the long-term split and join behavior of activities (i.e., it means focusing on the outgoing and incoming long-term dependency scores); using lower values indicates placing more emphasis on the detection of concurrency and loops.

Definition 5 (Long-Term Dependency) . Let $L \in \mathcal{B}(\mathcal{A}^*)$ be an event log over a set of activities \mathcal{A} , $\{a, b\} \subseteq \mathcal{A}$, and $\alpha \in [0, 1]$. We define the long-term dependency score (LD_α) of (a, b) in L as follows.

$$LD_\alpha(a, b, L) = \alpha \cdot ((OLD(a, b, L) + ILD(a, b, L)) / 2) + (1 - \alpha) \cdot \max\{0, \frac{\widehat{\#}(a, b, L) - \widehat{\#}(b, a, L)}{\widehat{\#}(a, b, L) + \widehat{\#}(b, a, L)}\}.$$

For the sake of simplicity, we define an event log $L_2 = [\langle a1, c, a2 \rangle, \langle b1, c, b2 \rangle]$ with two straightforward long-term dependencies $(a1, a2)$ and $(b1, b2)$, and we use this event log in the remainder of the section as our running example. For any $\alpha \in [0, 1]$, both relations $(a1, a2)$ and $(b1, b2)$ achieve a long-term dependency score (LD_α) of 1.

3.2 Pruning Long-Term Dependencies

Adding all discovered long-term dependencies to the causal graph abundantly increases the number of certain edges and, therefore, abundantly increases the number of candidate places. This has a huge impact on the time performance of the hybrid miner. We can use a threshold for filtering long-term dependencies based on the scores obtained by LD_α ; however, this is not sufficient. In our running example, LD_α achieves a score of 1 for the relations $(start, c)$, (c, end) , and $(start, end)$. These relations are clearly not the type of long-term dependencies we are interested in because they are implied by other dependencies. A hybrid Petri net modeling a free-choice between $a1$ and $b1$ after $start$ and another free-choice between $a2$ and $b2$ after c fixes the execution order of the activities $(start \rightarrow c \rightarrow end)$, and there is no need for additional places for modeling the long-term dependencies. Adding additional certain edges to the causal graph for such long-term dependencies leads to additional time costs without helping improve the quality of the final models. Therefore, we propose further reduction mechanisms to keep only “interesting” long-term dependencies.

We define an extended version of the causal graph miner that mines for long-term dependencies. Long-term dependency relations between activities are evaluated based on the scores obtained by LD_α , and they are then filtered based on the metrics defined in Definition 4. We use a parameter t_{LD} to set a minimum threshold for LD_α . For the weight α , we use the same parameter used to set the weight α for $Caus_\alpha$. We recommend using high values for t_{LD} in order to avoid obtaining “fake” long-term dependencies resulting from loops, noise, or concurrency. We are often interested in long-term dependency relations that achieve high scores for LD_α .

Definition 6 (Extended Causal Graph Miner) . *Let $L \in \mathcal{B}(\mathcal{A}^*)$ be an event log over a set of activities \mathcal{A} , $\alpha \in [0, 1]$, $t_{R_S} \in [0, 1]$, $t_{R_W} \in [0, 1]$, and $t_{LD} \in [0, 1]$. A causal graph $G = (\mathcal{A}, R_S, R_W, R_{LD})$ is discovered for L as follows:*

- $R_{DD} = \{(a, b) \in \mathcal{A} \times \mathcal{A} \mid Caus_\alpha(a, b, L) \geq t_{R_S}\}$ is the set of direct dependency certain edges.
- $R_{LD} = \{(a, b) \in \mathcal{A} \times \mathcal{A} \mid a \neq b \wedge (a, b) \notin R_{DD} \wedge LD_\alpha(a, b, L) \geq t_{LD} \wedge$
 $\forall x \in \mathcal{A} \setminus \{a, b\} OLD(a, b, L) > OLD(a, x, L) \cdot OLD(x, b, L) \wedge$
 $\forall x \in \mathcal{A} \setminus \{a, b\} ILD(a, b, L) > ILD(a, x, L) \cdot ILD(x, b, L) \wedge$
 $OLD(a, b, L) > \frac{\sum_{x \in \mathcal{A} \setminus \{a\}} ODD(a, x, L) \cdot OLD(x, b, L)}{\sum_{x \in \mathcal{A} \setminus \{a\}} ODD(a, x, L)} \wedge$
 $ILD(a, b, L) > \frac{\sum_{x \in \mathcal{A} \setminus \{b\}} ILD(a, x, L) \cdot IDD(x, b, L)}{\sum_{x \in \mathcal{A} \setminus \{b\}} IDD(x, b, L)} \}$
 the set of long-term dependency edges.
- $R_S = R_{DD} \cup R_{LD}$ is the set of all certain edges.
- $R_W = \{(a, b) \in \mathcal{A} \times \mathcal{A} \mid Caus_\alpha(a, b, L) \geq t_{R_W} \wedge (a, b) \notin R_S\}$ is the set of uncertain edges.

3.3 Example Application

In this section, we apply the extended causal graph miner (Definition 6) to our running example in order to investigate the seven conditions used to filter long-term dependencies. Moreover, we apply the approach to another log that covers more advanced structures.

We apply the extended causal graph miner to a the event log $L_2 = [\langle a1, c, a2 \rangle, \langle b1, c, b2 \rangle]$ using the parameters $t_{RS} = t_{LD} = 0.3$ and $\alpha = 0.5$. The discovered causal graph is shown in Fig. 4a. The certain edges discovered based on long-term dependencies are visualized using yellow arcs. The causal graph is annotated with outgoing and incoming dependency scores; i.e., long-term dependency edges (R_{LD}) are annotated with *OLD* and *ILD* scores, and other certain edges (R_{DD}) are annotated with *ODD* and *IDD* scores. We use this example to explain the seven conditions that a causal relation must fulfill in order to be accepted as a long-term dependency (i.e., the seven filters used for constructing R_{LD} in Definition 6):

- Self-loops are filtered out. We accept a distance of 0 when computing the eventually-follow score (Definition 4); i.e., each activity is always eventually followed by itself, resulting in a self-loop.
- The second filter ensures avoiding duplicate certain edges. For example, no long-term dependency edge can be discovered for $(a1, c)$ because a certain edges is discovered for $(a1, c)$ based on direct succession dependencies.
- The parameter t_{LD} is used to set a minimum threshold for LD_α .
- The fourth and fifth filters are based on the idea of filtering out a long-term dependency relation if it is covered by other two relations. Outgoing long-term dependency scores are used in the fourth filter; incoming long-term dependency scores are used in the fifth filter. For example, the long-term dependency relation $(start, end)$ can be filtered out based on the fourth filter because $OLD(start, end, L_2) = 1 \not\geq 1 \cdot 1 = OLD(start, c, L_2) \cdot OLD(c, end, L_2)$. All three *OLD* scores have the value 1 because in all traces *start* is eventually followed by both *c* and *end*, and *c* is eventually followed by *end*.
- The last two filters exploit both direct succession dependencies and long-term dependencies. Outgoing dependency scores are used in the sixth filter; incoming dependency scores are used in the seventh filter. The idea is to filter out a long-term dependency if the relation can be covered by the direct neighbors of the source (sixth filter) or by the direct neighbors of the target (seventh filter). For example, the long-term dependency relation $(start, c)$ can be filtered out based on the sixth filter. *start* is directly followed by *a1* with a probability of 0.5 and is directly followed by *b1* with a probability of 0.5 (i.e., $ODD(start, a1, L_2) = ODD(start, b1, L_2) = 0.5$). We do not need to consider further nodes as *a1* and *b1* are the only direct neighbors of the source *start* (i.e., other nodes achieve an *ODD* score of 0). Both *a1* and *b1* are always eventually followed by *c* (i.e., $OLD(a1, c, L_2) = OLD(b1, c, L_2) = 1$). Therefore, we obtain: $OLD(start, c, L_2) = 1 \not\geq 1 = (0.5 \cdot 1 + 0.5 \cdot 1) / (0.5 + 0.5)$.

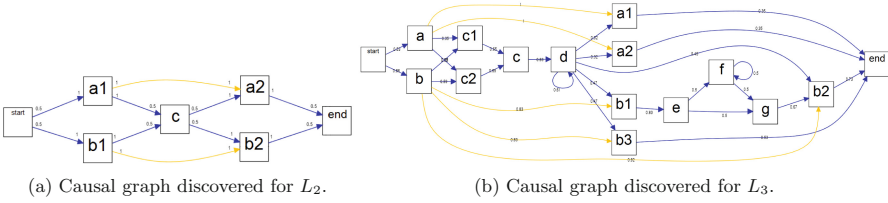


Fig. 4. Causal graphs discovered using the extended causal graph miner (Definition 6).

A long-term dependency edge was added to the causal graph to represent the causal relation $(a1, a2)$ because this relation fulfills all seven conditions:

- $a1 \neq a2$.
- $(a1, a2) \notin R_{DD}$.
- $LD_\alpha(a1, a2, L) = 1 \geq 0.3 = t_{LD}$.
- $OLD(a1, a2, L) = 1 > OLD(a1, x, L) \cdot OLD(x, a2, L)$ for all $x \in \mathcal{A} \setminus \{a1, a2\}$.
- $ILD(a1, a2, L) = 1 > ILD(a1, x, L) \cdot ILD(x, a2, L)$ for all $x \in \mathcal{A} \setminus \{a1, a2\}$.
- $OLD(a1, a2, L) = 1 > 0.5 = \frac{1 \cdot 0.5}{1} = \frac{ODD(a1, c, L) \cdot OLD(c, a2, L)}{ODD(a1, c, L)}$.
- $ILD(a1, a2, L) = 1 > 0.5 = \frac{0.5 \cdot 1}{1} = \frac{ILD(a1, c, L) \cdot IDD(c, a2, L)}{IDD(c, a2, L)}$.

In order to test our approach on more complex long-term dependencies, we extend our running example to include loop, choice, and concurrency structures. We define an event log L_3 that consists of the following traces: $\langle a, c1, c2, c, d, d, a1, a2 \rangle$, $\langle a, c2, c1, c, d, a2, a1 \rangle$, $\langle b, c2, c1, c, d, d, b3 \rangle$, $\langle b, c1, c2, c, d, b3 \rangle$, $\langle b, c2, c1, c, d, b2 \rangle$, $\langle b, c1, c2, c, d, b2 \rangle$, $\langle b, c1, c2, c, d, b1, e, f, f, g, b2 \rangle$, and $\langle b, c2, c1, c, d, d, d, b1, e, g, b2 \rangle$. More complex long-term dependencies can be identified in this log. The log contains a long-term concurrency relation between $a1$ and $a2$ after a . Moreover, there is a long-term choice between the activities $b2$ and $b3$ after b . In case $b2$ is selected, a sequence of activities starting with $b1$ can be executed before $b2$.

Figure 4b shows the causal graph discovered for L_3 using the parameters $t_{LD} = \alpha = 0.5$ and $t_{RS} = 0.3$. The long-term concurrency relation between $a1$ and $a2$ after a is captured by the long-term dependency edges $(a, a1)$ and $(a, a2)$. The long-term choice between the activities $b2$ and $b3$ after b is captured by the long-term dependency edges $(b, b2)$ and $(b, b3)$. An additional long-term dependency edge $(b, b1)$ is discovered for representing the case where an optional sequence of activities starting with $b1$ is executed after b .

Table 1. Time results of the evaluation of the parameter t_{LD} . We use $t_{0.5}$ (resp. $t_{0.7}$, $t_{0.9}$, and $t_{1.1}$) to denote the results obtained using $t_{LD} = 0.5$ (resp. 0.7, 0.9, and 1.1). We highlight high time values (higher than 5 min) in red and moderate time values (between 1 min and 5 min) in yellow.

Log	$ R_{LD} $				#Places				time (seconds)			
	$t_{0.5}$	$t_{0.7}$	$t_{0.9}$	$t_{1.1}$	$t_{0.5}$	$t_{0.7}$	$t_{0.9}$	$t_{1.1}$	$t_{0.5}$	$t_{0.7}$	$t_{0.9}$	$t_{1.1}$
BPI2011	6695	1378	37	0	123	126	100	68	1697.03	191.62	31.25	24.96
BPI2012	12	9	4	0	20	19	18	15	5.88	6.02	6.79	6.06
BPI2014	15	3	0	0	8	8	8	8	42.41	41.54	43.97	43.36
BPI2015	4508	2194	239	0	616	687	391	165	167.73	161.41	49.95	18.36
BPI2016	2767	575	17	0	22	25	22	10	188.23	150.41	39.63	9.62
BPI2017	10	3	0	0	20	20	20	20	12.20	8.02	8.15	8.14

4 Evaluation

In this section, we evaluate the extended causal graph miner based on real-life event logs³. We evaluate the effect of changing the value of the parameter t_{LD} on the time performance of the hybrid miner (Sect. 4.1) and the quality of the discovered models (Sect. 4.2).

4.1 Time Performance

In this section, we use the same six BPI Challenge data sets [2, 8–12] used to evaluate the initial version of the hybrid miner in [7]. We use the parameters $t_{Rs} = w = 0.5$. For the long-term dependency threshold (t_{LD}), we test the values 0.5, 0.7, 0.9, and 1.1 ($t_{LD} = 1.1$ means deactivating the detection of long-term dependencies). We use the plugin “Extended Hybrid Petri Net Miner” in ProM [13] to discover hybrid Petri nets based on the discovered causal graphs. The parameter $t_{eval} = 0.6$ is used in the second discovery step.

The results of the evaluation are shown in Table 1. For each case, we report the number of discovered long-term dependency edges in the first discovery step, the number of discovered places in the second discovery step, and the time needed to discover the models. The results show that, in general, decreasing the value of t_{LD} increases the time required to discover the models. For instance, decreasing the value of t_{LD} from 0.9 to 0.7 for BPI2016 increased the time from 39.63 s to 150.41 s. This behavior was expected because adding more long-term dependency edges leads to the generation of more places in the second discovery step, and the evaluation of these places is a time-consuming step. However, we observe that increasing the number of long-term dependency edges does not necessarily mean increasing the number of places in the final hybrid Petri net. For example, decreasing the value of t_{LD} from 0.7 to 0.5 for BPI2011 increased the number

³ A new plugin “Extended Causal Graph Miner” has been implemented in ProM [13] to support the approach introduced in this paper.

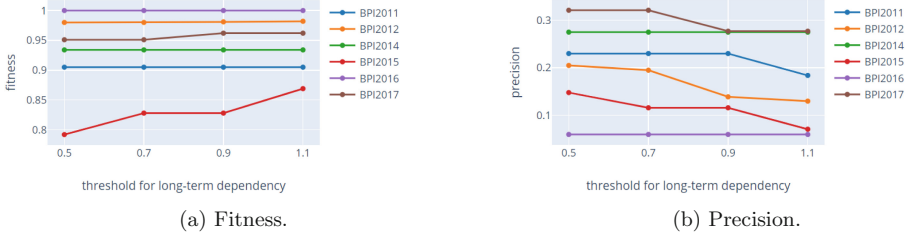


Fig. 5. Conformance checking results of the evaluation of the parameter t_{LD} .

of long-term dependency edges from 1378 to 6695, but it decreased the number of places from 126 to 123. This might be due to the replacement of a group of places by a larger place covering all of their underlying dependencies.

We observe that many long-term dependencies are not transformed into places in the second discovery step. This behavior was expected as the core idea of the hybrid miner is to create models with both formal and informal parts. For instance, let us consider the log BPI2014. Decreasing the value of t_{LD} from 0.9 to 0.5 generated 15 long-term dependency edges, but no places are added in the second discovery step based on them.

4.2 Qualitative Evaluation

We evaluated the quality of the discovered hybrid Petri nets in Sect. 4.1 by applying conformance checking techniques⁴ [1]. We were not able to get conformance checking results for many of the models due to out-of-memory exceptions. Therefore, we repeat the experiment using the same settings but after preprocessing the event logs by filtering them. For BPI2016, we filter out activities that are present in less than 50% of traces; for BPI2017, we filter out trace variants that cover less than 1% of traces; for the other logs (BPI2011, BPI2012, BPI2014, and BPI2015), we filter out trace variants that have an absolute frequency of 1.

We omit the time results of the second experiment because all models were discovered in less than a second. The conformance checking results are shown in Fig. 5. For BPI2014 and BPI2016, no places were discovered based on long-term dependencies, and the quality of the discovered models did not change, therefore. For the other four logs (BPI2011, BPI2012, BPI2015, and BPI2017), decreasing the value of t_{LD} improved the precision of the discovered models. For BPI2011, enabling the detection of long-term dependencies increased the precision from 0.184 to 0.230 without affecting the fitness. However, we observe a decrease in fitness after decreasing t_{LD} for other cases. For instance, decreasing the value of t_{LD} from 0.9 to 0.7 for BPI2017 decreased the fitness from 0.962 to 0.951 and increased the precision from 0.277 to 0.321. This behavior was expected because detecting long-term dependencies leads to the generation of additional places,

⁴ In order to apply conformance checking techniques, hybrid Petri nets are transformed into standard Petri nets by simply removing all informal arcs.

and adding more places often lowers the fitness of the model as more traces become non-fitting.

Summary. The time performance evaluation (Sect. 4.1) shows that the detection of long-term dependencies generates additional time costs. The qualitative evaluation (Sect. 4.2) shows that detecting long-term dependencies improves the precision of the models, but it can reduce the fitness as more places are discovered. We assess these differences in time and fitness to be acceptable as the goal of detecting long-term dependencies is to reduce the representational bias of the hybrid miner and produce more precise models. Based on our evaluation, we recommend using high values for the long-term dependency parameter (t_{LD}) in order to avoid high time costs and achieve a trade-off between fitness, precision, and simplicity.

5 Related Work

In [6], van der Aalst presented a wide range of process discovery approaches. The heuristic miner [6, 15] is a two-step discovery approach that produces an informal dependency graph in the first discovery step, and it uses this informal model to generate a formal causal net in the second discovery step. The causal graph created by the hybrid miner is inspired by the first discovery step of the heuristic miner. The idea of combining formal and informal models in process discovery and the notation of hybrid Petri nets were first introduced in [7]. The initial hybrid miner [7] was inspired by the idea of modeling vagueness suggested in [3, 4]. In [5, 17], other types of hybrid process models are defined by combined declarative and imperative modeling notations.

In this paper, we introduce an approach for improving causal graphs to detect and filter long-term dependencies. The discovery approach introduced in [14] also mines for long-term dependencies as a post-processing step. However, the approach in [14] is restricted to simple long-term dependencies as it assumes an equal frequency for any pair of activities of a long-term dependency relation. Our approach allows for the detection of more complex long-term dependency structures. In [16], another approach for detecting non-free-choice constructs is proposed. The main difference between this approach and our approach is that we mine for long-term dependencies in the first discovery step; we generate a causal graph that only contains the filtered set of long-term dependencies. In [16], redundant implicit dependencies are dynamically eliminated while generating the places of the final Petri net.

6 Conclusion

Process discovery is one of the main branches of process mining. Based on an event log, a process discovery approach generates a process model that captures the behavior recorded in the log. The hybrid miner is a two-step process discovery approach that combines formal Petri nets with an informal representation of

vague structures. In the first discovery step, an informal causal graph is discovered, and this graph is used to generate candidate places in the second discovery step. In this paper, we introduced an extended version of the first discovery step. The new causal graph discovery approach enables the detection of long-term dependencies. This helps to reduce the representational bias of the hybrid miner as the additional long-term dependencies are used to generate candidate places in the second discovery step. We implemented the extended version of the causal graph miner in ProM [13] and we evaluated it using real-life event logs.

We propose multiple ideas for future work. Although enabling the detection of long-term dependencies reduces the representational bias of the hybrid miner, it is still not able to model some simple structures. For instance, the hybrid miner is not able to model any structures that require using silent or duplicate transitions. Another important topic for future work is to investigate the time performance of the hybrid miner and to propose solutions for speeding up the generation and evaluation of candidate place in the second discovery step. We also suggest tailoring process conformance checking techniques to support hybrid Petri nets and defining new metrics for evaluating the quality of hybrid Petri nets. Finally, the idea of combining formal and informal models is not restricted to the discovery of hybrid Petri nets. This idea can be applied to other types of process models to discover new types of hybrid models.




References

1. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: *Conformance Checking - Relating Processes and Models*. Springer, Cham (2018)
2. Dees, M., van Dongen, B.: *BPI Challenge 2016: Clicks Logged In* (2016)
3. Herrmann, T., Hoffmann, M., Loser, K.U., Moysich, K.: Semistructured models are surprisingly useful for user-centered design. In: *Proceedings of the 4th International Conference on Designing Cooperative Systems*, pp. 159–174. IOS Press (2000)
4. Herrmann, T., Loser, K.-U.: Vagueness in models of socio-technical systems. *Behav. Inf. Technol.* **18**(5), 313–323 (1999)
5. Reijers, H.A., Slaats, T., Stahl, C.: Declarative modeling—an academic dream or the future for BPM? In: Daniel, F., Wang, J., Weber, B. (eds.) *BPM 2013*. LNCS, vol. 8094, pp. 307–322. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_26
6. Wil, M.P., van der Aalst, W.: *Process Mining - Data Science in Action*, 2nd edn. Springer, Cham (2016)
7. van der Aalst, W.M.P., De Masellis, R., Di Francescomarino, C., Ghidini, C.: Learning hybrid process models from events. In: Carmona, J., Engels, G., Kumar, A. (eds.) *BPM 2017*. LNCS, vol. 10445, pp. 59–76. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_4
8. Van Dongen, B.: *Real-life event logs - Hospital log* (2011)
9. Van Dongen, B.: *BPI Challenge 2012* (2012)
10. Van Dongen, B.: *BPI Challenge 2014* (2014)
11. Van Dongen, B.: *BPI Challenge 2015* (2015)
12. Van Dongen, B.: *BPI Challenge 2017* (2017)

13. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H.M.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The ProM framework: a new era in process mining tool support. In: Ciardo, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536, pp. 444–454. Springer, Heidelberg (2005). https://doi.org/10.1007/11494744_25
14. Weijters, A.J.M.M., Ribeiro, J.T.S.: Flexible heuristics miner (FHM). In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, part of the IEEE Symposium Series on Computational Intelligence 2011, pp. 310–317. IEEE (2011)
15. Weijters, A.J., Van der Aalst, W.M.: Rediscovering workflow models from event-based data using little thumb. *Integr. Comput. Aided Eng.* **10**(2), 151–162 (2003)
16. Wen, L., Van Der Aalst, W.M., Wang, J., Sun, J.: Mining process models with non-free-choice constructs. *Data Min. Knowl. Discov.* **15**(2), 145–180 (2007). <https://doi.org/10.1007/s10618-007-0065-y>
17. Westergaard, M., Slaats, T.: Mixing paradigms for more comprehensible models. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 283–290. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_24



What Can Database Query Processing Do for Instance-Spanning Constraints?

Heba Aamer¹ , Marco Montali² , and Jan Van den Bussche¹ 

¹ Hasselt University, Hasselt, Belgium

{heba.mohamed, jan.vandenbussche}@uhasselt.be

² Free University of Bozen-Bolzano, Bolzano, Italy
montali@inf.unibz.it

Abstract. In the last decade, the term *instance-spanning constraint* has been introduced in the process mining field to refer to constraints that span multiple process instances of one or several processes of particular relevance, in this setting, is checking whether process executions comply with constraints of interest, which at runtime calls for suitable monitoring techniques. Even though event data are often stored in some sort of database, there is a lack of database-oriented approaches to tackle compliance checking and monitoring of (instance-spanning) constraints. In this paper, we fill this gap by showing how well-established technology from database query processing can be effectively used for this purpose. We propose to define an instance-spanning constraint through an ensemble of *four database queries* that retrieve the satisfying, violating, satisfying-pending, and violating-pending cases of the constraint. In this context, the problem of compliance monitoring then becomes an application of techniques for incremental view maintenance, which is well-developed in database query processing. In this paper, we argue for our approach in detail, and, as a proof of concept, present an experimental validation using the DBToaster incremental database query engine.

Keywords: Compliance monitoring · SQL · Databases

Q: What's in a constraint?

A: Two (or four) database queries!

1 Introduction

Constraints that are posed over business processes can be very general and can refer to a variety of requirements [24]. Non-compliance of certain constraints can be very costly and risky, so compliance checking and monitoring are of utmost importance to the enterprise [34]. The following are various examples of such constraints. We will refer to these again in the remainder of this paper.

C1 “A *shipping car* can be used to deliver at most seven packages per day”

C2 “The time taken to deliver a package is between two and five days”

© Springer Nature Switzerland AG 2023

C. Cabanillas et al. (Eds.): BPM 2022, LNBIP 460, pp. 132–144, 2023.

https://doi.org/10.1007/978-3-031-25383-6_11

- C3 “*Conducting a patient’s surgery must be preceded by examining the patient*”
 C4 “*Packages that are delivered to the same neighbourhood on the same day must be delivered by the same shipping car*”

Constraints can be very simple in terms of their scope, i.e., the process instances they involve, and the conditions they impose such as C2 and C3. These are examples of constraints to be enforced on activity instances belonging to the same process instance. This type of constraint is often referred to as *intra-instance* [33, 34]. On the other hand, there are constraints that can be much more complex, both in their scope and in the conditions they impose. Specifically, constraints where the scope spans multiple process instances, or combinations of entities involved in multiple process instance, have been referred to as *inter-instance* [27, 33], or, more recently, *instance-spanning* constraints (ISCs) [14, 30]. C1 and C4 are examples of ISCs.

Although our focus is on studying ISCs, similar complex features would be required when checking intra-instance constraints on complex processes. In what follows, we will hence just talk about (process) *constraints* in general.

Constraints must be checked against execution logs, which are files or databases holding data about past and current executions of all process instances in the enterprise. *Post-mortem checking* and *compliance monitoring* are the two types of compliance checking that are commonly distinguished depending on the nature of the data used. The latter is more challenging since it checks the execution of the currently running process instances, for a live log.

There is a striking similarity between the problem of compliance monitoring and the problem of *incremental view maintenance*, a well-researched problem in databases [8, 16–18, 21, 22]. There, a *view* is the materialized result of a (possibly complex) query posed against a database. The problem of view maintenance is then to keep the view consistent with its definition under changes to the database. In general, these changes may be CRUD operations such as in particular insertions, deletions, or updates. This is perfectly in line with the execution of a process, where events witness the execution of tasks that, in turn, are typically associated to CRUD operations used to persist relevant event data in an underlying storage.

In this paper, we put forward the idea that incremental view maintenance is applicable to do compliance monitoring. We need to answer three questions: (1) What is the database? (2) What are the updates? (3) What is the query?

The first two questions are easily answered: the database is a representation of the contents of the log, and events trigger insertions to the log to leave a trace about their occurrence. In this context, only insertion operations are thus used, to append the occurrence of an event to those occurred before. Every insertion, triggered by the execution of some activity instance, stores the corresponding event data in the database, including the timestamp of the event and which data payload it carries.

What is then the query? To answer this question, we first need to indicate which dimensions we want to tackle when expressing constraints. Given the nature of ISCs, we want to comprehensively tackle multi-perspective constraints

dealing with several cases and their control-flow, time, and data dimensions. Instead of defining a specific constraint language that can accommodate such different perspectives, we directly employ full-fledged SQL for the purpose. Hence, a constraint is expressed as a query or, more precisely, an ensemble of queries, the number of which depends on whether compliance has to be assessed post-mortem or at runtime. In post-mortem checking, a constraint is expressed as a pair of queries ($Q_{\text{case}}, Q_{\text{viol}}$), which will be defined in Definition 1. Intuitively, Q_{case} defines the “scope” of the constraint, while Q_{viol} defines the violating subset of Q_{case} .

At runtime, we take inspiration from previous works in monitoring processes and temporal logic specifications [4, 10, 25], and consider that each constraint may be, in principle, in one of four possible states: currently satisfied (resp., currently violated), that is, satisfied (resp., violated) by the current event data, but with a possible evolution of the system that will lead to violation (resp., satisfaction); permanently satisfied (resp., permanently violated), that is, satisfied (resp., violated) by the current event data, and staying in that state no matter which further events will occur in the future. For well-studied languages only tackling the control-flow dimension, such as variants of linear temporal logics over finite traces, such states can all be automatically characterized starting from a single formula formalizing the constraint of interest [11]. This is not the case for richer languages tackling also the data dimension, as in this setting reasoning on future continuations is in general undecidable [6, 12]. We therefore opt for a pragmatic approach where constraint states are manually identified by the user through dedicated queries [7, 27]. In particular, a monitored constraint comes with an ensemble of four queries ($Q_{\text{case}}, Q_{\text{viol-perm}}, Q_{\text{viol-pend}}, Q_{\text{sat-pend}}$), which will be defined later in Definition 2.

To monitor constraints, we have used the system DBToaster for incremental query processing [21, 22, 32] in a proof-of-concept experiment. We monitor a number of realistic constraints on experimental data taken from the work by Winter et al. [34]. We demonstrate our approach in Sects. 2 and 3 of the paper.

Importantly, while we employ here the de-facto standard query language in databases, SQL, any other general data model (capable of suitably representing execution logs) with a sufficiently expressive declarative query language would do as well. Examples are the RDF data model with SPARQL, or graph databases with Cypher, which have been recently used in the context of object-centric process mining [13]. It should be noted, however, that incremental query processing is the most advanced for SQL. Indeed, relational database management systems are still the most mature database technology in development since the 1970s.

The paper is organized as follows. In Sect. 2, we formalize our approach, discuss examples of constraints and express them as SQL queries. In Sect. 3, we elaborate on the problem of compliance monitoring. In Sect. 4, we present the experimental results. In Sect. 5, we discuss query language extensions for sequences that can be useful as an approach. We conclude in Sect. 6. A full version of this paper is available, giving details on experiments, more examples, additional methodological remarks, and fully worked out SQL queries [3].

2 Post-mortem Analysis by Queries

Typically, post-mortem checking targets only full (completed) executions stored inside a historical log. We capture a constraint as a query that returns the set of cases incurring in a violation.

Definition 1 (Constraint, Post-mortem Variant). *A constraint C is a pair (Q_{case}, Q_{viol}) of queries where Q_{case} is a scoping query that returns all the cases subject to the constraint C , while Q_{viol} is a violation detection query that returns the violating cases such that Q_{viol} is always a subset of Q_{case} .*

This definition settles our approach for post-mortem checking. It is simply an application of query answering, where the queries are asked against a database instance (representing the execution log) that consists only of completed process instances. In that case, when a tuple $t \in Q_{case} \setminus Q_{viol}$, then t represents a case that satisfies the constraint (i.e., $t \in Q_{sat}$).

Remark 1. Note that an equivalent approach is to represent the constraint as the pair of queries (Q_{case}, Q_{sat}) instead. The two approaches are interchangeable; sometimes one is simpler to specify, sometimes the other.

Guaranteeing that, for a constraint (Q_{case}, Q_{viol}) , query Q_{viol} always returns a subset of Q_{case} is under the responsibility of the modeler. One way to ensure this is to write Q_{viol} as a query that takes Q_{case} and extends it with a filter to identify violations; however, alternative formulations may be preferred for readability and/or performance needs.

2.1 Database Schema

The structure of the database schema representing the data of the execution log, and how to get a database instance with the data, are important issues. In this paper, we assume these are available. A comprehensive treatment is given by de Murillas et al. [28]. For our purposes of giving illustrating examples, we will simply assume the following two relations in our database, in line with the XES standard extensions [19]:

- `Log(CaseId, EventId, ActivityLabel, Timestamp, Lifecycle)` which is the main relation.
- An auxiliary `EventData` relation that contains the extra information of the logged events. Events are identified by the combined key `(EventId, Lifecycle)`, and the extra attributes depend on the application.

2.2 Examples

In the following examples, we assume that the relation `EventData` has the schema `(EventId, Lifecycle, PackageId, CarId)`. We also assume that in our processes, we have two activities with the labels “purchase package” and “deliver package”.

Example 1 (Same Shipping Car Constraint). Consider constraint C1 from the Introduction. As we have mentioned before, we have a great flexibility in defining what a violation is (in other words, what is the scope of the constraint). One possibility is to define the cases to be tuples (CarId,Day,CountOfDeliveries). Following this view, the constraint can be represented by the following pair of queries (in favor of saving space, Q_{viol} is only shown, and Q_{case} is merely the same without the last condition in line 6):

```

1 SELECT e.CarId, DATE(1.Timestamp), COUNT(e.PackageId)
2 FROM Log 1, EventData e
3 WHERE 1.EventId=e.EventId AND 1.ActivityLabel='deliver package' AND
4       1.Lifecycle='complete' AND e.Lifecycle='complete'
5 GROUP BY e.CarId, DATE(1.Timestamp)
6 HAVING COUNT(e.PackageId) > 7;
```

A less fine-grained scope, having tuples (CarId,Day) as our cases, is also possible. In this case, the queries are similar to the ones discussed above, but dropping the third selected column.

Example 1 demonstrates possible queries that define an instance-spanning constraint. To show the uniformity of our approach, the following is an example of an intra-instance constraint.

Example 2 (Shipping Time Constraint). Consider now constraint C2. In what follows, we consider a case to be a package identifier. Again, here we only show Q_{viol} , and Q_{case} is exactly the same without the last condition in line 7.

```

1 SELECT e.PackageId FROM Log 11, Log 12, EventData e
2 WHERE 11.TraceId=12.TraceId AND 12.EventId=e.EventId AND
3       11.ActivityLabel='purchase package' AND
4       12.ActivityLabel='deliver package' AND
5       11.Lifecycle='complete' AND 12.Lifecycle='complete' AND
6       e.Lifecycle='complete' AND
7       DATE(12.Timestamp) - DATE(11.Timestamp) NOT BETWEEN 2 and 5;
```

3 Compliance Monitoring as Incremental View Maintenance

If we want to monitor a constraint *dynamically*, we have to refine our definition. The reason is that the database instance representing the execution log is continuously progressing. Thus, the database instance will contain the data of running (non-completed) process instances along with the completed process instances. Hence, at any moment, any case that is subject to some constraint will be in one of four different states [4, 10, 25]: 1) a *permanently* violating state; 2) a *permanently* satisfying state; 3) a *currently* violating state that may later be in a satisfying state as a result of the occurrences of new events; and 4) similarly, a *currently* satisfying state that may later be in a violating state. We will refer to the last two states as *pending* states. Note that the set of cases are constantly changing, so new cases can pop up, while others can simply cease to exist. Notice

that it depends on the constraint under study whether all such four states have to be actually considered, or whether instead the constraint only requires a subset thereof. Example 3 discusses a simple constraint such that we can have its cases belonging to the different states.

Regardless of the formal tools, languages, approaches, there is always a “methodology” to go from informal specifications to formal realization.

Example 3 (Monitoring “Followed-By” Constraint). Consider a process that comprises three activities with the labels A , B , and C . Consider the constraint “Every instance of activity A must be directly followed by an instance of activity B within 20h”. Each process instance is a case here. An instance where A is directly followed by an activity C , say, would be a violation. However, an instance where A is the last event for now, but the instance is not yet completed, is pending violating, as long as A was not longer than 20 h ago.

In general, we propose:

Definition 2 (Constraint, Compliance monitoring Variant). A constraint C consists of four queries (Q_{case} , $Q_{viol-perm}$, $Q_{viol-pend}$, $Q_{sat-pend}$), where Q_{case} returns all the cases subjected to the constraint C , and $Q_{viol-perm}$, $Q_{viol-pend}$, and $Q_{sat-pend}$ return the cases that are permanently violating, pending violating, and pending satisfying, respectively. On any database instance, $Q_{viol-perm}$, $Q_{viol-pend}$, and $Q_{sat-pend}$ always return three mutually exclusive subsets of Q_{case} .

Some of the four queries may be empty, witnessing that the corresponding monitoring state is not compatible with the constraint at hand. Also, the permanently satisfying cases can be derived from the four queries as $Q_{sat-perm} = Q_{case} - (Q_{viol-perm} \cup Q_{viol-pend} \cup Q_{sat-pend})$. Typically, the query Q_{viol} in the post-mortem checking variant corresponds to the union of $Q_{viol-perm}$ and $Q_{viol-pend}$ in the compliance monitoring variant. Similarly, the query Q_{sat} corresponds to the pair $Q_{sat-perm}$ and $Q_{sat-pend}$.

Example 4 (Monitoring Same Shipping Car Constraint). Consider again constraint C1. Queries Q_{case} and $Q_{viol-perm}$ are the same as Q_{case} and Q_{viol} of Example 1. Then for $Q_{sat-pend}$, we use the following query:

```

1 SELECT e.CarId, DATE(1.Timestamp), COUNT(e.PackageId)
2 FROM Log l, EventData e
3 WHERE l.EventId=e.EventId AND l.ActivityLabel='deliver package' AND
4       l.Lifecycle='complete' AND e.Lifecycle='complete' AND
5       DATE(1.Timestamp)=CURRENT_DATE
6 GROUP BY e.CarId, DATE(1.Timestamp)
7 HAVING COUNT(e.PackageId) <= 7;
```

The fourth query, $Q_{viol-pend}$, will always be empty for this constraint, because once a shipping car has been used too often, it is immediately a permanent violation, since the situation cannot be salvaged anymore.

4 Experiments

Once constraints are expressed as queries per our methodology, compliance monitoring becomes an application of view maintenance. DBToaster is a state-of-the-art incremental query processor [21, 22, 32]. As a proof-of-concept of our approach, we tested DBToaster on constraints from the work of Winter et al. on automatic discovery of ISCs [34]. We have also used the execution logs provided by these authors as sample input data [9]. To manage our experiments, we performed some preprocessing steps (detailed in the full version [3]).

The tested constraints are expressed over the three processes whose models are shown in Fig. 1. Any order has a corresponding initiated “Bill” process. Moreover, printers are considered a shared resource between all the processes.

We have specifically run tests on five constraints ISC1, ISC2a, ISC2b, ISC3, and ISC4 from Winter et al. Due to space limitations, here, we only present results on the following three constraints:

- ISC1 There is exactly one delivery activity per day in which all the finished orders/bills of that day so far are delivered to the post office simultaneously.
- ISC2a All print jobs must be completed within 10 min in at least 95% of all cases per month.
- ISC3 If a flyer or poster order is received $P2$ is started afterwards. Moreover, the corresponding bill process must be started before the order is delivered to the post office.

We slightly modified the original constraints [34] to better match with the log data [9]. Each constraint was expressed using SQL queries, according to Definition 2 [3].

Running Time. We measured the running time of the five monitored constraints, averaged over 10 runs. The time is measured after every 300 insertions for a total of 30636 insertions (the number of events in the dataset). This experiment was performed on a personal laptop running macOS 12.2.1 with RAM of 16 GB and processor speed of 2.6 Hz.

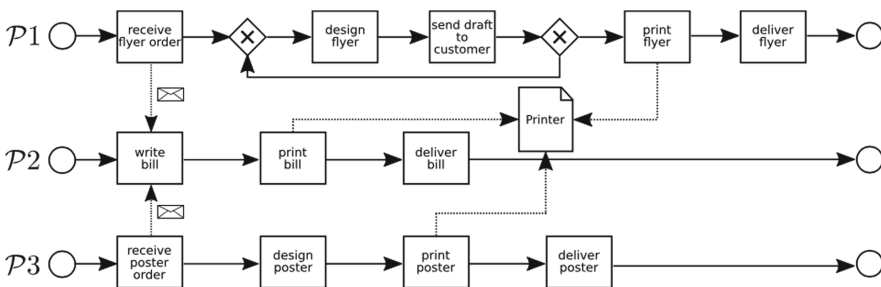


Fig. 1. Processes: flyer order, bill, and poster order [34].

The attentive reader may note that the line sometimes goes down. This is an artifact of the way in which we avoided a bug in the Scala version of DBToaster. We reran DBToaster on ever larger sequences of insertions; the queries the system uses internally to retrieve the current snapshot are sometimes slightly faster on slightly larger database instances. The outcome of this experiment produces the average time needed, per event, to maintain the queries defining the constraint. We can see that the slope is significantly higher for the first constraint; indeed, this constraint requires rather complex SQL queries. For ISC1, the maintenance time is less than half a millisecond; for ISC3, less than 1/6th of a millisecond; and for the other constraints less than 1% of a millisecond.

Queries Result Sizes. Figure 3 shows query result sizes (the number of cases) relative to time (number of insertions). This shows how cases are changing their status (pending or permanent, violating or satisfying). The query size is reported every 500 insertions in the case of ISC1, every 100 insertions for ISC2a, as it displays a more fine-grained behavior.

Tracing Cases. As an illustration of the feasibility of our approach and its compatibility with monitoring on a very detailed level, we show in Fig. 4 the evolution in status of some individual cases of ISC1 over time.

5 Sequence Data Extensions of Query Languages

We have mentioned before that any data model with a sufficiently expressive query language can be used to express the constraints. Although, we chose to work with the relational data model with SQL for the reasons we mentioned, it is interesting to briefly discuss query languages for the relational data model extended with sequences [23, 31]. Indeed, a trace is a sequence of events. Hence, representing the relative order of the events is quite natural in a sequence data model. This level of abstraction, of viewing traces as sequences of abstract events, is often assumed when working with temporal and dynamic logics [10, 15, 29].

Sequence Datalog [2, 5, 26] is an extension of the query language Datalog, to work with sequences as first class citizens. We will briefly showcase this language by considering an example of a constraint that is usually handled with temporal logic.

Example 5. (Strict Sequencing [15]). Let a and b be two activities. There is a *strict sequencing* relation [1] between a and b if the log satisfies the following:

- there exists a trace where a is immediately followed by b ; and
- there are not any traces where b is immediately followed by a .

There are two possible violations of this constraint: (1) not having a trace with b directly following a ; (2) having a trace with a directly following b .

For the purpose of expressing this constraint, assume we have the relation $\text{Log}(\text{TraceId}, \text{Events})$, where Events are sequences of labels of activities. Then, this constraint can be expressed by the following Sequence Datalog program.

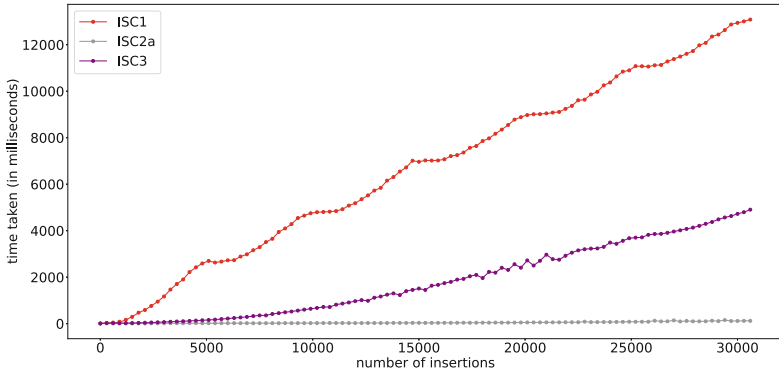


Fig. 2. Running time taken to monitor the constraints.

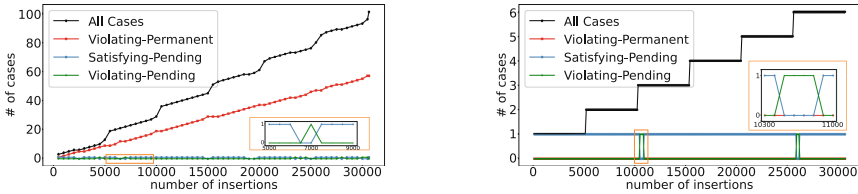


Fig. 3. Result sizes of queries for ISC1 and ISC2a. Since our measurements consist of 600 data points (even 3000 for ISC2a), the plots are at rather large scale. To show more detail, we provide insets that zoom in on selected regions (orange rectangles). (Color figure online)

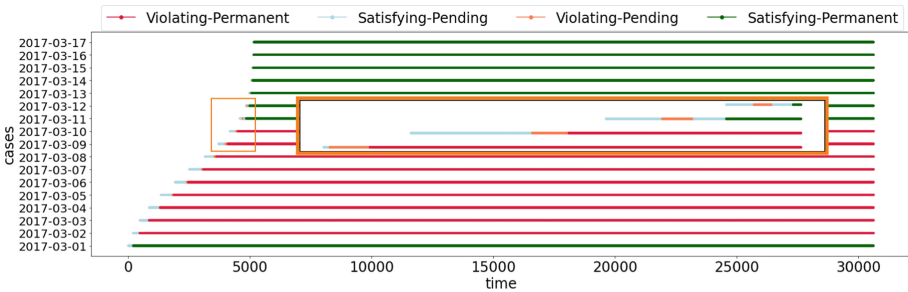


Fig. 4. Some ISC1 cases (days of March) and how each is changing its status through time. Here the measurement consists of 30600 data points per case, so the plot is at a very large scale. The inset shows more detail by zooming on the selected region (orange rectangle). (Color figure online)

```

a_before_b() ← Log(@traceId, $pre.a.b.$post).
violation() ← ¬a_before_b(). % violation (1)
violation() ← Log(@traceId, $pre.b.a.$post). % violation (2)

```

This program illustrates a number of Sequence Datalog features:

- the dot is the concatenation operator.
- @traceId is an *atomic* variable (indicated by the @ symbol) representing atomic values (in this case, trace identifiers).
- \$pre and \$post are *sequence* variables (indicated by the \$ symbol) representing (possibly empty) sequences of atomic values.

The utility of using Sequence Datalog can be appreciated if we compare the above program with the same query expressed in SQL which is more complicated and much longer [3]. This paves the way towards the adoption of Sequence Datalog to express, check and monitor constraints, once techniques for (incremental) query processing will be implemented.

6 Discussion

In this paper, we have looked into the problems of post-mortem checking and compliance monitoring of constraints over business processes. Specifically, we focused on ISCs as recently introduced in the process mining field, and caught attention since it refers to complex constraints that span multiple process instances. Although there have been extensive works on inventorying and categorizing ISCs [30, 34], a crisp definition of what is or is not an ISC, however, seems to be elusive. Indeed, the notion of constraint is so broad that we propose here to *define* any constraint as two or four queries posed against the database instance that represents a (partial) execution log. This approach gives us huge flexibility, and moreover, allows the use of incremental query processing techniques out of the box.

In using the DBToaster system for our experiments, we faced a few technical issues. The main challenge was that the Scala version of DBToaster gets stuck when retrieving snapshots over the course of the insertions. Another limitation is that SQL is not yet fully supported, although complex queries can be expressed. This required us to sometimes rewrite queries in equivalent form. Finally, some built-in functions (e.g., on strings or dates) are missing from the Scala version. Thus, our experiments should be seen more as a proof-of-concept of the feasibility of our approach.

In this discussion, we briefly touch upon the main difference between our approach and the main approach to monitoring ISCs, based on the Event Calculus (EC) [20, 24, 27]. Most monitoring systems based on EC are implemented using Prolog. Using EC to express a constraint seems to be very *procedural* albeit being defined in logical programming language. For example, to monitor a constraint such as C1, in EC one would define a rule that increments a counter every time a delivery event occurs. At the end, that counter value should be at most seven as per the constraint. A similar approach was followed in the paper by Montali

et al. [27] to monitor intra-instance constraints with EC. Events come in time, and Prolog rules that fire every new time instant, are used to check various constraints dynamically. However, these incremental rules are manually implemented. On the contrary, using an incremental query processor shifts the focus on what the queries (or constraints) themselves are rather than what the rules are that are responsible for this incremental maintenance. Hence, our approach is more declarative.

At the end of this discussion, we mention a few points for further research. Since there are some algorithms that are used to discover ISCs from execution logs [34], and these algorithms search for explicit patterns, one could define a common language to report the results of those algorithms and use those results to automatically write the SQL queries monitoring each of the reported constraints. Thus, the whole process could be automated. An engineering issue is to investigate which discipline of query formulation works best with incremental view maintenance methods. Another natural continuation of this work is to explore the possibility of monitoring object-centric processes where multiple interrelated objects are co-evolved. Our approach can be readily applied to this setting, considering that our techniques operate over a full-fledged relational database.

Acknowledgments. We thank Stefanie Rinderle-Ma and Jürgen Mangler for initial discussions. Heba Aamer is supported by the Special Research Fund (BOF) (BOF19OWB16). Marco Montali is supported by the PRIN Italian Project PINPOINT, and by the unibz ID project ADAPTERS.

References

1. van der Aalst, W.M.P., Weijters, T., et al.: Workflow mining: discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **16**(9), 1128–1142 (2004)
2. Aamer, H., Hidders, J., Paredaens, J., Van den Bussche, J.: Expressiveness within sequence datalog. In: *PODS* (2021)
3. Aamer, H., Montali, M., Van den Bussche, J.: What can database query processing do for instance-spanning constraints? [arXiv:2206.00140](https://arxiv.org/abs/2206.00140) (2022)
4. Bauer, A., Leucker, M., Schallhart, C.: Runtime verification for LTL and TLTL. *ACM Trans. Softw. Eng. Methodol.* **20**(4), 1–64 (2011)
5. Bonner, A., Mecca, G.: Sequences, datalog, and transducers. *J. Comput. Syst. Sci.* **57**, 234–259 (1998)
6. Calvanese, D., De Giacomo, G., Montali, M., Patrizi, F.: Verification and monitoring for first-order LTL with persistence-preserving quantification over finite and infinite traces. In: *IJCAI-ECAI* (2022)
7. Cardoso, E., Montali, M., Calvanese, D.: Representing and querying norm states using temporal ontology-based data access. In: *EDOC* (2019)
8. Chirkova, R., Yang, J.: Materialized views. *Found. Trends Databases* **4**(4), 295–405 (2012). <https://doi.org/10.1561/19000000020>
9. CRISP project at Universität Wien: Logs Webpage. <http://gruppe.wst.univie.ac.at/projects/crisp/index.php?t=discovery>. Accessed 29 Apr 2022

10. De Giacomo, G., De Masellis, R., Grasso, M., Maggi, F.M., Montali, M.: Monitoring business metaconstraints based on LTL and LDL for finite traces. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) BPM 2014. LNCS, vol. 8659, pp. 1–17. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10172-9_1
11. De Giacomo, G., De Masellis, R., Maggi, F.M., Montali, M.: Monitoring constraints and metaconstraints with temporal logics on finite traces. TOCEM (2022)
12. Demri, S., Lazic, R.: LTL with the freeze quantifier and register automata. ACM Trans. Comput.Logic **10**(3), 1–30 (2009)
13. Esser, S., Fahland, D.: Multi-dimensional event data in graph databases. J. Data Semant. **10**(1), 109–141 (2021). <https://doi.org/10.1007/s13740-021-00122-1>
14. Fdhila, W., Gall, M., Rinderle-Ma, S., Mangler, J., Indiono, C.: Classification and formalization of instance-spanning constraints in process-driven applications. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 348–364. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_20
15. Giacomo, G.D., Felli, P., Montali, M., Perelli, G.: HyperLDL: a logic for checking properties of finite traces process logs. In: IJCAI (2021)
16. Gupta, A., Mumick, I.S. (eds.): Materialized Views: Techniques, Implementations, and Applications. MIT Press, Cambridge (1999)
17. Gupta, A., Mumick, I.S.: Maintenance of materialized views: problems, techniques, and applications. IEEE Data Eng. Bull. **18**(2), 3–18 (1995)
18. Gupta, A., Mumick, I.S., Subrahmanian, V.S.: Maintaining views incrementally. SIGMOD **22**, 157–166 (1993)
19. IEEE 1849–2016 XES Standard. <https://www.xes-standard.org/>
20. Indiono, C., Mangler, J., Fdhila, W., Rinderle-Ma, S.: Rule-based runtime monitoring of instance-spanning constraints in process-aware information systems. In: Debruyne, C., et al. (eds.) OTM 2016. LNCS, vol. 10033, pp. 381–399. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48472-3_22
21. Kennedy, O., et al.: DBToaster: agile views for a dynamic data management system. In: CIDR (2011)
22. Koch, C., et al.: DBToaster: higher-order delta processing for dynamic. Frequently Fresh Views. VLDB J. **23**(2), 253–278 (2014). <https://doi.org/10.1007/s00778-013-0348-4>
23. LDBC graph query language task force: G-CORE: a core for future graph query languages. SIGMOD (2018)
24. 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>
25. Maggi, F.M., Montali, M., Westergaard, M., van der Aalst, W.M.P.: Monitoring business constraints with linear temporal logic: an approach based on colored automata. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 132–147. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23059-2_13
26. Mecca, G., Bonner, A.: Query languages for sequence databases: termination and complexity. IEEE TKDE **13**(3), 519–525 (2001)
27. Montali, M., et al.: Monitoring business constraints with the event calculus. ACM Trans. Intell. Syst. Technol. **5**(1), 1–30 (2013)
28. de Murillas, E.G.L., Reijers, H.A., van der Aalst, W.M.P.: Connecting databases with process mining: a meta model and toolset. Softw. Syst. Model. **18**(2) (2019). <https://doi.org/10.1007/s10270-018-0664-7>
29. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: EDOC (2007)

30. Rinderle-Ma, S., Gall, M., Fdhila, W., Mangler, J., Indiono, C.: Collecting examples for instance-spanning constraints. [arXiv:1603.01523](https://arxiv.org/abs/1603.01523) (2018)
31. Shen, W., et al.: Declarative information extraction using datalog with embedded extraction predicates. In: VLDB (2007)
32. DBToaster Webpage. <https://dbtoaster.github.io/index.html>
33. Warner, J., Atluri, V.: Inter-instance authorization constraints for secure workflow management. In: SACMAT (2006)
34. Winter, K., et al.: Discovering instance and process spanning constraints from process execution logs. *Inf. Syst.* **89**, 101484 (2020)

**2nd International Workshop on Business
Process Management and Routine
Dynamics (BPM&RD 2022)**

2nd International Workshop on Business Process Management and Routine Dynamics (BPM&RD 2022)

The workshop on Business Process Management (BPM) and Routine Dynamics (RD) aims to bring together researchers who are interested in how organizational work is performed. Research has described BPM and RD as “two islands of process research” [1]. While both fields are interested in the study of organizational processes, there is only limited exchange between them. To leverage the synergies between both fields [1, 2], we organized and held the 2nd Workshop on Business Process Management and Routine Dynamics in conjunction with the 20th International Conference on Business Process Management in Münster, Germany. This year, the workshop attracted two submissions. Each submission was reviewed by three members of the program committee. We accepted one paper for presentation at the workshop and invited Hajo Reijers to give a keynote reflecting on the history of BPM.

First, Pentland et al. [3] presented their paper on the effects of concurrency in complex service organizations. Based on digital trace data from the healthcare records of a dermatology clinic, they perform regression analysis to determine the effect of concurrency on the duration of a patient visit. Their core finding is that the synchronization of resources during a visit (within concurrency) reduces the visit’s duration and that the number of visits that overlap with a given visit (between concurrency) prolong a respective patient visit. Subsequently, Hajo Reijers gave a keynote on the history of BPM. He walked the audience through the beginnings of the BPM discipline, outlined the recent decades of BPM research, and discussed links of BPM to other disciplines.

We would like to thank everyone who contributed to this year’s iteration in one way or another. We want to thank all authors for submitting their work to our workshop. We thank the program committee members for their reviews and the workshop chairs for their support regarding all organizational matters. Last, we thank all workshop participants for their attendance and contributions to the workshop.

For the upcoming iterations of the workshop, we encourage all kinds of thought-provoking papers that help strengthen the connection between BPM and RD. We are open to papers irrespective of methods and exact topics chosen. Amongst other things, we are curious about technical papers that help us to better analyze organizational processes and behavioral papers that use process mining and related methods to examine and theorize process dynamics on the grounds of digital trace data [2, 4]. We are happy to provide early feedback on the fit of a research project or paper with the workshop to prospective authors.

November 2022

Bastian Wurm
Thomas Grisold
Waldemar Kremser
Jan Mendling

Organization

Workshop Chairs

Bastian Wurm	LMU Munich School of Management, Germany
Thomas Grisold	University of Liechtenstein, Liechtenstein
Waldemar Kremser	Johannes Kepler Universität Linz, Austria
Jan Mendling	Humboldt-Universität zu Berlin, Germany

Program Committee




Christian Bartelheimer	Paderborn University, Germany
Markus Becker	University of Southern Denmark, Denmark
Iris Beerepoot	Utrecht University, the Netherlands
David Chapela-Campa	University of Tartu, Estonia
Christian Mahringer	University of Stuttgart, Germany
Martin Matzner	University Erlangen-Nürnberg, Germany
Jan Recker	University of Hamburg, Germany
Maximilian Röglinger	Universität Bayreuth, Germany
Christoph Rosenkranz	University of Cologne, Germany
Anton Yeshchenko	Vienna University of Economics and Business, Austria

References

1. Pentland, B.T., Vaast, E., Wolf, R.: Theorizing process dynamics with directed graphs: A diachronic analysis of digital trace data. *MIS Quarterly*. **45**, 967–984 (2021)
2. Wurm, B., Grisold, T., Mendling, J., vom Brocke, J.: Business Process Management and Routine Dynamics. In: *Cambridge Handbook of Routine Dynamics*. pp. 513–524. Cambridge University Press (2021)
3. Pentland, B.T., Kim, I., Zhang, Q., Wolf, J.R.: Effects of Concurrency in Complex Service Organizations: Evidence from Electronic Health Records. In: *BPM 2022 Workshop Proceedings*. LNBIP (2022)
4. Grisold, T., Wurm, B., Mendling, J., vom Brocke, J.: Using Process Mining to Support Theorizing About Change in Organizations. In: *53rd Hawaiian International Conference on System Sciences (HICSS 2020)* (2020)



Effects of Concurrency in Complex Service Organizations: Evidence from Electronic Health Records

Brian T. Pentland¹  , Inkyu Kim¹, Quan Zhang¹, and Julie Ryan Wolf² 

¹ Michigan State University, East Lansing, MI, USA
Pentland@broad.msu.edu

² University of Rochester, Rochester, NY, USA

Abstract. We use Kremser and Blagoev's [1] role-routine ecology to theorize about the effects of concurrency in complex service organizations, such as outpatient medical clinics. In a typical clinic, teams of specialized individuals serve multiple clients at the same time. There can be concurrency within a patient visit (a technician may be preparing for a procedure while the doctor talks to the patient) and concurrency between patient visits (multiple patients being treated in the clinic). Using data from electronic health records, we estimate the effects of concurrency within and between patient visits on the duration of patient visits in a set of dermatology clinics. As expected, we find that concurrency within patient visits is associated with reduced duration, while concurrency between visits is associated with increased duration. We discuss the implication of these findings for process mining and discovery of process models in organizations where process instances are not independent.

Keywords: Organizational routines · Role-routine ecology · Concurrency · Electronic health records

1 Introduction

A hallmark of process mining has been the focus on concurrency within a process. Van der Aalst [2] places an emphasis on concurrency and objects as way to reveal the “true fabric of business processes” [3]. At the same time, one of the on-going challenges in process mining and process management concerns the execution of multiple, concurrent process instances [4, 5]. For example, in an outpatient medical clinic, there is almost always more than one patient in the clinic at a time. As a result, the patients in the clinic are competing for the time and attention of the clinical staff. There is concurrency within patient visits, but also between patient visits. Thus, the outpatient medical clinic is an interesting context for theorizing about organizations where process instances are not independent.

In this paper, we examine the effects of concurrency within and between process instances using the role-routine ecology, a new conceptual framework from Kremser and Blagoev [1]. In a role-routine ecology, work is organized by the competing needs of

the routines (e.g., treating a patient) and the roles (e.g., being a physician in a clinic). In organizations that have a complex role-routine ecology, where multiple roles are engaged in multiple routines at the same time, the workflow is emergent. By emergent, we mean that the “existence and nature” of the behavior “depend upon entities at a lower level, but the behavior is neither reducible to, nor predictable from, properties of entities found at the lower level” [6]. We argue that the nature of the role-routine ecology has implications for process mining and discovery of process models.

We begin by defining the role-routine ecology and the concept of concurrency within and between process instances. Then we analyze electronic health record (EHR) data from a set of outpatient dermatology clinics to demonstrate the effects of concurrency. Finally, we discuss the implications of these findings in terms of the role-routine ecology. In a complex role-routine ecology, workflow is an emergent product of competing priorities, which raises challenges for conventional process mining and for the prospects of more sophisticated models, such as Digital Twins of Organizations [7, 8].

2 Background

2.1 Role-Routine Ecology in Complex Service Organizations

Kremser and Blagoev [1] introduce the concept of a role-routine ecology as a way to analyze the competing priorities that govern the work processes in a consulting organization. On one hand, actions are prioritized according to the needs of the routine: the “repetitive, recognizable pattern[s] of interdependent actions, involving multiple actors” [9] that are oriented toward the accomplishment of a “day-to-day operational task” [10]. On the other hand, actions are prioritized based on the needs of the role. Kremser and Blagoev [1] argue that “a role performance is a sequence of actions...”. Like routines, roles are not static; they are “continuously constructed and reconstructed as individuals engage in... Interaction with incumbents of alter roles” [11].

Like routines, roles can be conceptualized as patterns of action, but the logic of their enactment is different. Within a routine, the logic of enactment is analogous to control flow in a business process, where one action triggers the next [12]. Sequential triggering of actions within a routine gives rise to a recognizable pattern. To the extent that patient treatment is routinized, it should have a recognizable pattern of actions.

Within a role, however, the logic of what to do next may have little to do with the flow of the routine. For example, the office assistant who checks patient into the clinic serves each patient as they arrive. They perform roughly the same actions for each patient who arrives and they are not concerned with (or aware of) the rest of the process. The clinical technician who brings the patient to the room and takes vital signs has a similarly limited role. These specialized roles perform repetitive, recognizable patterns of actions, but they are driven by the functional requirements of the role. They see every patient, but they see only one part of the overall clinical routine.

Using data collected through participant observation, Kremser and Blagoev [1] analyze the patterns of action in an organization where consultants with specialized roles work on several concurrent projects. Kremser and Blagoev [1] note that the needs of each client workflow can change unexpectedly, as can the availability of the consultants. Changes in one workflow can cascade through the other workflows. Such cascading

interactions are common in outpatient clinics, where an unexpectedly difficult patient can tie up the clinical staff and delay the treatment of other patients.

We can contrast the complex service organization studied by Kremser and Blagoev [1] with other, less complex organizational forms, as in Fig. 1.

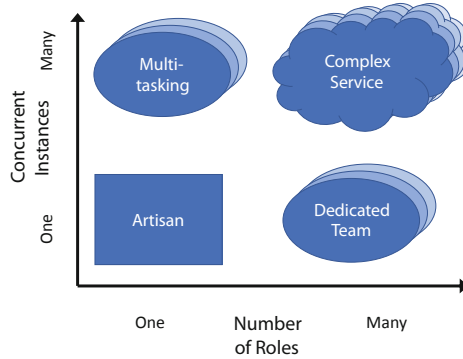


Fig. 1. Four kinds of role-routine ecology

The horizontal axis of Fig. 1 refers to the number of distinct roles involved in providing the service. For example, in a typical clinical visit, there may be a nurse, a physician and an office administrator, plus other roles. The vertical axis refers to the number of concurrent process instances (e.g., patients in the clinic at the same time). Using these two dimensions, four archetypal role-routine ecologies are easily identified.

- Artisan: One physician treats one patient at a time (e.g., sole practitioner).
- Dedicated team: a group of specialists treats one patient at a time (e.g., surgery).
- Multi-tasking: one nurse cares for several patients at the same time (e.g., in an in-patient hospital ward).
- Complex service: a team of specialists treats several patients at the same time (e.g., a typical outpatient clinic).

A wide variety of work processes fall into the category of complex service organizations, such as restaurants [13], professional service firms [1], software development teams [14], and the example we use here, outpatient medical clinics.

2.2 Concurrency in Complex Service Organizations

Concurrency is a pervasive aspect of the complex service organization. Concurrency can be formally defined in terms of Petri nets [15] which are widely used to represent business processes. In a Petri net, two events are *concurrent* if they occupy parallel paths in the network. When two or more events are concurrent, the specific sequence of their execution is irrelevant to the outcome of the process, as long as they are all completed. For example, van der Aalst [2] uses a Petri net to model a medical process where “lab

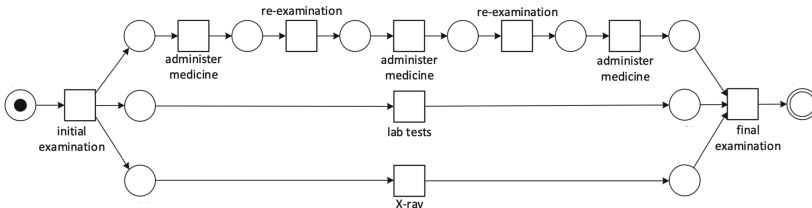


Fig. 2. Concurrency in a Petri net

tests” and “x-ray” occur concurrently with each other and with a series of other activities, as shown in Fig. 2.

In this paper, concurrency refers to actual overlap in time. As we use the term here, two events are concurrent if one starts before the other finishes. For example, in an outpatient clinic, two patient visits are concurrent if one patient arrives before the other patient leaves. From the perspective of the whole days’ work at the clinic, those patients could be seen in any order. However, the additional restriction of temporal overlap reflects the reality of clinical work. Treating multiple patients at the same time adds complexity to the clinical process because they are competing for resources [16].

2.3 Concurrency Within Process Instances

Traditional research on process management and process mining has emphasized the importance of concurrency within process instances [2]. While dependencies between multiple process instances is a recognized issue, mainstream theory and practice in process management generally treat process instances as independent. Process mining algorithms generally attempt to identify control flow within a process instance, rather than dependencies between process instances [3, 17].

2.4 Concurrency Between Process Instances

At the same time, research on process management has long recognized that workflows may be interdependent [18, 19]. When multiple instances of a process occur at the same time, they may compete for the same resources. Multiple instance patterns are defined to “describe situations where there are multiple threads of execution active in a process model which relate to the same activity” [20]. Common activities may also involve sharing common resources, such as a printer [16]. One approach to this general problem has been to treat multiple, concurrent process instances as a single, complex workflow. However, as Heinlein [19] notes, merging workflows (or messaging between workflows) faces a combinatorial explosion. If providers are serving n concurrent workflows, then “ 2^n variations of each workflow would be necessary in principle to describe its behaviour in every possible combination with n other workflows”.

The challenge of handling multiple process instances continues to be an active area of research in process management. Traganos, Spijkers, Grefen and Vanderfeesten [21] note that BPMN (Business Process Management Notation) lacks strong support for operations such as buffering, bundling, and unbundling physical objects in a manufacturing

workflow (such as parts on a palette). These types of operations are needed to handle multiple, concurrent process instances. Senderovich, Leemans, Harel, Gal, Mandelbaum and van der Aalst [22] analyze the use of event logs to discover queues. Suriadi, Wynn, Xu, van der Aalst and ter Hofstede [23] propose a method for discovering prioritization from event logs. Fahland, Denisov and van der Aalst [5] note that queueing for shared resources can introduce unexpected behavior in a process which is “particularly important for distributed systems with shared resources, e.g., one case can block another case competing for the same machine, leading to inter-case dependencies in performance.” Klijn, Mannhardt and Fahland [4] have proposed a graph-based framework for analyzing the inter-case dependencies involving actions and actors in digital trace data.

2.5 Competing Effects of Concurrency

Within a process instance, concurrency generally facilitates efficiency. When activities can be performed in parallel, it tends to increase the capacity of a work system. And when those activities can be performed in any sequence, it adds flexibility to a work system.

However, when there is concurrency between process instances, the effects are not so clear cut. The effects depend on the level of available resources and structure of the work system. If resources are limited, concurrent instances will be in competition [16]. The nature of that competition will be defined, in part, by the structure of the role-routine ecology in the organization. In an organization of artisans, each of whom works independently on a single process instance, resource constraints should be manifest in queuing [22]. In contrast, in a complex service organization, where multiple roles serve multiple clients at the same time, the effects of concurrency will depend on how work is coordinated, as well as resource constraints.

3 Research Context

To investigate this phenomenon, we analyze data from dermatology clinics at an academic medical center in the Northeastern U.S. We chose this setting because it provides a clear example of a complex service organization with multiple roles and multiple concurrent process instances (patient visits).

3.1 EHR Audit Trail Data

The EHR audit trail is an ideal resource for analyzing the clinical documentation process because every action by each provider who touches the clinical record is time-stamped and recorded. Providers include nurses, physicians, technicians, office assistants, insurance specialists, administrators, and others. The audit trail is not the full patient medical record; it is a separate database of who did what. It does not contain notes, test results, medications, billing information, costs or any other information about the content or outcomes of the medical services performed. For this study, we use audit trail data from the EPIC EHR system. The data traces the clinical documentation process for patient visits from three dermatology clinics from January 2016 through December 2017 for a total of 21,785 patient visits.

3.2 Concurrency in the Clinic

Concurrency is built into the physical layout of the clinics. In each of the dermatology clinics we studied, there were multiple exam rooms. Providers move between rooms, from patient to patient, as they do their work. For example, after a technician records pulse and blood pressure for one patient, they leave the room to perform some other tasks and someone else continues the visit with that patient. The overall workflow depends on which patient happens to be in the next room and what needs to be done. And of course, whatever gets done needs to be documented. In this way, the EHR documentation work is woven into the fabric of the medical work. The audit trail data provides a detailed record that we can use to examine the temporal structure of this fabric. It allows us to see two layers of concurrency in the clinic:

- Between patient visits, there is a great deal of concurrency. We can see this very accurately in the EHR audit trail data.
- Within patient visits, there is also some concurrency. We measure this with the EHR audit trail data, using the method described by Iqbal and Riek [24] but the measure is not perfect because many actions are not directly recorded.

When we view the event log for a single patient visit, it is easy to overlook the fact that there is more than one patient in the clinic at the same time. Idealized models of a process often assume that concurrent instances are independent. That is clearly not the case in medical practice. There are almost always multiple patients competing for time and attention. In our data, the average number of concurrent patients was 6.35. The maximum was 27.

4 Methods

We use the audit trail data to show how process duration is influenced by concurrency within and between process instances. To do so, we control for everything that might influence duration so we can more accurately estimate the effects of concurrency. To be clear, the data we analyze here was collected as part of a larger study, so we are relying on available metrics for this analysis.

4.1 Descriptive Statistics

Table 1 describes all the variables that we used in this analysis. We use a natural log transformation for all variables except the dummy variables. Table 2 provides the correlation for the main variables in the analysis.

Table 1. Descriptive statistics of variables.

Variables	Mean	sd	min	max	Description
<i>Duration</i>	8.32	0.53	0	10.15	Time between the first and last activity
<i>Concurrency WITHIN</i>	0.45	0.20	0	1	Level of concurrency among actors in the visit
<i>Concurrency BETWEEN</i>	1.88	0.51	0	3.34	Number of other visits that overlap in time with this visit
<i>NProviders</i>	1.61	0.20	1.38	2.40	Number of providers in visit
<i>NProcedures</i>	0.79	0.42	0	4.49	Number of procedures performed in visit
<i>Newbies</i>	0.07	0.26	0	1	Any new employees on this visit? (0/1)
<i>FirstVisit</i>	0.55	0.50	0	1	Is this the first visit for this patient? (0/1)
<i>Diagnosis</i>					Complexity of diagnosis (3 levels)
<i>CPT Code</i>					Billing code for level of service (5 levels)
<i>Clinic</i>					Which clinic? (3 levels)
<i>Month</i>					Which month? (12 levels)

Table 2. Spearman correlation matrix for variables (n = 21,785)

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
1. Duration	1						
2. <i>WITHIN</i>	-0.522	1					
3. <i>BETWEEN</i>	0.408	-0.399	1				
4. NProcedures	0.082	-0.024	0.055	1			
5. NProviders	0.261	-0.185	0.094	0.063	1		
6. Newbies	0.025	-0.011	-0.036	0.002	0.023	1	
7. FirstVisit	-0.014	0.017	-0.018	0.062	-0.022	0.020	1

5 Model Specification and Findings

To examine the effect of concurrency on duration of the patient visits, we used the main empirical specification as follows:

$$\begin{aligned} \text{Duration} = & \beta_0 + \beta_1(\text{WITHIN}) + \beta_2(\text{BETWEEN}) + \beta_3(\text{NProcedures}) \\ & + \beta_4(\text{NProvider}) + \beta_5(\text{Newbies}) + \beta_6(\text{First}_{\text{visit}}) + \alpha + \lambda + \gamma + \delta \end{aligned}$$

where *Duration* is the log of duration of the patient visit, concurrency *WITHIN* is the level of overlap between actions within each visit and concurrency *BETWEEN* is the number of other patients in the clinic. We control for a number of other variables, including the number of providers (*NProviders*), the number of procedures performed during the visit (*NProcedures*), the involvement of new workers (*Newbies*), and whether this was the first visit for this patient (*First_Visit*). We further add fixed effects in the model to account for heterogeneity due to the clinic (α), diagnosis complexity (λ), level of service (γ), and monthly seasonality (δ).

Table 3 reports the estimated effects of concurrency on duration in the patient visits. In column (1), we use only the control variables and fixed effects. In columns (2) and (3), we show to add the stepwise effects of concurrency within and between visits. In column (4), we show the full model. These are standardized coefficients so we can compare their relative magnitudes.

As expected, each aspect of concurrency is associated with a significant change in the duration of patient visits. Concurrency within visits speeds up the work, while concurrency between slows it down. Concurrency within visits is the largest effect, roughly three times the size of concurrency between visits. However, when we take both effects into account at the same time, as in column (4), their magnitudes are somewhat reduced.

The control variables also provide some interesting insights. For example, the number of providers involved in a visit increases the duration as much as concurrency between visits. This is because each type of provider has a specialized role. In the simplest (fastest) visits, the patient interacts with 1–2 clinical staff members. If a visit requires the attention of more staff members, it is likely to involve a more elaborate and time-consuming process. Interestingly, the number of procedures performed (e.g., freezing a wart) has a comparatively small effect on duration. Likewise, the involvement of individuals who are new to their jobs (newbies) has a small positive effect. However, contrary to our expectations, first-time visits are not longer than follow-up visits.

Table 3. OLS regression results (standardized coefficients)

	(1)	(2)	(3)	(4)
Variables	Controls	Within	Between	Both
WITHIN		-1.5323*** (0.0492)		-1.1912*** (0.0431)
BETWEEN			0.4394*** (0.0096)	0.3908*** (0.0089)
NProviders	0.5111*** (0.0226)	0.4666*** (0.0220)	0.4476*** (0.0217)	0.4200*** (0.0213)
NProcedures	0.0581*** (0.0081)	0.0640*** (0.0078)	0.0649*** (0.0077)	0.0688*** (0.0074)
Newbies	0.0550*** (0.0128)	0.0505*** (0.0124)	0.0722*** (0.0121)	0.0668*** (0.0118)
FirstVisit	-0.0085 (0.0092)	0.0026 (0.0087)	-0.0072 (0.0084)	0.0014 (0.0081)
Constant	7.1051*** (0.1142)	7.4038*** (0.1064)	6.5918*** (0.1043)	6.8808*** (0.0985)
Observations	21,800	21,800	21,800	21,800
R-squared	0.1400	0.2026	0.2501	0.2865
LOS fixed effects	YES	YES	YES	YES
Clinic fixed effects	YES	YES	YES	YES
YM dummies	YES	YES	YES	YES
Diagnosis effects	YES	YES	YES	YES

Robust standard errors in parentheses

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

6 Discussion

Without question, concurrency is crucial to the true fabric of organization, but concurrency occurs in layers, within and between processes. In terms of the fabric metaphor, organizational fabric has multiple layers and they are loosely stitched together. In simple organizations, where processes and process instances are independent, it is relatively easy to understand the effects of concurrency within a process. But in complex service organizations, where multiple providers serve multiple concurrent clients, it is not so easy, because concurrency within each process instance interacts with concurrency between process instances. We can see in Table 2 that there is a strong negative correlation ($r = -0.399$) of concurrency within and between. When there are more patients in the clinic, the clinical staff are spread thin. They are less likely to be working on the same patient at the same time. They are more likely to be working on different patients.

This relationship will be different in other kinds of organizations, but it points to the possibility of multi-layer interactions.

6.1 What Controls the Flow?

An important insight from the role-routine ecology is that process flow can be controlled by multiple, competing priorities or logics [25, 26]. The next action or event is not necessarily triggered by actions or events in the same case (i.e., the same patient visit). Rather, it could be triggered by a pattern of action implied by the roles of the clinical staff.

Routines can be understood in terms of a sequential, control flow logic, where one event triggers the next. For some roles, “control flow” works very differently. For the office staff who check patients in, the work consists of checking in the next patient. For the clinical technician, the work consists of getting vital signs for the next patient. These specialized roles contribute the same steps to each patient visit. These roles just take the next patient in the queue. Similarly, as the nurses and physicians work their way around the exam rooms, from one patient to the next, they are executing role-based patterns of action. The fact that patients can be seen in any order (the broad definition of concurrency) adds flexibility to the workflow, but it also adds complexity to the event log.

6.2 Emergent Complexity and Model Quality

In research on organizational routines, there has been growing interest in the antecedents and consequences of complexity [27–29]. This research treats process complexity as the emergent product of situated actions. In research on process mining, Augusto, Mendling, Vidgof and Wurm [30] demonstrate that event log complexity can influence the quality of models discovered through conventional process mining. Their analysis starts from the complexity event log. Here, we are stepping back to consider why some event logs are more complex than others.

In a simple role-routine ecology, role logic and routine logic are likely to be aligned. For example, in a simple organization of artisans, where one individual performs one process instance at a time, the event log for each process instance will be independent of other instances. In this idealized case, the pattern of action for the role and the routine should be the same. This is the best-case scenario for process mining and the discovery of concurrency within the process. However, as we add multiple roles and concurrent process instances, the best-case scenario breaks down. In a more complex role-routine ecology, there is inevitably some conflict between roles and routines. To the extent that the event log is an emergent product of these competing logics, playing out over concurrent process instances, it will be more difficult to model.

7 Conclusion

As process mining capabilities become more mature, there is growing interest in more sophisticated applications of process discovery, such as digital twins of organizations

[7, 8]. Such models seem plausible for the best-case scenario when concurrency is limited to within process instances, and the control flow is governed by a single, uniform logic. However, in more complex organizations, where there are interdependent process instances (not to mention interdependent processes) and competing logics for each of them, discovering and modeling the true fabric of organization is inherently more difficult.

Acknowledgements. This material is based upon work supported by the National Science Foundation under Grants No. SES-1734237 and BCS-2120530. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This research was also supported in part by University of Rochester CTSA (UL1 TR002001) from the National Center for Advancing Translational Sciences (NCATS) of the National Institutes of Health (NIH). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. We are grateful to the reviewers for their comments.

References

1. Kremser, W., Blagoev, B.: The dynamics of prioritizing: how actors temporally pattern complex role-routine ecologies. *Adm. Sci. Q.* **66**, 339–379 (2021)
2. van der Aalst, W.M.: Concurrency and objects matter! Disentangling the fabric of real operational processes to create digital twins. In: Cerone, A., Ölveczky, P.C. (eds.) *ICTAC 2021*. LNCS, vol. 12819, pp. 3–17. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85315-1_1
3. van der Aalst, W.M.: Business process management as the “killer app” for Petri nets. *Softw. Syst. Model.* **14**, 685–691 (2015)
4. Klijn, E.L., Mannhardt, F., Fahland, D.: Classifying and detecting task executions and routines in processes using event graphs. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) *BPM 2021*. LNBP, vol. 427, pp. 212–229. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85440-9_13
5. Fahland, D., Denisov, V., van der Aalst, W.: Inferring unobserved events in systems with shared resources and queues. *Fund. Inform.* **183**, 203–242 (2021)
6. Hodgson, G.M.: Institutions and individuals: interaction and evolution. *Organ. Stud.* **28**, 95–116 (2007)
7. Park, G., Van Der Aalst, W.M.: Realizing a digital twin of an organization using action-oriented process mining. In: *2021 3rd International Conference on Process Mining (ICPM)*, pp. 104–111. IEEE (2021)
8. van der Aalst, W.M., Hinz, O., Weinhardt, C.: Resilient digital twins. *Bus. Inf. Syst. Eng.* **63**, 615–619 (2021)
9. Feldman, M.S., Pentland, B.T.: Reconceptualizing organizational routines as a source of flexibility and change. *Adm. Sci. Q.* **48**, 94–118 (2003)
10. Rerup, C., Feldman, M.S.: Routines as a source of change in organizational schemata: the role of trial-and-error learning. *Acad. Manag. J.* **54**, 577–610 (2011)
11. Turner, J.H.: *Handbook of Sociological Theory*. Handbooks of Sociology and Social Research, p. 730. Springer, Dordrecht (2006)
12. Cohen, M.D., Bacdayan, P.: Organizational routines are stored as procedural memory: Evidence from a laboratory study. *Organ. Sci.* **5**, 554–568 (1994)

13. Tanizaki, T., Shimmura, T.: Modeling and analysis method of restaurant service process. *Proc. CIRP* **62**, 84–89 (2017)
14. Dikert, K., Paasivaara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: a systematic literature review. *J. Syst. Softw.* **119**, 87–108 (2016)
15. Murata, T.: Petri nets: Properties, analysis and applications. *Proc. IEEE* **77**, 541–580 (1989)
16. Mangler, J., Rinderle-Ma, S.: Rule-based synchronization of process activities. In: 2011 IEEE 13th Conference on Commerce and Enterprise Computing, pp. 121–128. IEEE (2011)
17. Rojas, E., Munoz-Gama, J., Sepúlveda, M., Capurro, D.: Process mining in healthcare: a literature review. *J. Biomed. Inform.* **61**, 224–236 (2016)
18. Alonso, G., Agrawal, D., El Abbadi, A.: Process synchronization in workflow management systems. In: *Proceedings of SPDP 1996: 8th IEEE Symposium on Parallel and Distributed Processing*, pp. 581–588. IEEE (1996)
19. Heinlein, C.: Workflow and process synchronization with interaction expressions and graphs. In: *Proceedings 17th International Conference on Data Engineering*, pp. 243–252. IEEE (2001)
20. van der Aalst, W.M., Ter Hofstede, A.H., Kiepuszewski, B., Barros, A.P.: Workflow patterns. *Distrib. Parallel Databases* **14**, 5–51 (2003)
21. Traganos, K., Spijkers, D., Grefen, P., Vanderfeesten, I.: Dynamic process synchronization using bpmn 2.0 to support buffering and (un) bundling in manufacturing. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) *BPM 2020. LNBP*, vol. 392, pp. 18–34. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58638-6_2
22. Senderovich, A., Leemans, S.J.J., Harel, S., Gal, A., Mandelbaum, A., van der Aalst, W.M.P.: Discovering queues from event logs with varying levels of information. In: Reichert, M., Reijers, H.A. (eds.) *BPM 2015. LNBP*, vol. 256, pp. 154–166. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42887-1_13
23. Suriadi, S., Wynn, M.T., Xu, J., van der Aalst, W.M., ter Hofstede, A.H.: Discovering work prioritisation patterns from event logs. *Decis. Support Syst.* **100**, 77–92 (2017)
24. Iqbal, T., Riek, L.D.: A method for automatic detection of psychomotor entrainment. *IEEE Trans. Affect. Comput.* **7**(1), 3–16 (2015)
25. Thornton, P.H., Ocasio, W.: Institutional logics. *Sage Handb. Organ. Inst.* **840**, 99–128 (2008)
26. Reay, T., Hinings, C.R.: Managing the rivalry of competing institutional logics. *Organ. Stud.* **30**, 629–652 (2009)
27. Goh, K.T., Pentland, B.T.: From actions to paths to patterning: toward a dynamic theory of patterning in routines. *Acad. Manag. J.* **62**, 1901–1929 (2019)
28. Hansson, M., Hærem, T., Pentland, B.T.: The effect of repertoire, routinization and enacted complexity: explaining task performance through patterns of action. *Organ. Stud.* 01708406211069438 (2021)
29. Danner-Schröder, A., Ostermann, S.M.: Towards a processual understanding of task complexity: constructing task complexity in practice. *Organ. Stud.* **43**, 437–463 (2022)
30. 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)

**15th International Workshop on Social
and Human Aspects of Business Process
Management (BPMS2 2022)**

15th International Workshop on Social and Human Aspects of Business Process Management (BPMS2 2022)

The involvement of human aspects in Business Process Management takes place both on a social and individual level. Social information systems [1], such as social media, Enterprise 2.0, and social platforms, are spreading quickly in society, organizations, and economics. Integrating business process management and social information systems is becoming more widespread [2, 3]. New approaches for using social information systems and business process management appear frequently.

Social information systems are used both in external and internal business processes. Companies can co-create products and services, e.g., integrate customers into product development to capture ideas and features. Thus, communication with the customer is increasingly bi-directional. Social information systems differ from traditional information systems by enabling emergent interactions [4]. Emergent interactions are defined during run-time by two or more stakeholders. No plan or approval from a supervisor or management is necessary. Emergent interactions enable the articulation of personal into the collective knowledge, thus representing mechanisms for harnessing collective intelligence in the digital age.

Integrating business process management and social information systems enables the creation of new business models using social platforms. Prominent examples are TripAdvisor, Uber, and Airbnb. Using the value-creating mechanisms of social information systems, business models that were not realizable before became possible. For example Airbnb uses a crowdsourcing model for quality control by using users' reviews of apartments. In this way, a quality assessment of products and services became possible that was too costly so far.

Human aspects complement the social perspective on business process management. The fact that more and more enterprises are using business process management implies that the human individual is involved in many business processes. Individuals must cope with multiple process contexts and thus must administer data appropriately. Digital assistants such as Alexa [5] integrate individuals into processes that could not interact with conventional computers. Human aspects of business process management relate to the individual who creates a process model, to the communication among people, during and after the process execution, and to the social process of collaborative modeling. They also relate to the interaction/collaboration/coordination/cooperation that should be implemented in the business process or to specific human-related aspects of the business process and their representations in models.

Before this background, the goal of the BPMS2 workshop [6] is to explore how social information systems integrate with business process management [7], and how business process management may profit from this integration [8]. Furthermore, the workshop investigates the human aspects of Business Process Management by involving human actors. Examples are using crowdsourced knowledge and tasks, and the need for new user interfaces, e.g., augmented reality and voice bots.

Two papers were accepted for presentation at the BPMS2 2022 workshop. Tatiane Neves Lopes, Renata Mendes de Araujo, Tadeu Moreira de Classe, and Thayná Gomes proposed the development of a particular type of game, defined as Business-Process-Based Digital Games (BPBDG), and its application in their paper with the title “PYP4Training - Ludifying Business Process Training”. They used a Design Science Research approach and explored the development and application of BPBDG for process training in an organization in the judicial sector. The results showed that the game has playability and learning potential.

In their paper with the title “On Current Job Market Demands for Process Mining: A Descriptive Analysis of LinkedIn Vacancies”, Simin Maleki, Amy Van Looy, Babara Weber, and Maximilian Röglinger explored current job market demands in process mining by means of an empirical and analytical study. Their dataset uncovered a wide variety of vacancies from 47 countries. The vacancies are issued by end-user companies, vendors, or consultancy firms and include a combination of technical or business orientations.

We wish to thank all the people who submitted papers to BPMS2 2022 for sharing their work with us, the many participants creating fruitful discussions, and the members of the BPMS2 2022 Program Committee, who made a remarkable effort in reviewing the submissions. We also thank the organizers of BPM 2022 for their help with the organization of the event.

Selmin Nurcan
Rainer Schmidt

Organization

Program Committee

Alfred Zimmermann

Ralf Klamma

Flavia Santoro

Holger Günzel

Norbert Gronau

Kathrin Kirchner

Jan Bosch

Moe Thandar Wynn

Marco Brambilla

Mohammad Ehson Rangiha

Gustavo Rossi

Miguel-Angel Sicilia

Reutlingen University, Germany

RWTH Aachen University, Germany

UERJ, Brazil

University of Applied Sciences München,
Germany

University of Potsdam, Germany

Technical University of Denmark, Denmark

Chalmers University of Technology, Sweden

Queensland University of Technology,
Australia

Politecnico di Milano, Italy

City University, UK

LIFIA-F. Informatica. UNLP, Italy

University of Alcala, Spain

References

1. Schmidt, R., Alt, R., Nurcan, S.: Social Information Systems. In: Proceedings of the 52nd Hawaii International Conference on System Sciences. pp. 2642–2646, Hawaii (2019)
2. Schmidt, R., Nurcan, S.: BPM and Social Software. In: Ardagna, D., Mecella, M., Yang, J., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., and Szyperski, C. (eds.) Business Process Management Workshops. pp. 649–658. Springer Berlin Heidelberg (2009). <https://doi.org/10.1007/978-3-642-28115-0>
3. Bruno, G., et al.: Key challenges for enabling agile BPM with social software. *Journal of Software Maintenance and Evolution: Research and Practice*. **23**, 297–326 (2011).
4. Schmidt, R., Kirchner, K., Razmerita, L.: Understanding the Business Value of Social Information Systems – A Research Agenda. In: 53rd Hawaii International Conference on System Sciences (HICSS). pp. 2639–2648 (2020). <https://doi.org/10.24251/HICSS.2020.321>
5. Schmidt, R., Alt, R., Zimmermann, A.: A Conceptual Model for Assistant Platforms. In: 54th Hawaii International Conference on System Sciences (HICSS). pp. 4024–4033, Wailea (2021). <https://doi.org/10.24251/HICSS.2021.490>.
6. Nurcan, S., Schmidt, R.: Introduction to the First International Workshop on Business Process Management and Social Software (BPMS2 2008). In: Business Process Management Workshops. pp. 647–648 (2009).
7. Schmidt, R., Nurcan, S.: Augmenting BPM with Social Software. In: Business Process Management Workshops. pp. 201–206, Ulm (2010).
8. Erol, S., et al.: Combining BPM and social software: contradiction or chance? *Journal of Software Maintenance and Evolution: Research and Practice*. **22**, 449–476 (2010). <https://doi.org/10.1002/smr.460>



PYP4Training - Ludifying Business Process Training

Tatiane Neves Lopes^{1(✉)}, Renata Mendes de Araujo^{1,2,3(✉)},
Tadeu Moreira de Classe^{4(✉)}, and Thayná Gomes²

¹ University of São Paulo (USP), São Paulo, Brazil
`tatiane.n.lopes@usp.br`

² Mackenzie Presbyterian University (UPM), São Paulo, Brazil
`renata.araujo@mackenzie.br`

³ Brazilian National School of Public Administration, Brasilia, Brazil

⁴ Federal University of the State of Rio de Janeiro (UNIRIO), Rio de Janeiro, Brazil
`tadeu.classe@gmail.com`

Abstract. Process training is an important activity in the business process management lifecycle, when the organizational actors must be taught in how to perform existing or redesigned work processes. Process training is a continuous activity performed whenever processes are significantly changed, innovations are introduced or new professionals are integrated to the organization. Due to their motivational nature, games have been seen as alternatives to support process training in organizations. However, the approaches to design games for ludifying process training in organizations, presented in the literature, are incipient in methods and results. This research proposes the development of a particular type of games, defined as Business-Process-Based Digital Games (BPBDG), and its potential application in process training. Using Design Science Research, we conducted the first research cycle by exploring the development and application of BPBDG for process training in an organization in the judicial sector. This is to inform you that as the Institutional email address of the corresponding author is not available in the manuscript, we are displaying the private email address in the PDF and Springer-Link. Do you agree with the inclusion of your private e-mail address in the final publication?. The results of this design cycle demonstrated that the game has playability and learning potential from the perspective of the process expert and point out for further research steps.

Keywords: Business process training · Serious digital games · Business process-based digital games · Play your process

1 Introduction

In an increasingly connected world, digital technologies have been one of the main drivers of change in organizations in the search for greater efficiency and effectiveness in their business processes and leading to a complete change in the

way an organization works [21]. Organizations that apply BPM [6] to face this challenge need to frequently train their professionals to institutionalize new processes and process changes [1]. Process training is an important activity in the business process management lifecycle, when the organizational actors must be taught in how to perform existing or redesigned work processes. Process training is a continuous activity performed whenever processes are significantly changed, innovations are introduced or new professionals are integrated to the organization. Process training is considered an important activity in the BPM lifecycle, where processes designed for the organization are effectively institutionalized, and professionals learn and execute business processes [6,27].

Serious games [25] can serve as complementary learning and training tools, acting as triggers to engage people in specific purposes, and in developing new knowledge and skills, accompanied by tension and joy and a feeling of being different from everyday life [9]. Serious games have the potential to improve the efficacy of formative programs, to increase organizational productivity, and to solve problems [20]. However, recent literature reviews [13,14] show that approaches of game design for business process training in organizations are still incipient in methods and results. This research work aims at exploring ideas for ludifying the activity of training actors in organizational processes. Although the term *gamification* is broadly used to refer to approaching real situations as games, the research field differentiates this concept from others. In our research, our aim is to *ludify* (to recreate the work environment and process execution into a virtual and magic world, where work and fun can be balanced, as described by [9]), and not to *gamify* (to include game elements - points, badges etc. - in work activities to stimulate human motivations, as described in [5]).

Previous research [16] argues that serious games (or games with purpose [26]) can make the actors (executors, customers, managers, etc.) involved in a business process understand the functioning and characteristics of these processes, including opportunities for improvement and innovation. [4] defined a specific genre for games for this purpose, called Business-Process-Based Digital Games (BPBDG). These are games capable of presenting business processes playfully and engagingly, allowing players to understand how the processes work. Their players can also develop an awareness of the goals, challenges and characteristics of the organization's business processes. The same research proposed a business-process-based digital game design method - Play Your Process (PYP). This method comprises steps for designing this game genre from a business process model [4].

The Business-Process-Based Digital Games concept and the Play Your Process method can be promising approaches for building games for training business processes. In its original proposition, the PYP aimed to build games focused on understanding business processes by customers or consumers of the processes and not for training actors in the business process execution. Therefore, our research question (RQ) is defined as: **How to build Business-Process-Based Digital Games for business process training?**

In this article, we evaluate the application of PYP to build BPBDG for process training and observe opportunities to improve the method specifically for

this purpose. The research is based on the Design Science Research methodology [7], where we present the first cycle of its execution, exploring the application of PYP in the construction of the Mediator Game (Jogo do Mediador, in portuguese), a game for training the selection process of conflict mediators in the judiciary, and its qualitative evaluation by a specialist in this process.

The article is organized as follows: Sect. 2 presents the concepts that underlie the research. Section 3 discusses related work. Section 4 presents the research design. Section 5 describes the conception of the PYP4Training method and its demonstration in the game design of the Mediator Game. Section 6 presents the limitations. Finally, in Sect. 7, final considerations are presented.

2 Background

2.1 Business Process Training

Training comprises necessary actions to change attitudes, increase knowledge or acquire skills necessary for the adequate performance of human capital in organizations [1]. For Kirkpatrick [11], at least one of the following items - knowledge, skills and attitudes - must be modified so that the change in the professional's behavior at work needs to be considered. Given the growing competitiveness, professionals need to be frequently updated regarding changes in the way the organization works. In this scenario, the traditional forms of training, such as lectures or readings, may not be enough [10]. It is necessary to create and develop an internal culture favorable to learning and committed to the organizational changes. The training objectives go beyond improving the performance and competence of professionals through knowledge, skills and attitudes. They should establish a high degree of motivation and outline the individual responsibility of all parties involved [1]. The human factor is essential for successful business processes [1]. Process training can ensure that all parties involved in the business process can acquire competence in execution and awareness of the relevance of the organizational process.

The classic references of BPM rarely address process training. Although the authors say that training is a critical phase to be considered in organizational management, they do not much explore how to do it [6,27]. No matter how well business process modeling and implementation activities are technically executed, the human component strongly impacts the execution. Thus, professionals must have the training and acquire the necessary competencies and skills to execute the processes as expected.

2.2 Business Process-Based Digital Games

Serious digital games are games that engage the user and contribute to the achievement of a defined purpose [25], that is, games in which there is a secondary objective (the main one is the challenge and the fun) of teaching something to the player, and not intended simply for entertainment [25]. Serious games have also

been explored in organizational process management [13]. In this context, this research is interested in the game genre called Business-Process-Based Digital Games (BPBDG). BPBDG are serious digital games that present a business process in a gamified way and allow players to understand and learn how the process works in a fun and engaging way and develop reflections regarding its need, practice, values, challenges and limitations of execution [4].

This specific game genre implements the conceptual elements presented in a business process model as game elements, based on a conceptual mapping between these elements [4]. This mapping considers specific game genres, for example, the adventure game genre. In this way, actors, activities, rules, resources, events etc., in a process model can be mapped respectively as characters, tasks, rules, resources, events etc., in an adventure game, during game design.

2.3 Play Your Process

The Play Your Process (PYP) is a method of designing digital games based on business process models. PYP guides the designer in building games based on business processes, from conceptualization to evaluation, through iterative steps, based on information obtained from business process models [4]. The steps for executing the method are 1) Context Study: It consists of the understanding of the entire design team about the business process to be implemented in the game. 2) Mapping of process elements to game elements: This step aims to map the elements that will be used in the digital game design from the business process model. 3) Game design: This is a game designer's creativity step, which will define the usual aspects expected for game design in general. The stage is based on Schell's game design vision [24]. 4) Development and prototyping: The development stage comprises the coding of the game in an environment. 5) Validation: The validation step proposes that the games must undergo three evaluations. The first evaluation is with the design team, the second is conducted with the process managers, and the third is conducted with the process actors. 6) Packaging: This step comprises the delivery and publication of the game.

3 Related Work: Serious Games for Process Training

The surveys carried out and reported in the literature (by the authors [14] and by other BPM researchers [13]) showed that the use of digital games for process training is still little explored scientifically and has gaps on how to design and demonstrate the effectiveness of games for process training in practice. Some authors claim that games are an essential mechanism for learning and training the process modeling activity [12, 22, 23]. For Moller and Hansen [17] and Santorum [23], simulation games are widely used for business modeling, learning and process training, and they propose in their studies an approach to improve and understand the business process to motivate stakeholders to create, share, collaborate and maintain business processes in an orderly and straightforward way

with a simulator. Lainema and Makkonen [12] emphasize the need for training models to understand and represent the business process by game participants.

A good part of the analyzed studies has as objective the use of games for learning BPM and/or modeling business processes in educational and organizational environments. Very few works focus on applying and evaluating serious digital games for process training in organizations, i.e., training staff in how to perform existing or new/redesigned business processes using games.

In our research, we intend to demonstrate the possibility of building, using and evaluating BPBDG as an innovative way to process training in organizations. We are not targeting educational settings in BPM and/or organizational management learning for students. We are not also targeting simulation environments, where the real world is digitally reproduced for training purposes. Actually, we are proposing a very specific game genre which turns a business process description into an adventure game.

4 Research Design

The work was based on the epistemological-methodological framework of Design Science Research (DSR) [7]. The DSR presupposes formative research of the construction of an artifact based on design cycles. The starting point of research based on DSR [7, 18] is the definition of the problem and specific context. Process training is our problem in context: the institutionalization of business processes can be hampered by the low knowledge about the process, the fragility in developing skills to execute the process, and the low engagement necessary to train the actors of a process.

Therefore, this research aims to propose an innovative way of conducting process training in organizations that facilitates the development of skills and engages process participants, in this case, using BPBDG. The primary artifact - a business process-based game design method for processes training (PYP4Training) - will be designed based on behavioral conjectures to solve the problem in context: (1) people can learn from games [15], (2) people engage in learning when using games [9], (3) people understand the process of using digital games based on business processes [4, 16], (4) people can develop competencies in business processes during training (knowledge, skill and attitude) when using games [11].

The primary artifact (method) and the secondary artifact (games) will be acceptable to solve our research problem if: the method allows the development of games based on business processes for training the business process, and the games prove to be fun for professionals (players) and be able to promote understanding of the process, as defined by the organization. An empirical evaluation with managers and the target audience should answer the following questions: (1) Does the use of use digital games based on business processes designed by the method allow people to learn during process training? (2) Does the primary artifact enable the design of game training based on business processes? (3) Are the games generated by the solution enjoyable to be used and help players understand and learn the process?

5 PYP4Training Design

This section introduces the first design cycle of PYP4Training, based on PYP in its current version (Subsect. 2.3) and its application for developing a BPBDG for training processes in organizations, particularly in conflict mediation in the judiciary sector. The aim was to build a BPBDG with the potential to train the actors in the process, evaluating the PYP's capacity to build this game. Given that the PYP was not built for this purpose, the contribution of this cycle is to present the PYP adaptation points to advance the PYP4Training design in future cycles. The following sections show the execution of each step of the method for building the game.

5.1 Study of the Context

The Judicial Centers for Conflict Resolution and Citizenship (Os Centros Judiciários de Solução de Conflitos e Cidadania in portuguese [CEJUSCs]) are judicial units of the first instance, preferably responsible for conducting and managing conciliation and pre-procedural and judicial mediation sessions¹. The conflict mediation process includes scheduling a mediation session at one of the CEJUSCs, where the parties can resolve the existing conflict. Then, the mediator explains the process, listens to the conflict between the parties, and helps them reach an agreement to resolve it. For the construction of the BPBDG, the “Pre-Procedural Mediation Process for Conflict Solutions” (Processo de Mediação Pré-Processual de Soluções de Conflitos in portuguese) was selected. The objective of the process is to generate the scheduling of a mediation session between the conflicting parties and the selection of mediators to help the mediation.

The process was analyzed by the designers and in brainstorming with the participation of the process specialist. The analysis highlighted the importance of addressing the training of actors who take on the secretariat role, as the turnover for providing this service is high. In this way, the cut of the process used for the game's development corresponds to the activities of selection of mediators performed by the secretariat, as shown in Fig. 1.

5.2 Element Mapping

This step aimed to map the elements that will be used in the digital game design from the business process model. The mapping of elements followed the guidelines defined in the PYP and was carried out with the support of a computational tool [3]. From this mapping, a first version of the GDD (Game Design Document), a document that presents in detail all the characteristics of a game [8], was created. The mapping result is shown in Table 1.

¹ <https://www.tjrj.jus.br/web/guest/institucional/mediacao/cejusc>.

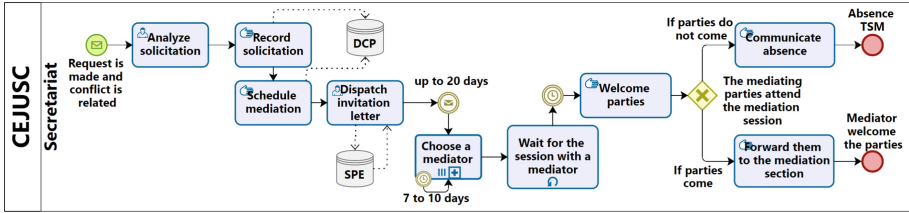


Fig. 1. Pre-procedural mediation process

Table 1. Mapping process model elements to game design

Process model elements	Game elements	Mediator game elements
Participants (lanes)	Player	Secretary
	Characters	Receptionist, Mediators, Intern (Narrative)
Events	Plot: Start event	The notification of request is made and the conflict is reported
	Plot: Failures	Absence TSM
	Plot: Solution	The mediator welcome the parties
Environments/places (lanes or black box)	Game world	Office
Gateways	Rules	R1: If parties do not come, it is required to communicate the absence.
		R2: If parties come, it is needed to forward them to the mediation section.
Process' instances	Story	The story begins when there is a conflict, and the citizen looks for a CEJUSC to help solve it.
	Narrative	The citizen relates a conflict and creates a solicitation and gives it the right way. Then, the secretary analyzes the solicitation and gives it the right way.
Activities (tasks and subprocess)	Tasks	Analyze solicitation, Record solicitation, Dispatch invitation letter, Choose a mediator, Wait for the session with a mediator, Welcome parties, Communicate absence, etc.
	Feedback	All information that is generated inf each activity
Process' flows (sequence, information, or Messages)	Interactions	Player performs task (action - mechanic)
		Player access information/systems (action - mechanic)
	Rules	Sequence: DPC records information - DPC schedules mediation >SPE
	Goal	Goal

5.3 Game Design

In the game design stage, the designers brainstormed to deepen the initial GDD with ideas and concepts regarding the theme, mechanics, and dynamics, to build the game's first version. As a result of this step, design decisions were made, such as positioning the player as one of the actors in the process - the secretary - since this actor/character is the primary target audience of the training. The game comprises the execution of the activities foreseen in the process model. After the secretary becomes aware of the request and analyzes it, scheduling and summoning begin via sending an invitation letter by e-mail. The conflicting parties are received and forwarded to the mediation room on the scheduled date.

The designers chose to create an office environment for the game setting and Q&A dynamic for this activity. It is necessary to answer a quiz after executing the dialog about the necessary information (process rules) and the generated information (feedback) of the activity, as shown in Fig. 2(A). In this way, the

player gets immediate feedback on the knowledge acquired about the analysis request activity. The activity to select mediators (the game mission) has the purpose of selecting the mediators who will participate in the mediation sessions on a day and time scheduled by the Court with the parties (Plaintiff and Defendant). To complete the mission, the player must talk to characters who are candidates for mediation and answer a quiz about the information needed for the selection. The mediator is selected based on his experience, knowledge of the subject at hand, and availability. Figure 2(B) illustrates the dialogue between the mediation secretary and the candidate.



Fig. 2. (A) Task of analyze solicitations (in portuguese). (B) Choose of the mediator (in portuguese).

5.4 Development and Prototyping

The technology for implementing the game was software Construct 3. The narrative was developed in an ad-hoc way. Game mechanics and aesthetics were implemented based on the GDD.

5.5 Validation

The validation step proposes three evaluations: evaluation by the design team, evaluation by the process manager/owner, and evaluation with the process actors. The evaluation of the Mediator Game by the design team comprised technical game design issues and will not be detailed in this paper. Taking into account that the purpose of this research cycle was mainly to explore how the PYP method would produce a business process-based digital game with potential to training a business process in this organization in particular, we concentrated our focus on the assessment with the process manager. Since we could not, in his research cycle, guarantee yet that the game was suitable for training, we did not evaluate the game with the process actors.

Qualitative research was conducted with the process manager and had the following objective [2]: O1) Analyzing the digital game Mediator Game; to evaluate the perception of usability, game experience and learning process according to the MEEGA+ (Model for the Evaluation of Educational Games) evaluation

model [19]; from the process manager (player) point of view; in the context of conflict mediation. The game was evaluated for its training potential qualitatively and in the expectation of the business specialist who knows the challenges of training the process in the organization.

This study was conducted in May 2021. The research participant (player) is a man over 50s. He does not often play digital games. He has experience in Process Management and related projects in the Judiciary in Conflict Mediation concerning Alternative Means of Conflict Resolution (Meios Alternativos de Resolução de Conflitos in portuguese). The study was based on the evaluation of the game through the measurement instrument (questionnaire) of the MEEGA+ model [19] adapted to include the learning evaluation of the process implemented in the Mediator Game.

The form available in the MEEGA+ method was used, containing 35 fixed items (33 of player experience and 2 of short-term learning) and 6 exclusive questions to verify the game’s learning objectives, totaling 41 questions. The learning objectives considered for digital games based on business processes for structured process training in the MEEGA+ model are presented in Table 2.

Table 2. Learning perception items

	Dimension	Code	Description
Learning perception	Short-term learning	ACP1	The game contributed to my learning about the mediator selection process
		ACP2	The game was efficient for my learning compared to other sources of information (CEJUSC website)
	Learning goals	OBA1	The game contributed to my learning about what to do when the process starts
		OBA2	The game contributed to my learning about when the process ends.
		OBA3	The game contributed to understanding the sequence of necessary activities to execute the process
		OBA4	The game contributed to my learning about how the process activities end
		OBA5	The game contributed to the learning of the actors in the process
		OBA6	The game contributed to my learning about important decisions

The evaluation process considered the steps presented in Table 3. The execution of the Mediator Game evaluation study took place online. For the application of the experiment, the Free and Informed Consent Term (ICF) was made available. The questionnaire was made available by e-mail so that the process manager could answer it.

The analysis and interpretation of results werre conducted as suggested by the MEEGA+ model: usability, player experience and learning process. Questions with a rating above 0 (neutral) were considered positive perceptions. The player experience rating was generally positive across the board. Regarding usability aspects, the game obtained a result of disagreement for aesthetics (Texts, colors and fonts match and are consistent), learnability (I needed to learn a few things to be able to start playing the game.), operability (The rules

Table 3. Game evaluation steps

Step	Description	Time
Training	The participant receives the basic training (video or supervised play) highlighting game rules and gameplay	8 min
Game execution	Participants play the Mediator Game in one or more matches. Each match has about 10 to 20 min	10 to 25 min
Evaluation questionnaire	Participants must answer the evaluation questionnaire after playing the Mediator Game	5 to 20 min

of the game are clear and understandable). The player demonstrated neutrality for the dimensions and satisfaction (Learning to play this game was easy for me). Social interaction (I was so involved in the game that I lost track of time) and relevance (I'd rather learn from this game than an otherwise (another method). Short-term learning aspects and learning objectives were evaluated regarding the perceived learning quality factor results. The items were evaluated as positive, obtaining a result of an agreement for all items. The sequence of actions within the activities necessary for the mediation process was surprising for the specialist. Besides, it allowed remembering the older games and the perception of how to mediate conflicts. It is understood that the player's perception of the Mediator Game was, in general, positive and may indicate a good acceptance of the game as a support tool for training organizational processes. However, for future applications of the questionnaire that aim to use games to evaluate training processes, a written explanation is recommended and submitted to the player better to evaluate aspects of player experience such as usability.

5.6 Packaging

The game was published on the research group's website².

6 Limitations

In this investigation cycle, the objective was to propose the construction of BPBDG for training and to evaluate the effectiveness of the application of PYP in its current version without considering more in-depth training aspects, which will be discussed in the subsequent design cycle of the artifact. As a secondary artifact in our research, the Mediator Game is a quite simple game prototype, based on a process model with low complexity and its development was not conducted by professional game artists/designers. Another limitation of the results is that only one participant performed the evaluation, and it had to be so, because the process had only one manager. Although these results cannot be generalizable, they are sufficient as insights for the next design cycle. Finally, the game evaluation was conducted remotely imposed by the COVID-19 pandemic. An explanatory remote game session was performed about the rules and how to play the game to help the participant understand it and minimize obstacles while playing the game.

² <http://jocom.uniriotec.br/games/mediador/>.

7 Final Remarks

As research findings, from the insights produced with the evaluation, the first cycle of the research pointed out relevant issues for adapting the Play Your Process when explicitly applied to the development of games that offer training. For process training, we realized the importance of detailing the initial steps of the method to define the training/learning objectives in skills to be developed by the player, the construction of narratives based on process instances, mapping the narratives with the training/learning objectives, the evaluation of player learning and especially the impact of training on the organization.

The results of this research and its future steps intend to contribute scientifically to the BPM area, presenting a method that systematizes the construction of games for training processes in organizations. For the next steps of this research, a new cycle of the investigation will be conducted to adapt Play Your Process to meet the issues identified in this cycle, as well as its application in the design of a BPBDG for training a process in a partner company and the validation of the conjectures not evaluated in this cycle. We also hope to have the opportunity in the future to explore more processes and game designs in different contexts.

Acknowledgments. This research is partially funded by Mackpesquisa, CNPq (313210/2019-5) and FAPERJ (proc. E-26/210.231/2021 and proc. E-26/010.002458/2019).

References

1. Back, G., Daniel, K.: Process training to support change necessary within the scope of process implementation. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 48–61. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23471-2_4
2. Basili, V.R.: Software modeling and measurement: the goal/question/metric paradigm. Technical report, University of Maryland (1992)
3. Classe, T., Araujo, R.M., Xexéo, G.B.: Process model game design: Uma ferramenta para apoio a sistematização de design de jogos digitais baseados em processos de negócio. English title: Process Model Game Design: A Tool to Support the Systematization of Digital Games Based on Business Process). In: XVII Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames 2018) (2018)
4. De Classe, T.M., De Araujo, R.M., Xexéo, G.B., Siqueira, S.: The play your process method for business process-based digital game design. *Int. J. Serious Games* **6**(1), 27–48 (2019)
5. Deterding, S., Sicart, M., Nacke, L., O'Hara, K., Dixon, D.: Gamification. Using game-design elements in non-gaming contexts. In: CHI 2011 extended abstracts on human factors in computing systems, pp. 2425–2428 (2011)
6. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*, 2nd edn. Springer, Heidelberg (2018)
7. Hevner, A.R.: A three cycle view of design science research. *Scand. J. Inf. Syst.* **19**(2), 4 (2007)

8. Hira, W.K., Marinho, M.V.P., Pereira, F.B., Barboza Jr., A.: Creating a conceptual model for game design documentation. In: XV Brazilian Symposium on Games and Digital Entertainment, pp. 329–336 (2016)
9. Huizinga, J.: *Homo ludens* 86, vol. 3. Routledge, Oxon (2014)
10. Kapustina, L.V., Martynova, I.A.: Training employees in the digital economy with the use of video games. In: Ashmarina, S., Mesquita, A., Vochozka, M. (eds.) *Digital Transformation of the Economy: Challenges, Trends and New Opportunities*. AISC, vol. 908, pp. 444–454. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-11367-4_44
11. Kirkpatrick, D., Kirkpatrick, J.: *Evaluating Training Programs: The Four Levels*. Berrett-Koehler Publishers (2006)
12. Lainema, T., Makkonen, P.: Applying constructivist approach to educational business games: case REALGAME. *Simul. Gaming* **34**(1), 131–149 (2003)
13. Leitão, T.M., Navarro, L.L.L., Cameira, R.F., Silva, E.R.: Serious games in business process management: a systematic literature review. *Bus. Process Manag. J.* (2021)
14. Lopes, T.N., Araujo, R.: Um mapeamento sistemático da literatura sobre aplicação de jogos digitais no treinamento de processos organizacionais. *iSys-Braz. J. Inf. Syst.* **14**(2), 96–125 (2021)
15. McGonigal, J.: *Reality is Broken: Why Games Make us Better and How they Can Change the World*. Penguin (2011)
16. Mendes, R., Classe, T., Siqueira, S., Xexéo, G.: Public processes are open for play. *Digital Government: Research and Practice* (2021)
17. Moller, C., Hansen, P.K.: Business process innovation: the lego case. In: 2006 IEEE International Technology Management Conference (ICE), pp. 1–8. IEEE (2006)
18. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **24**(3), 45–77 (2007)
19. Petri, G., von Wangenheim, C.G., Borgatto, A.F.: Evolution of a game evaluation model for computer teaching. In: *Proceedings of the XXV Workshop on Computer Education*. SBC (2017)
20. Petridis, P., et al.: State of the art in business games. *Int. J. Serious Games* **2**(1) (2015)
21. Rogers, D.L.: *Digital transformation: rethinking your business for the digital age*. Autêntica Business (2017)
22. Rosenthal, K., Strecker, S.: Business process modelling as serious game: findings from a field study. *Research Papers* (2018)
23. Santorum, M.: A serious game based method for business process management. In: 2011 Fifth International Conference on Research Challenges in Information Science, pp. 1–12. IEEE (2011)
24. Schell, J.: *Tenth Anniversary: The Art of Game Design*, 3rd edn. A K Peters/CRC Press, New York (2019)
25. Susi, T., Johannesson, M., Backlund, P.: *Serious games: an overview* (2007)
26. Von Ahn, L., Dabbish, L.: Designing games with a purpose. *Commun. ACM* **51**(8), 58–67 (2008)
27. Weske, M.: *Business Process Management: Concepts, Languages, Architectures*, 3rd edn. Springer, Heidelberg (2019)



On Current Job Market Demands for Process Mining: A Descriptive Analysis of LinkedIn Vacancies

Simin Maleki Shamasbi¹  , Amy Van Looy¹ , Barbara Weber² ,
and Maximilian Röglinger³ 

¹ Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2,
9000 Ghent, Belgium

{simin.maleki, amy.vanlooy}@ugent.be

² Institute of Computer Science, University of St. Gallen, Rosenbergstrasse 30, 9000 St. Gallen,
Switzerland

barbara.weber@unisg.ch

³ FIM Research Center, University of Bayreuth, Wittelsbacherring 10, 95444 Bayreuth,
Germany

maximilian.roeglinger@fim-rc.de

Abstract. Process mining is growing to a billion-dollar market, focusing on dedicated techniques for improving existing business processes. With the increasing popularity and application of process mining, most scholars have focused on technical research while the organizational and people-related aspects remain under-investigated. To partly fill the gap, this paper explores current job market demands in process mining by means of an empirical and analytical study of vacancies on LinkedIn platform. Our dataset uncovers a wide variety of vacancies from 47 countries, including organizations of different sizes and 12 sectors. The vacancies are issued by end-user companies, vendors or consultancy firms and include a combination of technical or business orientation. Given this wide variety among process mining vacancies, future research can also benefit from better complying with companies' needs.

Keywords: Process mining · Vacancy · Job advertisement · Skills · Human resource · Descriptive analysis

1 Introduction

Process mining with an estimated market growth of \$1 billion in 2022 [4], is considered a suitable technology to enable people, processes, services, channels, business models and operating technologies. The popularity of process mining applications is growing in both industry and academia [16]. Therefore, more calls for research on its application in organizations have been issued in recent years [8, 13–15]. Concurrently, the number of software companies providing process mining capabilities and consultancy firms that offer such services is increasing. There are over 40 commercial process mining tools

worldwide [17] among which Gartner monitors a few of them regularly. While process mining is getting more complex, the implementation is also getting more complicated across different domains [1]; Hence, organizations are allocating more resources and are hiring specialized employees, contracting consultants, or training existing analysts and managers to adopt this technology within the boundary of their improvement initiatives. Although process mining as an innovative IT tool can be a great enabler for organizations, getting the right people on board is a key success factor [11]. In this regard, it is crucial for organizations to be aware of the individual competencies and specialties they would require for succeeding in their process mining journeys.

Some organizations are assigning specific users to work with process mining tools. Given that a single business process can involve several departments, their corresponding managers and process participants are also involved [15]. Thus, process mining implementations can be highly challenging, and it makes human resources an essential factor that should be taken into consideration. At the same time, new technology and tools in process mining require new skills, roles, responsibilities and education of a new generation of experts with new competencies and a thorough understanding of computer science, data and IT [12]. In this regard, it is beneficial for organizations to be aware of the job market and the required skills and job positions by other organizations. Such information will help them plan for an appropriate job definition in process mining practices and know the variety of roles and skills in the market in order to plan for their best team composition. Process mining adoption in organizations and the reason for its failure/success are still under-researched, yet crucial [14]. Two factors that are required to successfully leverage the potential benefit of process mining are team composition and skills. Despite their importance and the high priority of this area, an open issue exists on the subject [8]. In this regard, we argue that gaining insight into what job demands process mining is currently sought by organizations will shed light on this gap. To the best of our knowledge, this area has not been specifically investigated yet. We argue that vacancies reflect the current needs of organizations and make a link between needs and demands.

Hence, our research question is:

– **RQ. What are the current job market demands for process mining?**

In this regard, we use vacancies as a proxy to reflect upon those demands for process mining practices in organizations.

To address this question, we developed the following objectives:

- (O1) To report the geographical distribution of vacancies
- (O2) To report the vacancies based on the size and sector of organizations
- (O3) To refine the job titles in vacancies and report what ‘job types’ and ‘job titles’ are sought by different types of organizations and to what extent they are technical or business-oriented

In order to accomplish our objectives, we followed an empirical and analytical approach. We extracted all vacancies related to process mining from LinkedIn jobs in May 2022. In this regard, we considered process mining vacancies from all countries in the

world without considering the original language used in the vacancy. The final dataset had 921 vacancies, after removing the duplicates and unrelated items regarding our inclusion and exclusion criteria. We performed a descriptive analysis to unfold the knowledge behind the dataset and address the research objectives.

The remainder of the article is structured as follows. In Sect. 2 the related studies are discussed. Section 3 presents the research design for this study. In Sect. 4, we explain and demonstrate the results of statistical analysis. In Sect. 5, we discuss our findings and their implications, our research limitations, and suggestions for future work. Finally, the paper is concluded in Sect. 6.

2 Literature Background

The popularity of process mining applications in the industry is growing in both industry and academia. Meanwhile, the integration of process mining with machine learning, simulation and other complementary trends, such as digital twins of an organization, is gaining significant attention in recent years [16]. In most cases, organizations adopt process mining to visualize, analyze and improve business processes that shape information systems [5]. Nevertheless, process mining has moved into wider areas in recent years and Gartner identified ten capabilities such as predictive and prescriptive analysis, customer interaction support, and task mining [4], just to name a few. Such capabilities provide the opportunity for the application of process mining in digital transformation journeys, RPA, hyperautomation, artificial intelligence usage or operational resilience.

Since the inception of process mining in the early 2000s, researchers have mainly focused on the development of technical - rather than organizational aspects [16]. While the focus of research has been on the technological aspect of process mining and the number of organizations that adopt process mining is increasing, more organizational and managerial research is required. In this regard, calls for research on the application of process mining has been issued in recent year. Studies have tried to emphasize the importance of considering this perspective and the existing gaps [8, 14]. Based on a Delphi study on the existing challenges of process mining, four challenges were identified as directly-related to people competencies and data/process orientation of the organization culture [8]. Van der Aalst [18] was one of the first ones to discuss process mining from the perspective of applications in industrial practice.

In fact, the technical view of academics has come a long way, but people- and culture-related aspects present very prominent challenges [8]. People play a key role to put process mining into action, which implies that it is crucial to select the right staff to deliver these kinds of projects [11]. Moreover, even if the bases of models are built automatically using process mining insights, human domain knowledge is still crucial [9]. These statements represent the Importance of people playing role in process mining practices and their skills. Furthermore, the notion of Augmented Business Process Management Systems (ABPMSs) is going to introduce a new class of process-aware information systems [2]. In this regard, each level of the Augmented BPM pyramid requires specific capabilities and skills on the organizational side. Therefore, considering the competencies of people who are performing process mining in an organization directly affects their success and progress in this pyramid.

Different job positions are dealing with process mining in organizations such as process analysts, process participants, process stakeholders, and external partners [5]. In this regard, online job portals like LinkedIn and monster websites have been considered appropriate tools for understanding the demographics of BPM professionals, their skills and the job positions in BPM projects [6, 7]. Notwithstanding the emphasis on organizational and managerial perspectives, to the best of our knowledge, there has been not a specific study on job positions and skillsets that are applicable in process mining practices in organizations.

3 Research Design

Our research consists of three main steps as depicted in Fig. 1. At first, we dealt with data collection and created a vacancies dataset. Then, we performed several data cleaning and preparation steps. We also derived more columns based on the existing columns in the dataset. Third, we applied descriptive statistics to derive insights from the cleaned dataset.

To achieve our research objectives, we applied the search terms of Table 1 to all LinkedIn jobs, which was the leading website that publishes vacancies [10], and downloaded them in an Excel worksheet. The search was conducted in May 2022. To analyze vacancies that were directly related to end-users and process mining in practice, we defined inclusion and exclusion criteria (Table 1). Furthermore, in order to take benefit from all vacancies around the world, we used an auto-translation library to translate non-English vacancies into English. In sum, 656 vacancies were in English, whereas the rest were in 12 different languages (e.g. German, Dutch, French, etc.). Subsequently, we describe the steps applied to this study.

3.1 Data Collection

To obtain the dataset, according to the inclusion criteria in Table 1, we developed a web crawler using the Selenium library of Python that could automatically go through all search queries related to process mining vacancies in different countries. Regarding the limitation of LinkedIn on showing job search results with a maximum of 1000 records, we had to include the search query for every country within our web crawler because searching for “process mining” in “worldwide” would return more than 3000 results. All vacancies were at the end extracted into an Excel worksheet. During the scraping, the language was automatically detected by GoogleTrans library and if it was a language rather than English, the job title and job description columns were automatically translated into English and saved in separated columns in the Excel file. Meanwhile, a field was also automatically generated for each record based on a list of process mining vendors in order to identify which organization is a vendor. After running the web crawler twice within a period of ten days in May 2022, we obtained a dataset containing 5,460 records to start the second phase of data preparation.

3.2 Data Preparation

Based on the exclusion criteria (Table 1), we manually identified not related records and labeled them as zero in the corresponding column to be excluded in our analysis. This goal was fulfilled by job title and if needed, job description (i.e., which contained job responsibilities and required skills).

Meanwhile, there was an unintegration in the sector column as the sector name was not following a specific rule. We also found that some sectors were not correctly defined. Therefore, we used uniform NACE codes [3] to redefine the sector column. Meanwhile, the organization size was missing in some records. This issue was manually resolved by looking at the company profile on LinkedIn or the company website.

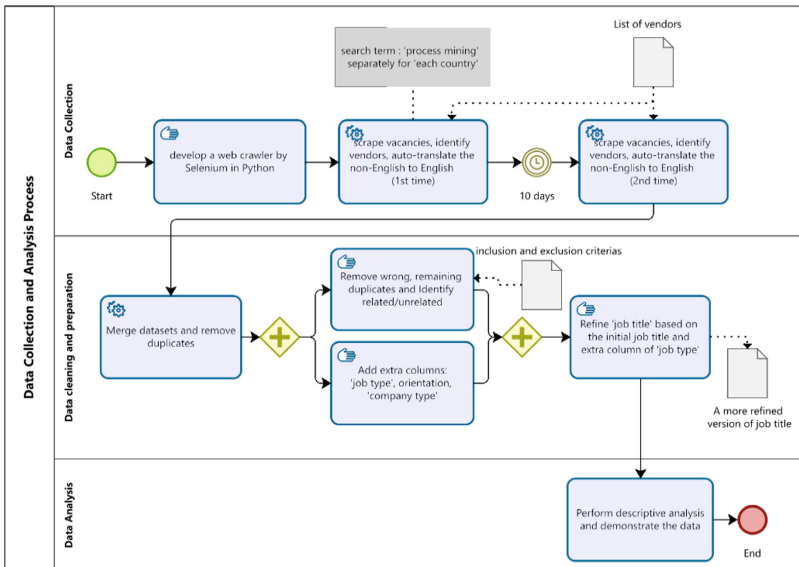


Fig. 1. Research process

Our dataset initially revealed 838 different job titles among 921 records, and we observed that job titles in vacancies contained multiple extra information (e.g., gender, city, etc.). After some basic cleanings and removing extra parts in the title (including the seniority), the number was reduced to 740 different job titles. This number was still too many to extract categorized information about job types and titles. In this regard, we added two new columns in the dataset as follows: ‘Job Types’ to reflect the job type/level of the prospect employee (e.g., manager, analyst, consultant, etc.) and ‘Job section’ as the subject of the vacancy like process, business, process mining, etc. The former was filled with deep consideration and attention to the initial job title and if required, the job description column. Nevertheless, since we received a lot of variety in job types, we made decisions upon merging some similar job levels with each other. For example, we considered merging expert, specialist, expertise and officer as specialists.

The combination of job section and job type will create a new job title that is more clear and more homogenized than the initial job title.

3.3 Data Analysis

Our final dataset consisted of 921 vacancies in process mining worldwide. Based on the initial data columns that were available, extra columns were constructed specifically for this research. We applied descriptive analysis for demonstrating different aspects of the dataset, in order to address our objectives and the research question.

Table 1. Research protocol

Search term	‘process mining’ separately for ‘each country’ in the world in ‘any languages’ on LinkedIn Jobs page
Inclusion criteria	All vacancies from end-users/consultancy firms that require either responsibilities or skills related to process mining implementation or both; without considering the percentage of their engagement
	All vacancies from end-users that make benefit from process mining results but are not really involved in process mining implementation
	All vacancies from vendors that have direct contact with customers for implementation, technical support or consultancy purposes
Exclusion criteria	Vacancies from end-users that do not clearly mention any skills or responsibilities related to process mining; although they might indicate that the company has process mining
	Vacancies from end-users planning to implement process mining in the future but do not mention process mining as a requirement in the job description or skillset
	Vacancies from end-users that have process mining skills only as an extra point or bonus but have not mentioned related responsibilities
	Vacancies from vendors that are not directly involved in process mining practices in organizations e.g. marketing, sales, inventory, algorithm development, software production, finance, training, procurement, account management, etc
	Vacancies from research institutes/universities for researchers/students

4 Results

4.1 Geographical Distribution of Vacancies (O1)

In order to address O1, we now report on our findings regarding the distribution of vacancies in process mining. Figure 2 presents the geographical distribution of vacancies throughout the world. According to the dataset, 47 countries were seeking for employees

to participate in their process mining journeys. The map clearly shows that the majority of these vacancies were centralized in some regions across Central Europe and North America, while Germany and the United States possessed about 40% of all vacancies. Furthermore, our dataset shows that vacancies came from 438 different organizations and 12 different sectors. This variety supports us to argue that our dataset is a small yet appropriate sample of required job positions that could be generalized to vacancies for process mining practices across the world. Our data can thus be seen as a good representation of the vacancies at this particular moment in time.

4.2 Distribution of Vacancies Based on Company Type and Sector (O2)

To address O2, we identified three types of organizations that advertise for vacancies (Fig. 3), namely end users, vendors and consultancy firms. The first category contained 476 vacancies from organizations that hire employees either for their own process mining practices or for other organizations while playing the role of recruiters.

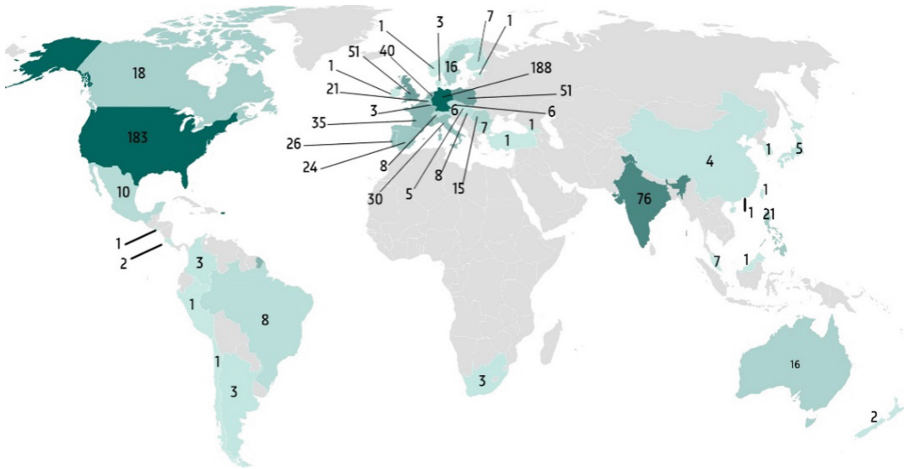


Fig. 2. Number of vacancies per country

The second category contained 82 vacancies from organizations providing process mining software applications and services as vendors. Among them, we found software companies that are offering process mining features within their existing services but also software companies that are empowering their teams to enter this rapidly-growing market. The third group was 363 vacancies from consultancy firms giving consultancy services to end-users for helping them to leverage process mining.

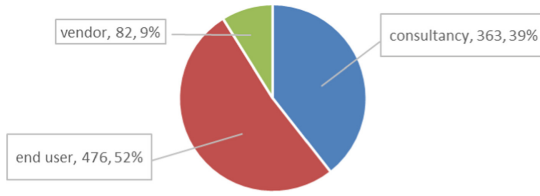


Fig. 3. Number of vacancies by organization type

Table 2 presents the size of the organizations and their sector. 43.4% of the sampled vacancies were related to organizations working in the sector of ‘Information and communications’, and more than 50% (486) of them came from large organizations with more than 10,001 employees. Furthermore, the size of organizations working in the sector of ‘Administrative and support service activities mostly contained organizations acting as recruiters. Therefore, it cannot present the size of the real employer and that size is not clear in those cases. Figure 4 shows the technical/business orientation of job types. Within each blue box, job types and the corresponding frequencies within that orientation are presented. Figure 5 demonstrate the job types regarding their frequency in either technical or business orientation or both.

Table 2. Vacancies by sector and size of the organizations (based on NACE)

Sector (NACE)	1–10	11–50	51–200	201–500	501–1,000	1,001–5,000	5,001–10,000	10,001 +	Total	%
Information and communication	8	7	33	24	14	88	30	196	400	43.4%
Manufacturing	2	2	1	2	1	20	23	133	184	20.0%
Professional, scientific and technical activities	4	2	7	1	4	40	3	72	133	14.4%
Administrative and support service activities	6	23	12	7	5	9	7	4	73	7.9%
Financial and insurance activities	4		1	2	3	8	5	43	66	7.2%
Human health and social work activities	1					1		18	20	2.2%
Wholesale and retail trade				2		3	1	10	16	1.7%
Transportation and storage			1		1	6	1	5	14	1.5%

(continued)

Table 2. (continued)

Sector (NACE)	1–10	11–50	51–200	201–500	501–1,000	1,001–5,000	5,001–10,000	10,001 +	Total	%
Construction			1			5	1	3	10	1.1%
Public administration and defence								2	2	0.2%
Education					1	1			2	0.2%
Other professional, scientific and technical activities						1			1	0.1%
Total	25	34	56	38	29	182	71	486	921	

4.3 Job Types, Job Titles and Technical/Business Orientation (O3)

In order to address O3, we identified three types of job categories based on the orientation of the job: technical, business, and both. Such orientation was identified based on the ‘job responsibilities’ and ‘required skill’ in the ‘job description’.

The distribution of these orientations and their relevant job types are presented in (Fig. 4). We identified 12 different job types. Figure 5 demonstrates these types based on the orientation to which they are related. The number on the ‘data labels’ presents the number of vacancies in each orientation for a given job title. After cleaning the job title as described in Sect. 3.2, the frequency of job titles was reduced to 252 items of which more than 50% have a frequency of below ten and 14% have a frequency of one. Because of the long list of output, we show only the first 20 results with a frequency of at least 10 (Table 3).

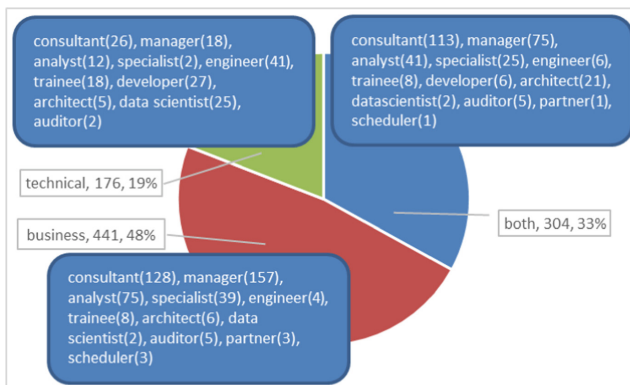


Fig. 4. Stat of Job Orientations and the frequency of job types in each section

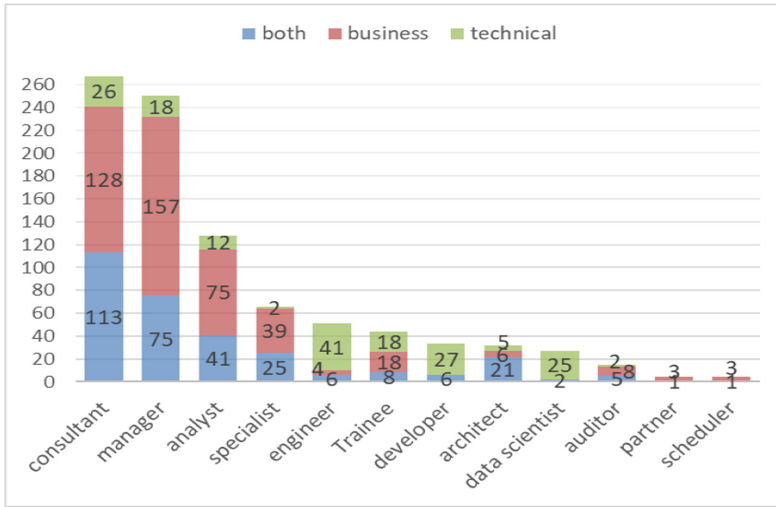


Fig. 5. The identified job types and their frequencies based on the job orientation

Table 3. The frequency and orientation of the job titles

Row labels	Business	Technical	Both	Total
Process mining consultant	15	4	39	58
Process manager	31		2	33
Business analyst	25		7	32
Process mining manager	7	2	19	28
Data scientist		25	2	27
Process analyst	22		3	25
Data analyst	6	8	9	23
Data engineer		21	1	22
Business transformation consultant	9		8	17
Process mining specialist	3		13	16
Intelligent automation manager	6	1	9	16
Customer value manager	15			15
Process mining analyst	2	2	10	14
Process consultant	12		1	13
RPA developer		12	1	13
Process mining architect			12	12
Process excellence manager	11			11

(continued)

Table 3. (continued)

Row labels	Business	Technical	Both	Total
Intelligent automation consultant	5	1	5	11
RPA consultant		3	7	10
Process mining trainee	1	5	4	10

5 Discussion

The results could address our objectives, answer the research question and present an understanding of job market demands and vacancies in process mining. In the following sub-sections, we discuss and present the implications of our study (Sect. 5.1) and acknowledge the limitations of this research and propose some avenues for future works (Sect. 5.2).

5.1 Implications

Our findings have some implications. First, end users can get a basic understanding of different vacancies and roles to consider in their process mining implementation. Second, the technical/business orientation is also an interesting outcome of our study. Given that process mining is connecting data science with process science, there are multiple job types and job titles that need a hybrid orientation. This shows that technical versus business orientations in process mining job titles have their own dedicated specifications and for some job types and job titles, the orientation differs. Indeed, process mining has caused new job definitions to be created in the field of BPM and Information Technology. Jobs such as process mining consultant, process mining analyst, process mining manager, etc. are some examples. Meanwhile, changes in job requirements of process analysts, process specialists, data analysts, etc. show the impact of this technology on the definition of such roles for process mining practices. Third, our result is helpful for job seekers to know what types of job opportunities exist in this market with technical, business or technical-business orientation.

Fourth, our study unveils the fact that there is no integrity in job titles in process mining projects. Therefore, almost all employers have their own language for defining job titles and job descriptions. Fifth, our study reveals the niche markets and newly growing demands while also unveiling the growing interest in software companies to adapt process mining techniques to their existing software applications and services. This finding provides insight for vendors and intermediaries to focus on expanding the market to other countries and also other sectors so as to empower the application of process mining in other areas. Simultaneously, our findings provide insight into improving their competitive advantage and letting their business grow in competition with multiple process mining providers. Last but not least, our approach and the methodology we proposed could be repeatable in other BPM-related roles.

5.2 Research Limitations and Future Work

Despite our in-depth descriptive analysis of process mining vacancies in all countries without considering the language, our research still faces some limitations. First, our dataset is limited to vacancies in May 2022. Although the extraction was performed twice within 10 days, it is a sample of the real world at a certain moment in time. Therefore, evolution over time and seasonal changes are not considered. Nevertheless, a longitudinal dataset containing vacancies across several months or different seasons will be beneficial. It will also give the opportunity for chronological research which compares the change and evolution of vacancies and job types during the time. Second, we extracted data from LinkedIn, which is yet the most popular platform for process mining vacancies. Other platforms like Monster.com or local job portals will also give a deeper insight into vacancies in certain geographical areas in the case of the latter. Third, we tried to homogenize job titles by considering the original job title and job description. This objective can only be fully explored by performing further research through skills extraction, text mining and/or conducting an expert panel. Fourth, there might be a potential bias regarding our search terms due to LinkedIn algorithms, although our query terms were searched in all vacancies' metadata.

For our next step and as future research, we are going to extend this study further on standardizing job roles and deriving individual skills and competencies that are required within process mining projects in organizations. Further, there might be a novel opportunity to investigate mapping such competencies with the levels of Augmented BPM pyramid [2] or a pre-defined maturity model.

6 Conclusion

Although process mining has matured as a research field over the past decade from a technical standpoint, there is a limited understanding of process mining from an organizational perspective [8]. In this study, we have conducted empirical and analytical research on vacancies in process mining practices throughout the world. Based on Sect. 2, we could argue that there has been no study on this topic so far and it remains unclear what kind of job types, job positions and skills with what kind of orientation exists or is required in process mining practices. Since vacancies are considered an appropriate source of data for studying job specifications, we argue that they can reflect the current needs of organizations by linking needs and demands. We studied 921 vacancies and tried to make clear what the state of job market demand for process mining at the moment in time is. Novelty in this study lies in considering real data from a well-known job portal and focusing on geographical distribution, employer specifications, job titles and their characteristics, presenting facts underlying vacancies and the job market.

References

1. Diba, K.: Towards a comprehensive methodology for process mining. In: Proceedings of the 11th Central European Workshop on Services and their Composition, Bayreuth, pp. 9–12 (2019)

2. Dumas, M., et al.: AI-augmented business process management systems: a research manifesto. *ACM Trans. Manage. Inf. Syst.* (2023). <https://doi.org/10.1145/3576047>
3. Eurostat, NACE Rev.2 (2015). <https://ec.europa.eu/eurostat/web/products-manuals-and-guidelines/-/ks-ra-07-015>
4. Gartner, Market Guid for Process Mining (2021)
5. 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>
6. Lohmann, P., Zur Muehlen, M.: Business process management skills and roles: an investigation of the demand and supply side of BPM professionals. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) *BPM 2015. LNCS*, vol. 9253, pp. 317–332. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_22
7. Lohmann, P., zur Muehlen, M.: Regulatory instability, business process management technology, and BPM skill configurations. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) *BPM 2019. LNCS*, vol. 11675, pp. 419–435. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_27
8. Martin, N., et al.: Opportunities and challenges for process mining in organizations: results of a delphi study. *Bus. Inf. Syst. Eng.* **63**(5), 511–527 (2021). <https://doi.org/10.1007/s12599-021-00720-0>
9. Pourbafrani, M., van der Aalst, W.M.P.: Hybrid business process simulation: updating detailed process simulation models using high-level simulations. In: Guizzardi, R., Ralyté, J., Franch, X. (eds.) *Research Challenges in Information Science. RCIS 2022. Lecture Notes in Business Information Processing*, vol. 446, pp. 177–194. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05760-1_11
10. Skeels, M.M., Grudin, J.: When social networks cross boundaries: a case study of workplace use of Facebook and LinkedIn. In: *Proceedings of the ACM 2009 International Conference on Supporting Group Work* (2009). <https://doi.org/10.1145/1531674.1531689>
11. Reinkemeyer, L.: *Process Mining in Action: Principles Use Cases and Outlook*. Springer, Cham (2020). <https://doi.org/10.1007/978-3-030-40172-6>
12. Reinkemeyer, L.: Status and future of process mining: from process discovery to process execution. In: van der Aalst, W.M.P., Carmona, J. (eds.) *Process Mining Handbook*, pp. 405–415. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_13
13. Thiede, M., et al.: How is process mining technology used by organizations? a systematic literature review of empirical studies. *Bus. Process. Manag. J.* **24**(4), 900–922 (2018). <https://doi.org/10.1108/BPMJ-06-2017-0148>
14. 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>
15. 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>
16. van Cruchten, R., Weigand, H.: Towards event log management for process mining - vision and research challenges. In: Guizzardi, R., Ralyté, J., Franch, X. (eds.) *Research Challenges in Information Science, RCIS 2022*, pp. 197–213. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05760-1_12
17. van der Aalst, W.M.P.: Process mining: a 360 degree overview. In: van der Aalst, W.M.P., Carmona, J. (eds.) *Process Mining Handbook. Lecture Notes in Business Information Processing*, vol. 448, pp. 3–34. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_1
18. van der Aalst, W.M.P., et al.: Business process mining: an industrial application. *Inf. Syst.* **32**(5), 713–732 (2007). <https://doi.org/10.1016/j.is.2006.05.003>

**1st International Workshop on Data-
Driven Business Process Optimization
(BPO 2022)**

1st International Workshop on Data-Driven Business Process Optimization (BPO 2022)

Remco Dijkman¹, Arik Senderovich², and Willem van Jaarsveld¹

¹ Eindhoven University of Technology, The Netherlands
{r.m.dijkman, w.l.v.jaarsveld}@tue.nl

² York University, Canada
sariks@yorku.ca

Preface

Business process management is a very promising paradigm for optimizing the way in which work is performed in an organization. Decisions that are always implicit in business processes, if they are described using traditional business process modelling techniques, include: assigning resources to the tasks for which they are most suited, deciding on the execution order of tasks to meet customer deadlines, and deciding on the overall design of the process to minimize the overall processing time of customer cases. While such questions are important in administrative processes, they are even more important in processes that have a physical component, such as transport processes, production processes, and clinical pathways, in which assigning tasks to the wrong resource, or performing them in the wrong order immediately leads to higher costs, dissatisfied customers or even health risks.

Traditionally, the research area of operations research has studied techniques for modeling and solving optimization problems in much detail. At the same time, the research area of business process management has studied techniques for aggregating the data that is needed for modeling, analyzing, and in the end optimizing business processes. Combining techniques from both areas makes it possible to solve optimization problems from practice, using models that are based on real-world data, with fewer assumptions. In particular, it allows us to create clear and realistic data-driven models of the way in which customer orders pass through the organization and of the behavior and performance of resources. While this provides clear benefits in terms of more realistic models and analysis, it also brings challenges in terms of the computational complexity of the used analysis and optimization techniques.

The goal of the Data-Driven Business Process Optimization workshop was to bring together researchers from both the area of Business Process Management and the area of Operations Research as well as other related areas, with the overall goal of developing techniques for optimizing business processes in an organization based on models that are created from real-world data.

The workshop covered both presentations on techniques for optimizing business processes and applications of such techniques to real-world problems. It received five paper submissions, out of which two were accepted for presentation. In addition, the

workshop also accepted abstracts for presentation, of which it received three, which were all accepted.

Matteo Di Cunzolo, Alberto Guastalla, Roberto Aringhieri, Emilio Sulis, Ilaria Angela Amantea, Massimiliano Ronzani, Chiara Di Francescomarino, and Chiara Ghidini presented the paper titled ‘Combining Process Mining and Optimization: A Scheduling Application in Healthcare’, which described a technique for operating room scheduling that combined scheduling techniques from operations research with process analysis techniques to create improved schedules.

Irene Bedilia Estrada Torres, Adela Del Río Ortega, and Manuel Resinas presented the paper titled ‘Defining Process Performance Measures in an Object-Centric Context’, which showed how process performance indicators should be defined in object-centric business processes.

Opher Baron, Dmitry Krass, Arik Senderovich, and Sijia Li presented the abstract titled ‘Mining Hybrid Machine Learning and Simulation Models for Healthcare Applications’, which proposed a technique for simulating processes that integrated machine learning to facilitate simulation of processes in the light of limited data availability.

Riccardo Lo Bianco presented the abstract titled ‘Solving the Vehicle Routing Problem with Order Lifecycle with Deep Reinforcement Learning’, which proposed a technique for embedding the process perspective into operations research problems and showed that it can lead to improved solutions of those problems.

Roberto Aringhieri, Stefano Branchi, Chiara Di Francescomarino, Chiara Ghidini, Alberto Guastalla, and Emilio Sulis presented the abstract titled ‘Process Mining Meets Optimization: The Case of the Workshifts Scheduling in Healthcare’, which demonstrated how workshift scheduling in hospitals can be improved by combining it with process analysis techniques.

We hope that the reader found the selection of papers and abstracts useful to get an insight into how operations research and business process management can be combined to solve business process optimization problems.

Organization

Organizing Committee

Arik Senderovich	Rotman School of Management, University of Toronto, Canada
Remco Dijkman	Eindhoven University of Technology, The Netherlands
Willem van Jaarsveld	Eindhoven University of Technology, The Netherlands

Program Committee

Akhil Kumar	Penn State University, USA
Anna Kalenkova	University of Adelaide, Australia
Avigdor Gal	Technion, Israel
Cristina Cabanillas	Universidad de Seville, Spain
Fernanda Gonzalez-Lopez	Pontificia Universidad Católica de Chile, Chile
Izack Cohen	Bar-Ilan University, Israel
Marcos Sepulveda	Pontificia Universidad Católica de Chile, Chile
Martin Matzner	Friedrich Alexander Universität, Germany
Massimiliano de Leoni	University of Padova, Italy
Matthias Weidlich	Humboldt University, Germany
Orlenys López Pintado	University of Tartu, Estonia
Robert Boute	Vlerick Business School, KU Leuven, Belgium
Ton de Kok	Center for Mathematics and Computer Science, The Netherlands
Yaron Shaposhnik	University of Rochester, USA
Yoav Kerner	Ben Gurion University, Israel



Combining Process Mining and Optimization: A Scheduling Application in Healthcare

Matteo Di Cunzolo¹, Alberto Guastalla¹, Roberto Aringhieri¹, Emilio Sulis¹,
Ilaria Angela Amantea¹, Massimiliano Ronzani²(✉),
Chiara Di Francescomarino², Chiara Ghidini², Paolo Fonio^{3,4},
and Marco Grosso^{3,4}

¹ Computer Science Department of the University of Torino, Corso Svizzera 185,
10149 Torino, Italy

² Fondazione Bruno Kessler, Via Sommarive 18, 38123 Povo, TN, Italy
mronzani@fbk.eu

³ Department of Surgical Sciences, Radiology Unit, University of Torino,
Via Genova 3, 10126 Torino, Italy

⁴ Radiology Department, A.O.U. City of Health and Science, Corso Bramante, 88,
10126 Torino, Italy

Abstract. Optimizing the scheduling of operating rooms is quite a challenging task, as different aspects, some of which the medical personnel is not completely aware of, may have a strong impact on the scheduling and need to be taken into account. This work aims at addressing such a problem by proposing a framework that combines process analysis and operations research. Process mining techniques are used for analysing interventional radiology data collected from the information system of a hospital and identifying delays and lagging cases, as well as the causes of these delays. Leveraging the knowledge acquired by looking at data (e.g., the procedures that are more often delayed), an optimization model able to take into account these aspects is designed. This paper describes the preliminary results of a proof-of-concept based on 3 months real-life data. The results show that, taking into account the information discovered from data, allows for obtaining a more accurate scheduling.

Keywords: Scheduling · Optimization · Process mining · Radiology

1 Introduction

Scheduling optimization in complex scenarios such as the operating rooms of big hospitals is a quite challenging task. These kinds of scenarios indeed often involve a number of medical and support teams, of operating rooms, as well as of different wards. Being able to have a complete picture of all these aspects is not easy—even for the medical personnel involved in the organizational process. One possibility to fill this gap is to extract from data information that otherwise

would be lost in these complex settings. To this aim, in this paper we propose a pipeline combining process mining and optimization techniques. Optimization techniques have been widely applied in many areas to find solutions that maximize or minimize specific objectives. In healthcare management, one can think of minimizing costs in the provision of a medical good or service, as well as decreasing waiting times in a service access program. In the management of an organization, process-oriented research has recently seen a considerable growth with the possibility of using real data recorded in information systems [1]. The application of mining techniques to organizational processes makes it possible to derive event flow information from time-stamped events, and the prospects are also very favorable in healthcare [2].

We focus on a case study in healthcare in Interventional Radiology (IR), which is an increasingly used medical specialty relying on the possibilities offered by new technologies to perform minimally invasive interventions or procedures through very small incisions or body orifices. Compared with traditional methods, the advantage is to decrease risks, pain and recovery time. Real-time visualization also allows precise guidance to the abnormality.

We report a proof-of-concept of the proposed pipeline applied to the scheduling optimization of the IR operating rooms. Leveraging a real-life dataset we built a healthcare event log, and analyzed it in order to discover the main causes for delays and lagging cases. The discovered information—such as the IR procedures requiring more time—is then used for an optimized scheduling able to take into account these aspects.

In the following of the paper we introduce some related work and the case study in Sect. 2. The process mining and optimization approaches are detailed in Sects. 3 and 4, respectively. The computational analysis is reported in Sect. 5. We conclude the paper in Sect. 6.

2 Background

Related Work. Process Mining (PM) is a relatively recent discipline that focuses on extracting knowledge from data collected in enterprise information system databases [3]. Such data can be processed and organized into event logs, which is a structured way of collecting a set of traces, each of which contains the activities performed for any particular process instance [4]. Process models can be represented using different modeling paradigms ranging from procedural [3] to declarative [5] and hybrid languages [6].

Process mining techniques can be used to strengthen the development of online optimization algorithms. The management of processes in real time is extremely challenging to face uncertainty in planning and, by consequence, determining inefficiency in terms of outputs and outcomes [7]. Online optimization algorithms, which demonstrated their validity in the management of several health processes (e.g., operating room planning, emergency medical services) can hence be used to ensure the delivery of efficient and of quality health services [8]. Hybrid simulation optimization models can be proposed to replicate

the patient flow in a clinical pathway, to embed optimization modules to deal with healthcare decision problems.

Healthcare PM and optimization techniques have been successfully adopted for discovering patient pathways from hospital database [9], to derive process models from event logs [10]. Integer Linear Programming has been used to discover a Petri Net model [11], as well as for extracting meaningful paths and constructing simulations that reflect behavioral and operational aspects [12]. The combination of PM, optimization and simulation methods can be a basis for building decision support systems in healthcare [13].

Case Study. We focus on the Hospital Department of Diagnostic Imaging and Interventional Radiology of the City of Health and Science (CHS) of Torino¹, one of the main hospitals in Italy. Interventional radiology (IR), which has complemented and, especially in recent years, partially replaced traditional surgical techniques, is constantly evolving, supported by the availability of sophisticated materials. The purpose is to offer therapies defined as minimally invasive, which makes it possible to achieve a therapeutic goal with the minimum trauma for the patients. Typical interventions are radiological methods—computed tomography scan (CT), Magnetic Resonance Imaging (MRI), ultrasound. An example is interventional neuroradiology used for the diagnosis and treatment of diseases of the head, neck, and spine—which has proven to be of decisive importance in the prevention of ischemic stroke.

Procedures can be diagnostic or therapeutic. Diagnostic IR procedures, in addition to traditional image analysis, allow a diagnosis to be made or further medical treatment to be guided. In addition, therapeutic procedures provide direct treatment: they include administration of drugs via catheter, placement of medical devices (e.g., stents), and angioplasty of narrowed structures. In this paper, we use the term *procedure* for indicating the diagnostic or therapeutic intervention a patient undergoes and the term *service* for indicating the specific test or treatment a procedure is composed of. For instance, a procedure can be composed of two types of arteriography and a chemoembolization treatment. Procedures and services can also be classified as clean and dirty. Dirty procedures and services (including Covid-19 cases), indeed, are the ones requiring that the operating room is cleaned more in-depth.

Dataset. The initial dataset provided by the CHS hospital contains anonymized data related to around 100 patients who have undergone an IR procedure in the period December 2021 – February 2022. The dataset contains, for each procedure carried out, the patient id and the date of the procedure, personal information related to the patient (i.e., age and gender), the clinical question (in natural language) that the procedure aims at addressing, the specific services provided when carrying out the procedure, the operating room number and the hospital ward the patient comes from. Moreover, the dataset contains detailed temporal data related to the time in which (i) the patient is called from the ward (*Chiamata*); (ii) the patient entered (*Ingresso in blocco*) and exited (*Uscita blocco*) the

¹ <https://www.cittadellasalute.to.it/>.

IR department; (iii) the patient entered (*Ingresso in sala*) and exited (*Uscita sala*) the operating room; (iv) the anaesthesia is delivered (*Anestesia*), when needed; (v) the patient is ready for the operation after receiving the anaesthesia (*Paz pronto a fine induzione anestesia*); (vi) the procedure is started (*Inizio intervento*) and ended (*Fine intervento*); (vi) the operating room is restored (*Ripristino sala*), when needed.

3 Process Mining

In this section we describe the process mining analysis carried out in order to identify—from the raw data described in Sect. 2—the delayed procedure instances and the potential causes of these delays, that is the critical aspects that require more attention when scheduling the procedures.

3.1 Analysing the Procedure Process

In order to use PM techniques for the analysis of the IR procedures we generate a procedure event log starting from the raw data described in Sect. 2.

The traces in the procedure event log focus on the history of the procedure carried out by the patient on a certain date. Each procedure instance is hence identified by the `patient_id` and the `date` attributes (the same patient indeed can undergo different procedures in different days). The temporally annotated raw data attributes described in Sect. 2 (e.g., *Chiamata*, *Ingresso paziente in blocco*) act as the activities of the traces describing the history of the procedure instance. The traces are finally enriched with some trace attributes such as the `operating room number`, the `age` and the `gender` of the patient, the `clinical question` (in natural language) that the procedure aims at addressing, the list of the specific services provided in the procedure (`service list`), as well as the `hospital ward` (HW) attribute.

In order to identify the procedure instances requiring more time than the expected one, we resort to compare the *actual time* spent by a patient in the operating room against the time that the procedure should require according to an estimate made by the doctors (*standard time*). The procedure instances requiring more time than the *standard time* for that specific procedure are classified as *delayed* instances. Out of the 105 IR procedure instances analyzed, 68 are *delayed* instances, i.e., around 65%.

Figure 1 reports the directly-follows graph discovered by the tool Disco² for the *delayed* and the *in-time* procedure instances, in which the average time required by the procedure instances is analyzed. The figure shows an average higher duration of all the central activities (i.e. those carried out in the operating room) for *delayed* procedure instances.

² <https://www.fluxicon.com/>.

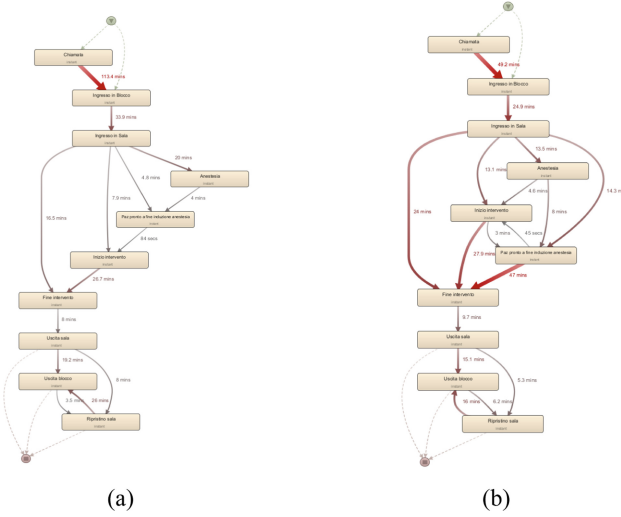


Fig. 1. Control flows and performance. (a) on time cases. (b) delayed cases.

3.2 Identifying the Causes of the Delay

With the aim of identifying potential causes for the delayed procedure instances, we compute the Fisher scores [14] so as to identify the most important attributes that allow for discriminating between delayed and in-time procedure instances. We first remove the sparse attributes from the set of the attributes of the procedure log. For instance, the `clinical question` attribute presents very low frequencies for all the values it can assume: out of 105 traces, the `clinical question` takes 82 different values; moreover, only 9 of these values appear more than once and only 4 of them appear more than twice. Once identified the set of attributes we are interested in, we compute the Fisher score for each of them. The Fisher score for the j -th attribute is computed as:

$$F_j = \frac{\sum_{i=1}^c n_i (\mu_{ji} - \mu_j)^2}{\sum_{i=1}^c n_i \sigma_{ji}^2}, \quad (1)$$

where n_i denotes the number of data points in class i (e.g., the class of the delayed procedure instances), μ_{ji} and σ_{ji}^2 denote mean and variance of class i corresponding to the j -th attribute, and μ_j is the mean of all data points corresponding to the j -th attribute. Figure 2 shows the Fisher score for each attribute. We can observe that the most important attributes discriminating between delayed and in-time procedure instances are the `hospital ward` (HW) and the `service list`, which identifies the specific procedure performed.

In the following we carry out the analysis of the dependence of the delay rate with respect to each of these two dimensions. For the most frequent values of each of the two attributes, we compute the average delay rate, that is the percentage of delayed instances with respect to all the instances with the same

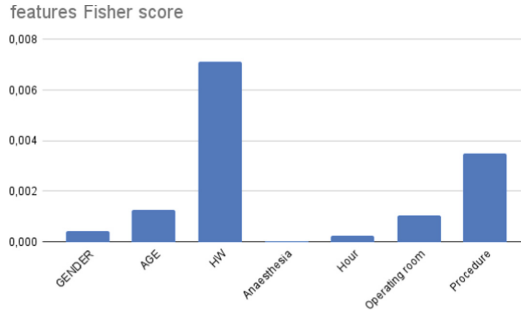


Fig. 2. Fisher score of the procedure log attributes. The most discriminative attributes are the hospital ward (HW) and the service list.

value. The most important attributes highlighted by the Fisher score values is the hospital ward (HW). Figure 3a displays the average delay rate for the five most frequent HWs (whose names have been anonymized), namely those appearing in at least five instances. We can observe that, for instance, the HW4 has an average delay rate equal to one, that means that all the procedure instances in the log corresponding to this HW (i.e., 10 cases) are delayed.

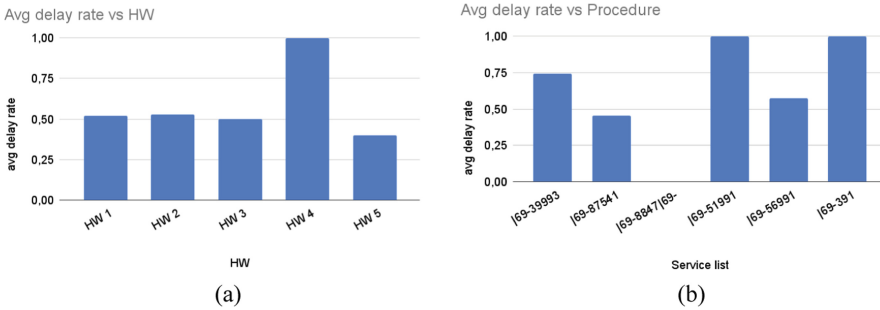


Fig. 3. (a) Average delay rate per hospital ward. (b) Average delay rate per service list (procedure).

The second most discriminative attribute highlighted by the Fisher score values is the list of services provided to the patient, that is the specific procedure carried out. Figure 3b shows the average delay rate for the six most common service lists, namely those appearing in at least five procedure instances. We can observe that the procedure performing the service 69-51991 and the procedure performing the service 69-391, which occur in 8 and 5 cases, respectively, have an average delay rate equal to one, namely in the considered data all instances related to those two procedures are delayed. Conversely, the procedure providing the three services 69-8847, 69-88495 and 69-99252, which occur in 8 instances

in our data, have an average delay rate equal to zero, that means that none of the cases related to that procedure was delayed.

4 An Optimization Model for the Interventional Radiology Process Scheduling Problem

In this section we describe an integer linear programming model to deal with the problem of selecting patients from the waiting list and to schedule them accordingly.

We consider the following operative context. We suppose to have four different operating rooms (ORs) denoted by S_1 , S_2 , S_3 and S_4 . The ORs have different equipment for different types of procedures: S_1 and S_2 are devoted for angioplasty while S_3 and S_4 are used for procedures requiring echography and/or computed axial tomography. These four ORs can operate in parallel in accordance with the work-shifts of the medical team. In this paper, we consider all the teams always available since this assumption does not affect the problem we are dealing with.

The patient flow is composed of inpatients and outpatients, and the inpatients come from the hospital wards. This means that the IR team is not in charge of the patient preparation before the procedure. Only a subset of procedures requires the presence of the anesthetist. The planning horizon is a week composed of five working days, from Monday to Friday.

We introduce the following notation. Let I , J and K be respectively the set of patients, specialties and ORs, which are indexed by i , j , k . We would remark that the set J is composed of two specialties, that is (i) the ones provided in S_1 and S_2 and (ii) the ones provided in S_3 and S_4 . Let I_j be the set of patients belonging to the specialty j . Let T be the set of the days belonging to the planning horizon, indexed by t . Let p_i and s_{kt} be respectively the duration of the procedure of the patient i , and the maximum time allowed for procedures in the OR k on day t . The binary parameter τ_{kt}^j is equal to 1 if the OR k is assigned to specialty j on day t , 0 otherwise. The binary parameter $u_{i\ell}$ is equal to 1 if the patient i should be scheduled before the patient ℓ , 0 otherwise. This parameter will be essential to input the main process mining finding into the optimization model. In particular, it will allow to model knowledge about patients' delay based on the findings made during the process mining phase. Further, it can be used to input the need of scheduling a Covid-19 patient as last procedure of the day.

Finally, the priority of the patient i is denoted with r_i , and computed as $100 \frac{L_i}{m_i}$, in which L_i is the waiting time of the patient i in days, and m_i is the maximum time (in days) before treatment of the patient i , i.e., the *ideal* time within which a patient should receive the treatment.

Before stating the optimization model, some decision variables should be introduced. The binary variable x_{ikt} is equal to 1 if the patient i is selected for a procedure and assigned to OR k on day t , 0 otherwise. The binary variable $y_{i\ell kt}$

is equal to 1 if the patient i is scheduled before the patient ℓ in the OR k on day t . Finally, the integer variable $\gamma_i \in \mathcal{Z}_+$ models the starting time (in minutes) of the procedure of the patient i . We assume that time starts from 0.

The objective of the optimization models is to maximise the OR utilisation, and, at the secondary level, to select the patients having higher priority r_i , that is maximum waiting time with respect to their maximum time before treatment. We would remark that maximizing the waiting time of selected patients is a proxy of the OR utilization as proved in [15,16]. Therefore the objective function of our optimization model is

$$\max z = \sum_{i \in I} r_i \sum_{k \in K} \sum_{t \in T} x_{ikt}. \tag{2}$$

The constraints (3) model the selection of the patients from the waiting list (constraints (3a)) and their assignment to an OR devoted to the specialty of the patient (constraints (3c) in which M^1 is an appropriate constants) without exceeding its time capacity (constraints (3b)).

$$\sum_{k \in K} \sum_{t \in T} x_{ikt} \leq 1, \quad i \in I \tag{3a}$$

$$\sum_{i \in I} p_i x_{ikt} \leq s_{kt}, \quad k \in K, t \in T \tag{3b}$$

$$\sum_{i \in I_j} x_{ikt} \leq M^1 \tau_{kt}^j. \quad j \in J, k \in K, t \in T \tag{3c}$$

The constraints (4) model the sequencing of the patients assigned to the same OR. Constraints (4a) reinforce the constraints (3b) by stating that the starting of the procedure of the patient i plus the duration should not exceed s_{kt} . Constraints (4b) guarantee that two patients i and ℓ do not overlap in the same OR k on day t . Constraints (4c) (in which M^2 is an appropriate constant) and (4d) determine the correct sequencing in accordance with the parameter $u_{i\ell}$.

$$\gamma_i + p_i \leq s_{kt}, \quad i \in I, k \in K, t \in T \tag{4a}$$

$$\gamma_i + p_i \leq \gamma_\ell + M_i^2(3 - x_{ikt} - x_{\ell kt} - y_{i\ell kt}), \quad i \in I, \ell \in I, i \neq \ell, k \in K, t \in T \tag{4b}$$

$$\gamma_i u_{i\ell} \leq \gamma_\ell(1 - u_{\ell i}) + M^2(2 - x_{ikt} - x_{\ell kt}), \quad i \in I, \ell \in I, i \neq \ell, k \in K, t \in T \tag{4c}$$

$$y_{i\ell kt} + y_{\ell i kt} = 1. \quad i \in I, \ell \in I, i < \ell, k \in K, t \in T \tag{4d}$$

Finally, the constraints (5) define the domains of the decision variables.

$$x_{ikt} \in \{0, 1\}, \quad y_{i\ell kt} \in \{0, 1\}, \quad \gamma_i \in \mathcal{Z}_+. \tag{5}$$

5 Quantitative Analysis

In this section, we provide a quantitative analysis to prove the effectiveness of our combined approach. The settings of our computational experiments are described in Sect. 5.1 while the analysis is reported in Sect. 5.2.

5.1 Computational Experiment Settings

In our computational experiments we consider the following operative context. The four ORs S_1, S_2, S_3, S_4 operates for 270 minutes everyday from Monday to Friday ($s_{kt} = 270 \forall k \in K$ and $t \in T$). We have two specialities, that are angioplasty procedures ($j = 1$) and echography and computed axial tomography ($j = 2$). The patients belonging to the specialty $j = 2$ are about 17% of the total number of the patients. About 20% of the patients are affected by Covid-19. The patient case mix and their hospital wards are randomly generated using the frequencies extracted from the data. Random generation has been preferred to a case mix extracted directly from real data in order to experiment with instances having different size than the original sample. The procedure duration p_i is computed by summing up the estimate duration (provided by the doctors) of the procedure services. The parameter r_i is randomly generated using a truncated normal distribution in the interval $[1, 120]$ with $\mu = 60$ and $\sigma = 10$.

To capture the possible delay information on the patients, we associate a binary flag to each patient indicating if the procedure of the patient could be on time (0) or delayed (1). This flag is determined according to the frequencies estimated during the process mining phase of the current work and, more specifically, based on the delay distribution related to one delay factor under analysis (e.g., the hospital ward or the service list). Another flag is used to determine if a given patient has an ongoing CoViD-19 infection (1) or not (0).

Such flags are then used to define the following ρ_i values: for a clean³ procedure on time $\rho_i = 1$ or delayed $\rho_i = 2$; for a dirty procedure on time $\rho_i = 3$ or delayed $\rho_i = 4$; for a covid procedure on time $\rho_i = 5$ or delayed $\rho_i = 6$. Using such ρ_i values, we can easily determine the model parameter $u_{i\ell}$ as follows: if $\rho_i < \rho_\ell$ then $u_{i\ell} = 1$, else $u_{i\ell} = 0$. for each $i, \ell \in I$. We generated 10 instances with 100 patients and 10 instances with 120 patients. The 10 different instances differ for the case mix (different patients) with different percentage of Covid patients and for the delay sources used for determining the delay distribution. The solutions of the mathematical model reported in Sect. 4 have been computed using Cplex 12.8⁴ on a Intel Core i5 8265U, 3.90 GHz with 16 GB DDR4 RAM.

5.2 Analysis

We report the results of our computational experiments. For each instance and for each type of delay (hospital ward or service list), one experiment consists in solving the mathematical model reported in Sect. 4. To foster solutions having more on time procedure scheduled, we consider three further experiments in which we consider a modified version of the objective function (2), that is

$$\max z = \sum_{i \in I} r_i d_i \sum_{k \in K} \sum_{t \in T} x_{ikt} \quad (6)$$

³ Whether a procedure is clean or dirty is known in advance, and it is not an information we generate according to some estimated frequency.

⁴ <https://www.ibm.com/products/ilog-cplex-optimization-studio>.

Table 1. Summary of the computational results. Instances with prefix T_HW and T_SL refer to the different sources of delay: HW for hospital ward and SL for service list.

	I	Only $u_{i\ell}$		$d_i = 0.75$		$d_i = 0.50$		$d_i = 0.25$	
		n	Precedence	n	Precedence	n	Precedence	n	Precedence
T_HW_1	100	74	[2, 5, 22, 45, 0, 0]	73	[3, 4, 25, 41, 0, 0]	72	[3, 4, 26, 39, 0, 0]	70	[4, 4, 26, 36, 0, 0]
T_HW_2	100	74	[2, 4, 16, 32, 6, 14]	73	[3, 3, 18, 29, 7, 13]	72	[3, 3, 18, 28, 8, 12]	70	[3, 3, 18, 26, 9, 11]
T_HW_3	100	74	[2, 4, 13, 20, 9, 26]	73	[3, 3, 15, 19, 9, 24]	73	[3, 3, 15, 17, 11, 24]	70	[3, 3, 15, 15, 12, 22]
T_HW_4	100	74	[0, 1, 6, 13, 18, 36]	73	[0, 1, 7, 11, 21, 33]	72	[0, 1, 8, 11, 21, 31]	70	[1, 1, 8, 9, 21, 30]
T_HW_5	100	74	[0, 0, 0, 0, 25, 49]	73	[0, 0, 0, 0, 27, 46]	72	[0, 0, 0, 0, 29, 43]	70	[0, 0, 0, 0, 30, 40]
T_SL_1	100	74	[3, 4, 17, 50, 0, 0]	73	[4, 3, 20, 46, 0, 0]	72	[4, 3, 21, 44, 0, 0]	65	[7, 3, 21, 34, 0, 0]
T_SL_2	100	74	[3, 3, 12, 36, 6, 14]	73	[4, 2, 14, 35, 6, 12]	72	[4, 2, 14, 32, 7, 13]	65	[4, 2, 14, 25, 10, 10]
T_SL_3	100	74	[3, 3, 10, 24, 7, 27]	73	[4, 2, 12, 22, 8, 25]	72	[4, 2, 12, 19, 9, 26]	65	[4, 2, 12, 12, 12, 23]
T_SL_4	100	74	[0, 1, 6, 14, 15, 38]	74	[0, 1, 6, 13, 16, 38]	72	[0, 1, 7, 11, 18, 35]	65	[3, 1, 7, 9, 18, 27]
T_SL_5	100	74	[0, 0, 0, 0, 20, 54]	73	[0, 0, 0, 0, 24, 49]	72	[0, 0, 0, 0, 25, 47]	65	[0, 0, 0, 0, 28, 37]
		22.43	0.33%	42.13	0.31%	28.47	0.45%	24.37	0.31%
T_HW_6	120	76	[2, 2, 41, 31, 0, 0]	75	[2, 2, 45, 26, 0, 0]	73	[2, 2, 50, 19, 0, 0]	65	[4, 2, 51, 8, 0, 0]
T_HW_7	120	76	[1, 2, 32, 22, 10, 9]	75	[1, 2, 36, 18, 10, 8]	73	[1, 2, 40, 13, 11, 6]	65	[3, 2, 41, 5, 11, 3]
T_HW_8	120	75	[0, 2, 23, 14, 20, 16]	75	[0, 2, 25, 13, 22, 13]	72	[0, 2, 28, 9, 24, 9]	65	[1, 2, 29, 4, 25, 4]
T_HW_9	120	76	[1, 0, 9, 9, 33, 24]	75	[1, 0, 9, 8, 37, 20]	71	[1, 0, 10, 5, 42, 13]	66	[1, 0, 10, 2, 44, 9]
T_HW_10	120	76	[0, 0, 0, 0, 43, 33]	75	[0, 0, 0, 0, 47, 28]	71	[0, 0, 0, 0, 53, 18]	65	[0, 0, 0, 0, 55, 10]
T_SL_6	120	76	[2, 2, 30, 42, 0, 0]	75	[3, 2, 33, 37, 0, 0]	71	[4, 2, 38, 27, 0, 0]	62	[7, 2, 39, 14, 0, 0]
T_SL_7	120	75	[2, 1, 22, 32, 7, 11]	75	[3, 1, 25, 27, 8, 11]	71	[4, 1, 29, 21, 9, 7]	62	[6, 1, 30, 11, 10, 4]
T_SL_8	120	76	[1, 1, 15, 24, 16, 19]	75	[1, 1, 16, 20, 20, 17]	71	[2, 1, 19, 16, 22, 11]	62	[3, 1, 19, 9, 24, 6]
T_SL_9	120	75	[0, 1, 8, 10, 25, 31]	75	[0, 1, 8, 10, 29, 27]	72	[0, 1, 9, 6, 33, 23]	62	[1, 1, 9, 3, 36, 12]
T_SL_10	120	75	[0, 0, 0, 0, 33, 42]	75	[0, 0, 0, 0, 36, 39]	71	[0, 0, 0, 0, 43, 28]	63	[0, 0, 0, 0, 46, 17]
		190.46	0.46%	54.70	0.42%	55.47	0.29%	23.71	0.38%

in which the parameter d_i is equal to 1 for each patient whose procedure is usually on time, and equal to $d_i = \{0.25, 0.5, 0.75\}$ for delayed procedures: the highest the value of d_i the higher probability to schedule the patient $i \in I$.

Table 1 reports a summary of the computational results. For each experiment, we report the number n of selected patients among those in I and their distribution with respect to the following precedence order: (i) clean procedure on time, (ii) clean procedure delayed, (iii) dirty procedure on time, (iv) dirty procedure delayed, (v) covid procedure on time, (vi) covid procedure delayed. For each instance, we report the results of four experiments, that is the first one using only the precedence matrix $[u_{i\ell}]$ while the subsequent ones varying the value of the d_i parameters. Finally, the middle and final rows report the average running time in seconds and the average solution gap.

The results reported in Table 1 prove the efficiency and the effectiveness of the proposed approach. In particular, the results in the columns “precedence” show that the number of patients on time increases, while the number of delayed patients decreases as soon as the d_i value decreases. When $d_i = 0.25$, the optimization fosters the selection of patients whose procedure duration is longer than those when $d_i > 0.25$. This justifies the reduction of the overall number of selected patients when $d_i = 0.25$.

To better understand the impact of our approach to the solution, we provide a visual representation of it generated using the python `Plotly.js`. In the following, we consider the instances T_HW_7 and T_SL_7 whose results are already reported in Table 1. In the following figures, a clean procedure on time or delayed

is denoted with the colour navy and dark green, respectively; a dirty procedure on time or delayed is denoted with the colour light green and yellow, respectively; a Covid procedure on time or delayed is denoted with the colour orange and red, respectively.

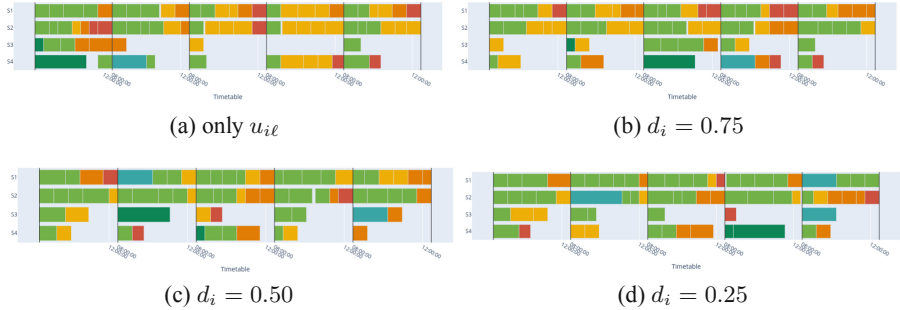


Fig. 4. Delays related to one of the hospital ward cases (instance T_HW_7).

Figure 4 shows the solutions provided by the model in the case of hospital ward delays over 5 days. Figure 4a shows the scheduling in which only the precedence $u_{i\ell}$ is considered while Figs. 4b, c and d show the scheduling using a decreasing value of d_i for delayed patients. All figures prove that the precedence constraints are satisfied, while the sequence (from Fig. 4a to d) shows the evident impact of the use of d_i determining a more robust scheduling as soon as the d_i value decreases and the number of light green procedures increases. The same remarks are confirmed in Fig. 5 in the case of delays determined by the service list.

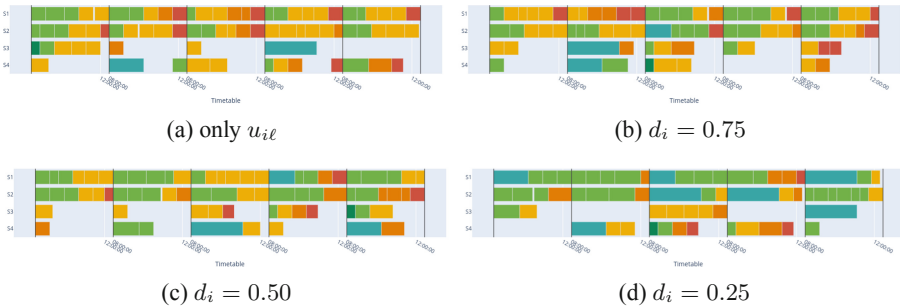


Fig. 5. Delays related to one of the service list cases (instance T_SL_7).

6 Conclusions and Future Work

The paper reports about a pipeline combining PM and operations research for the optimization of the scheduling of IR operating rooms. Although this is only a proof-of-concept analysis, the preliminary results show that PM actually allows for identifying useful knowledge to be leveraged in the design of the optimization model.

In the future we would like to go beyond the post-mortem analysis of the data and focus on the runtime predictions of delays, so as to provide an instrument for the real-time scheduling of interventions based on the needs that may occur.

Acknowledgments. The research is part of the “Circular Health for Industry” project funded by “Compagnia di San Paolo”, call “Intelligenza Artificiale, uomo e società”.

References

1. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*, 2nd edn. Springer, Heidelberg (2018). <https://doi.org/10.1007/978-3-662-56509-4>
2. Munoz-Gama, J., et al.: Process mining for healthcare: characteristics and challenges. *J. Biomed. Inform.* **127**, 103994 (2022)
3. van der Aalst, W.M.P.: *Process Mining - Data Science in Action*, 2nd edn. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
4. van der Aalst, W.M.P.: *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-3-642-19345-3>
5. Pesic, M.: *Constraint-based workflow management systems: shifting control to users*. Ph.D. thesis, TU/e (2008)
6. van der Aalst, W.M.P., De Masellis, R., Di Francescomarino, C., Ghidini, C.: Learning hybrid process models from events. In: Carmona, J., Engels, G., Kumar, A. (eds.) *BPM 2017*. LNCS, vol. 10445, pp. 59–76. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_4
7. Duma, D., Aringhieri, R.: An ad hoc process mining approach to discover patient paths of an emergency department. *Flex. Serv. Manuf. J.* **32**(1), 6–34 (2020)
8. Aringhieri, R.: Online optimization in health care delivery: overview and possible applications. *Oper. Res. Proc.* **2020**, 357–363 (2019)
9. Prodel, M., Augusto, V., Xie, X., Jouaneton, B., Lamarsalle, L.: Discovery of patient pathways from a national hospital database using process mining and integer linear programming. In: *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1409–1414 (2015)
10. Prodel, M., Augusto, V., Jouaneton, B., Lamarsalle, L., Xie, X.: Optimal process mining for large and complex event logs. *IEEE Trans. Autom. Sci. Eng.* **15**(3), 1309–1325 (2018)
11. van der Werf, J.M.E.M., van Dongen, B.F., Hurkens, C.A.J., Serebrenik, A.: Process discovery using integer linear programming. In: van Hee, K.M., Valk, R. (eds.) *PETRI NETS 2008*. LNCS, vol. 5062, pp. 368–387. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68746-7_24

12. Halawa, F., Madathil, S.C., Khasawneh, M.T.: Integrated framework of process mining and simulation-optimization for pod structured clinical layout design. *Expert Syst. Appl.* **185**, 115696 (2021)
13. Moreira, M.W.L., Rodrigues, J.J.P.C., Korotaev, V., Al-Muhtadi, J., Kumar, N.: A comprehensive review on smart decision support systems for health care. *IEEE Syst. J.* **13**(3), 3536–3545 (2019)
14. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: *Advances in Neural Information Processing Systems 18* [Neural Information Processing Systems, NIPS 2005, 5–8 December 2005, Vancouver, Canada], pp. 507–514 (2005)
15. Aringhieri, R., Duma, D.: Patient-centred objectives as an alternative to maximum utilisation: comparing surgical case solutions. In: Sforza, A., Sterle, C. (eds.) *ODS 2017. SPMS*, vol. 217, pp. 105–112. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67308-0_11
16. Aringhieri, R., Duma, D., Landa, P., Mancini, S.: Combining workload balance and patient priority maximisation in operating room planning through hierarchical multi-objective optimisation. *Eur. J. Oper. Res.* **298**(2), 627–643 (2022)



Defining Process Performance Measures in an Object-Centric Context

Bedilia Estrada-Torres^{1,2} , Adela del-Río-Ortega^{1,2} ,
and Manuel Resinas^{1,2} 

¹ Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla,
Seville, Spain

² SCORE Lab, I3US Institute, Universidad de Sevilla, Seville, Spain
{iestrada,adeladelrio,resinas}@us.es

Abstract. The calculation and analysis of process performance indicators (PPIs) and, in particular, the customized performance measures defined to measure a specific process domain, provide insight into whether a business process's results align with the strategic objectives within an organization. These measures and PPIs can be calculated using process execution data. This data is traditionally structured in such a way that for each process instance (case), there is a case notion (object), for example, the order in a purchasing process. Recently, the object-centric approach introduced the multiple case notion, i.e., the idea that several objects can be associated in the execution of tasks of one or several process instances, which better reflects what happens in reality. However, this approach generates more complex event logs that include data involving interacting instances and complex data dependencies. These changes impact the types of PPIs that can be defined and should therefore be analyzed in detail from a different perspective than the traditional one. In this paper, we focus on the PPI modeling area. In particular, we aim at extending the classical definition of PPIs for an object-centric context. For this purpose, we analyze how different customized performance measures are defined in the traditional context and identify a set of requirements to define those measures in an object-centric context. In addition, we propose to extend the established PPINOT metamodel, focused on the definition of PPIs, to integrate the identified requirements, thus laying the groundwork for the automatic calculation of such PPIs.

Keywords: Performance measurement · Process performance indicators · Multiple case notion · Object-centric

1 Introduction

Measuring the performance of business processes is a key operation within an organization to determine whether the expected objectives are being achieved.

This work has been funded by grants RTI2018-101204-B-C22 funded by MCIN/AEI/ 10.13039/501100011033/ and ERDF A way of making Europe; grant P18-FR-2895 funded by Junta de Andalucía/FEDER, UE; and grant US-1381595 (US/JUNTA/FEDER, UE).

According to Kronz [7], process management must involve the collection and analysis of key performance indicators (KPIs), also known as process performance indicators (PPIs), and this, in turn, forms a basis for consistent and continuous optimization of business processes. PPIs are quantifiable metrics that allow us to evaluate the efficiency and effectiveness of business processes. Certain metrics can be applied to almost any process, such as cycle time, defined as the time it takes to process a case or process instance from start to end [3]. However, to evaluate the performance of a process in a more detailed way and verify that the results are aligned with the organization's strategic objectives, it is important to be able to define measures and PPIs specific to the process domain under analysis. For example, the percentage of orders returned during a quarter with respect to the orders placed in that period.

PPIs can be computed directly from *event logs* [12]. In most traditional scenarios, an event log should have at least (i) a *case identifier* to indicate in which *case* or process instance the event occurred, (ii) an identifier of the *task* to which the event refers, and (iii) a *timestamp* indicating when the event occurred [3]. The attribute used to assign an event to a case is called the *case notion* [2]. An event log usually contains information on the executed activities of several cases, where it is assumed that each event is related to exactly one activity and is traceable to exactly one case. This feature, known as *single case notion* [2], is a widely accepted limitation in process management, and in particular in process mining, because it is considered not to be a faithful representation of reality.

Recent proposals such as [1,2,5], focused their efforts on the management and analysis of process execution data by assuming that multiple attributes can be considered as case notion (called *object types*), that these objects can coexist, and that an event can refer to any number of objects corresponding to different types. This *object-centric* behavior, which includes the *multiple case notion*, is considered closer to reality, thus seeking optimization in the process management. Figure 1 shows a simplified version of the Order-Delivery process. Figure 1a depicts the process using a traditional BPMN model, where a single case notion, the order, is used. Figure 1b shows the process considering the multiple case notion where several objects are involved (simplified version of the process described in [1]). In the latter, multiple relationships of different cardinalities (one-to-many and many-to-many) are identified between process activities/events (left) and object types (right), hereafter referred to as *objects*.

This change in the way processes are dealt with is also reflected in the way data is recorded, giving rise to standards such as Object-Centric Event Log OCEL [5], which is capable of supporting the multiple case notion and giving importance to object information and relationships. Moreover, this change also affects the way in which data is analyzed. Proposals such as [2] focus on calculating precision and fitness to determine the quality of the process itself. In the specific area of performance measurement, only [9] centers on the definition of a set of predefined process-related time measures. However, as far as we know, there is no proposal that allows the definition of customized and domain-focused PPIs and measures for each process in an object-centric context.

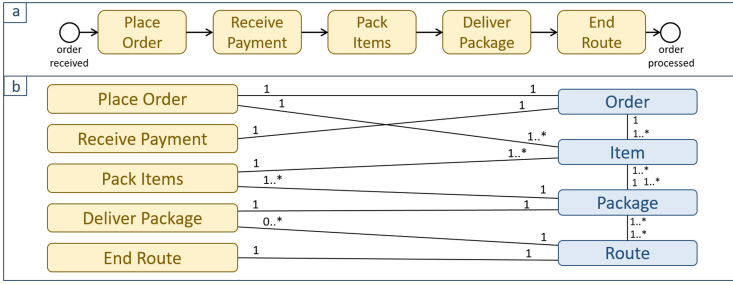


Fig. 1. (a) Order-to-Delivery(O2D) process with *order* as case notion. (b) Relationships between activities (left) and objects (right) of the O2D process adapted from [1]

Based on the example in Fig. 1a, where there is a single case notion (*order*), it would be possible to calculate the total processing time of an order using the registered execution times of the first and last activity in the process; or the average number of orders delivered during a period of time. However, if we analyze Fig. 1b there is no direct relationship between all objects and all process events, so in that context, the task of calculating the total processing time of an order or the orders delivered is not a trivial task. Furthermore, returning to Fig. 1a, since the *order* is the case notion of the process, in this scenario it would be complex to assume and manage the fact that, for example, a package could contain items from several orders. Therefore, we would not be able to define measures such as the average number of items from different orders that are delivered in the packages.

From the previous examples, we can deduce that in order to calculate measures in an object-centric context, it is necessary to perform intermediate operations that allow us, on the one hand, to trace information between different objects and, on the other hand, to relate elements associated with different instances of a process. These new features open up a wide range of possibilities for optimization and automation of process performance analysis, extracting performance data from events and objects recorded in object-centric event logs. This paper seeks to make a contribution to the modeling of PPIs, and in particular to the definition of customized performance measures by considering an object-centric approach. To this end, we identified a set of requirements to be taken into account in the definition of such measures and then integrated them into the existing PPI definition metamodel PPINOT [12]. To help guide our research, we formulated two research questions: (RQ1) *What are the main characteristics to be considered for measuring process performance from an object-centric event log?*, and (RQ2) *How can performance measures and PPIs be defined in an object-centric context?*

The remainder of this paper is organized as follows. Section 2 introduces some basic concepts and mentions relevant work in the area. Section 3 describes our approach to answering the research questions posed. Section 4 provides a discussion of the proposal. Finally, Sect. 5 concludes this paper.

2 Background and Related Work

Conventionally, event logs are generated following standards that allow structuring the information for the purpose of transferring these data in a unified way for its later extraction and analysis. The *eXtensible Event Stream* (XES) [6] is one of the most widely used for recording events associated with processes from a single case notion context. Therefore, to analyze a different process perspective, another notion (another object), it would be necessary to generate a new dedicated event log [8]. The *eXtensible Object-Centric* (XOC) format [8] supports multiple case notion and aims to deal with the relationships (one-to-many and many-to-many) between objects, avoiding the definition of a case notion. XOC logs have the disadvantages that they are usually very large, complex and have performance issues [1,5]. Recently, the Object-Centric Event Log (OCEL) [5] standard was proposed with the objective of exchanging data with multiple case notion between different information systems and process mining tools. This OCEL format allows the log data to be analyzed from different perspectives (multiple case notion) and makes it possible to deal with two concepts derived from this scenario: convergence and divergence [1]. We refer to convergence, when an event can be related to different cases. For example, for the process in Fig. 1a, if instead of using `order` as case notion, we use another object, such as `item`, if there are two or more items associated to an order, the `Place Order` activity for that order would appear in as many traces as items are linked to that order. The divergence, on the other hand, refers to when for a given case there may be multiple instances of the same activity. For example, for the process in Fig. 1a, the divergence arises when discussing the possibility that for a process instance, an order, the `Pack Items` activity can be executed several times and that the `items` packed in the first instance of that activity can be associated with the first, the second or the last package associated with the order. These multiple relationships derived from the multiple object analysis are not considered in traditional event logs, thus losing relevant process data.

The other area of interest related to our paper is performance measurement. Although there is no single way to define PPIs, they are usually described by a set of attributes. In this paper, we based on the definitions of performance measures and PPIs proposed in [11] and formally described in the *PPINOT metamodel* presented in [10] (adapted from [12]), which provides a detailed structure for defining PPIs. This metamodel emphasizes the need to include as PPI attributes at least one *identifier*, a descriptive *name*, a *process* in which the PPI is defined, a set of *goals* indicating the relevance of the PPI, a *measure definition* that specifies how to calculate the PPI, a *target* value to be reached, a *scope* to define the subset of instances to be considered to calculate the PPI value, and a set of human resources to be *responsible*, *accountable*, and *informed* about the PPI. A PPI is calculated through a measure, which in turn can be defined by other measures. Each measure is related to a *condition* that specifies when the measure value is obtained. A condition relates an *activity*, an *event* or a *data object* to their respective states. Three types of *base* measures are defined in [10], where a value is obtained for each measure: *time* (measures the time elapsed between

two events), *count* (measures the number of times an event occurs), and *data* (measures the value of a certain part of a data object). More complex measures, called *derived* measures, can be defined by combining the previous measures and applying mathematical or logical operations on them. Finally, all those measures can be *aggregated* according to aggregation functions such as sum, maximum, average. Given the importance of the measure definition for the calculation of PPIs, in this paper, we will focus on how to define these measures.

PPIs can be calculated automatically from data recorded in event logs [12]. Currently, most process mining tools can handle the event logs in XES format [3]. However, as there is a change in the way logs are generated, from single case notion to object-centric approach, it is reasonable to foresee some changes in the way calculation and analysis of object-centric data is performed. Recent works have begun to propose adaptations of the focus of analysis. In [4], a new measure is included to focus on conditions based on data objects attributes, but only under the single case notion context. Related to the definition of performance measures in an object-centric approach, we have only identified one proposal focused on the calculation of predefined time measures, such as waiting times or cycle time [9]. However, we have not identified any proposal that addresses the definition of customized performance measures in an object-centric context. In this paper, we seek to fill this gap and propose first steps for defining and modeling customized performance measures in an object-centric context.

3 Performance Measurement in the Multiple Case Notion

Event logs capable of reflecting multiple case notion (hereafter, *object-centric logs*), are formed by a set of *events* and *objects*. According to [5], an *event* represents an execution record of a business process and associates multiple elements (identifier, activity, timestamp and relevant objects) and optional characteristics such as event attributes. An *object* indicates the information of an object instance in the business process. It must contain a type and may contain several attributes that describe it. Since the content and structure of data recorded in object-centric logs is different from traditional logs, the way in which performance information is extracted and analyzed may also vary.

To establish these differences, we based on the PPINOT modeling characteristics for the definition of PPIs and measures to identify modeling requirements that support the object-centric approach (Sect. 3.1). In PPINOT, a *BaseMeasure* (*bm*) is calculated based on a case, in such a way that $bm(case) = mvalue$. The measures described by PPINOT (Sect. 2) are based on conditions that make it possible to specify (i) the time when something happens, for example, the start or end of the execution of an activity, or the time when the state of a data object changes; or (ii) the value of a certain part of a data object (data property), for example, the name of the customer (property) that requests an order (data object). However, in an object-centric context, it is necessary to take into account not only the relationship among the events occurred, but also among objects and their characteristics. The new modeling requirements were

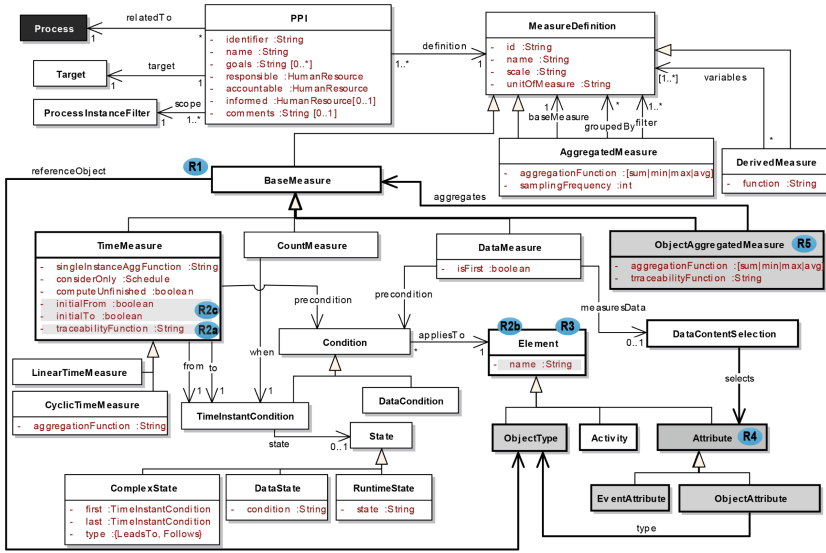


Fig. 2. Extension of the PPINOT metamodel integrating object-centric concepts.

integrated into the PPINOT metamodel, as shown in Fig. 2. They mainly affect the elements on which a measure can be calculated and a new measure was also incorporated. Those changes will be described in detail in Sect. 3.2.

3.1 Comparison of Measure Definitions

In this subsection, we focus on describing how different types of measures would be defined in a traditional context and the identified drawbacks and/or necessary changes to define similar measures in an object-centric context. For this purpose, we based on the types of measures proposed in [10]. Hereafter, when we refer to a traditional event log, we refer to a single case notion event log. In a traditional log, events recorded are related to a single case or process instance.

Notion of Measures. In a traditional context, a *BaseMeasure* bm is related to a single case c , $bm(c) = mvalue$, where $mvalue$ is the value of the measure. In the object-centric context, since there is a multiple case notion, the definition of the measure must change to specify the reference notion (the object type) on which each measure will be calculated. **Requirement 1: A measure must specify the object type used as a reference for its calculation.**

Time Measures. A *time measure* measures the duration of time between two time instants. In a traditional log, a time instant is related to the execution of a process activity or event (without duration), the whole process, or the change in the value of an object. For the Order-To-Deliver process in Fig. 1a, a measure

of the total time to process an order could be defined as: M_{time} = The duration between the time instant when **Place Order** changes to state **active** and the time instant when activity **Deliver package** changes to state **completed**. The tracking between the two instants is relatively straightforward since both activities, and all the activities executed in between, are related to the same case notion, an order. However, as shown in Fig. 1b, when referring to the object-centric context, not all objects are related to all activities, so there is no direct traceability between the different relationships. It would be relatively easy to calculate the instant at which an **order** is registered, since there is a relationship between **Place Order** and **Order**. Nevertheless, an **order** would be considered completed when all the **packages** associated with it are delivered. In this case, there is no direct relationship between **Order** and **End Route** and in the event log, this results in the fact that there is no **End Route** event that refers directly to any **Order**. Therefore, to calculate the measure, it would be necessary to perform a series of operations to trace the end of an activity (**End Route**) to another object (**Order**). Thus, **Requirement 2: Identify traceability between objects**.

If the reference object is *order*, the two time instants in our example should be calculated differently using a *traceability function*, but the reference object should be indicated for both of them (*Requirement 1*). Given that there is a direct relationship between the event **Place Order** and the object **Order**, the *start instant* could be defined as the traceability function $startInstant = getAttribute(attribute, Event, Object)$, where **attribute** indicates the attribute in the log to be analyzed (timestamp); **Event** is the event from which the attribute is obtained (**Place Order**), and **Object** is the reference object (**Order**) related to the event. To obtain the *end instant*, we must know when the last package associated with each order was delivered. To do this, we must define a traceability function between the objects **Order** and **Package** to then obtain the maximum timestamp of the delivery times registered of all the packages associated with an order. The traceability function for *end instant* would be defined as:

```

itemsOrder = getObjects(Item, Order, PlaceOrder)
packsItems = getObjects(Package, Item, PackItem)
packsDelivered = getObjects(Package, Package : packsItems, DeliverPackage)
routesPacksDel = getObjects(Route, Package : packsDelivered, DeliverPackage)
deliveries = getEvents(DeliverPackage, Route : routesPacksDel)
endInstant = max(timestamp, deliveries)

```

In the above, we used $getObjects(ObjectA, ObjectB, Event)$ to select objects, where *ObjectA* is the object returned, *ObjectB* is the object that *ObjectA* relates to in the *Event* event. $getObjects(ObjectA, ObjectB:listObjectsB, Event)$ is similar, but in this case, the relationship of *ObjectA* is analyzed with a subset of elements of type *ObjectB*. $getEvents(Event, Object)$ returns a subset of events of the type indicated in *Event*, where objects of type *Object* participate, or a set of these. Finally, the highest timestamp of all the packages delivered in an order is assigned as *end instant*.

Count Measures. A *count measure* measures the number of times something happens. This again refers to the measuring of something at a given instant, an event. For example, in the process of Fig. 1a, a count measure could be *the*

number of packages delivered to fulfill an order, defined as: M_{count} = The number of times activity **Deliver Package** changes to state **completed**. As was the case with time measures, to calculate the number of packages of an order in an object-centric context, it is needed to perform a traceability between a reference object (**Order**) and another element, in this case an object (**Package**). In addition to base a measure on the count of time instants, we could also be interested in counting elements associated with objects and their attributes. Such as the *number of items of type technology*, where **technology** would be an attribute of the object **Item**. This is not possible with the original definition of a count measure, therefore, we define the **Requirement 3: Extend the use of count measures to allow them to be applied, in addition to time instants, to objects and their attributes.**

Data Measures. A *data measure* measures the value of a certain attribute included in the event log. This measure can make a selection based on a property of an object, for example, the *invoice amount* or the *department* in which the order is placed. In a traditional approach, it could be defined as: M_{data} = The value of property **department** of an **Order**

In an object-centric context, the data measure should focus on object types and the attributes associated with them or with events, so it would be necessary to specify the element on which the data measure is applied. Therefore, **Requirement 4: Specify the element on which a data measure is applied so that it can be calculated on object types and attributes of object or events.**

Aggregated Measures. An *aggregated measure* uses an aggregation function *sum, minimum, maximum, average* to aggregate measures in several process instances, and the result can be grouped by the value of another measure. In a traditional context and taking as a basis the count measure M_{count} , an aggregated measure could be defined as: M_{agg_count} = *The sum of the number of times Deliver Package changes to state completed and is grouped by delivery city of Order.* In the above definition, the *aggregation function* is *sum*, the *grouping property* is *delivery city* that is an attribute of the object **Order**. As a result we would obtain, for example, $\{(150, \text{London}), (280, \text{Madrid}), \dots\}$

For the aggregated measure no new requirements have been identified in the object-centric context. The measures used to calculate it, should follow the requirements that apply to them (e.g., Requirement 3 for the count measure).

Derived Measures. A *derived measure* defines a function over other measures. Following our example of the Order-To-Deliver process (Fig. 1a), a derived measure could be *the percentage of paid orders*. In a traditional context, with **Order** as a case notion, the measure would be defined as: M_{der} = the function $\frac{paid}{regs} * 100$, where *paid* is the measure defined as the sum of the number of times **Receive Payment** changes to state **completed**, *regs* is the measure defined as the sum of the number of times **Place Order** changes to state **completed**.

As with aggregated measures, no new requirements have been identified for this measure in the object-centric context. A derived measure is defined over other measures by adding constraints (operations) and must support the definitions of those measures, taking into account the requirements defined for them.

Several Reference Objects. So far, in the object-centric context, we have mentioned the possibility of defining measures in which it is necessary to specify a reference object on which to pivot to perform the traceability that allows us to obtain the information desired. For example, `order`, when we want to know *the number of packages shipped for an order*. However, to exploit the possibilities of an object-centric event log it is also necessary to consider the possibility that a measure, for example an aggregated measure, involves more than one object type as reference. For example, in a measure defined as *the average time to deliver package for each order*, `Package` would be the reference object related to a time measure and an aggregated function would be applied taking `Order` as reference object. The above derives *Requirement 5: A measure can involve several objects (case notions) as reference object*.

3.2 Extension of the PPINOT Metamodel for the Definition of PPIs

This subsection aims to show how the modeling requirements identified in the previous subsection can be integrated into the PPI metamodel.

The result of this integration is shown in Fig. 2 as a UML diagram. White classes represent original PPINOT concepts taken or adapted from [10]. Gray classes and associations with thicker border are new elements derived from the identified requirements. White classes with thicker border represent classes slightly modified to adapt to the requirements. The shaded texts represent new attributes included. The blue circles indicate the requirements (Rx) to which that element of the metamodel is related. To extend the PPINOT metamodel, in addition to the requirements identified in Sect. 3.1, we adopted as reference the definitions of the object-centric elements described in [1].

For *Requirement 1* it was noted that, due to the multiple case notion in an object-centric context, it is needed to indicate a reference object on which the measure would be calculated. After analyzing all types of measures, we found that this only affects base measures, since aggregated and derived measures are not defined directly on data of executed process elements, but on other measures. Therefore, to cover this requirement, a `referenceObject` element was added and related to `BaseMeasure` ($R1$), so that: given a base measure bm , for an object of type ot , and an object $o \in ot$, the base measure is calculated as $bm(o) = mvalue$.

Requirement 2 points out the need to establish a relationship between different objects to trace them and determine time instants to make a measurement. This can be done through a sequence of operations, such as those defined during the analysis of the time measure described in Sect. 3.1: `getAttribute`, `getObject`, `getEvent`. These operations can be seen together as a `traceabilityFunction` of a `TimeMeasure` in which the participating elements (source and destination) are specified ($R2a$).

Since in the original metamodel a `TimeInstantCondition` could only be applied on `States` of `Activities` and `Objects` of a process, this should be extended so that a `condition` can be applied on `Activities`, as up to now, but also on a `ObjectType` and `Attribute`, both of *events* and *objects* (**R2b**). To provide more flexibility to the definition of the `TimeMeasure`, we included in it the attributes `initialFrom` and `initialTo`. They indicate whether the first or the last occurrence of the instance of an element should be taken into account for the calculation of the instants (**R2c**).

Requirement 3 again points out the need to extend the elements on which a measure can be applied to consider both object types and attributes of objects and events. The extension related to the count measure was addressed in (**R2b**) by associating the `Condition` to `Element` (**R3**), which in turn can be an `ObjectType`, `Activity` or `Attribute`. So a `CountMeasure` M_{count} , defined as the *number of items in a registered order* could be defined as $M_{Count(Item)}(Place\ Order, completed)$, where the reference object is `Item` and the value for the measure calculation is taken when the `Place Order` activity was completed; or another one as *the number of packages delivered* as $M_{Count(Package)}(deliveredPackage)$, where `delivered` is an attribute of `Package`.

With regard to **Requirement 4**, associated with the `DataMeasure`, slight changes were introduced. We specified in more detail the scope of application of the `DataMeasure`. In this case, to define a `DataMeasure` it is necessary to establish the `DataContentSelection` to indicate the part of data to be obtained with the measure. These `DataContentSelection` is defined by means of an `Attribute` (**R4**) either `ObjectAttribute` or `EventAttribute`. For an `ObjectAttribute`, it is necessary to specify the `ObjectType` to which the attribute of interest belongs.

The last requirement, **Requirement 5**, is related to the possibility of involving several object types in the definition of a measure. To address this requirement, we included a new base measure called `ObjectAggregatedMeasure`. Its purpose is to be able to apply an `aggregationFunction` on a `BaseMeasure`, using a reference object different from that of the `BaseMeasure`. This is reflected through the `aggregates` association. The `aggregationFunction` is defined on a reference object (*obj1*), and `BaseMeasure` by definition is also associated to an object (*obj2*) by means of the association `referenceObject`. The relationship between *obj1* and *obj2* is established through the `traceabilityFunction`. For example, the measure `ObjectAggregatedMeasure` would allow the definition of measures such as: *the sum of orders whose number of items is greater than 5*, which can be expressed as $M_{ObjectAggregated1} = SUM_{Order}(M_{Count(Item)}(Place\ Order, completed)) > 5$). In this example, `SUM` is the `aggregationFunction` applied on the `Order` object (*obj1*) and $M_{Count(Item)}$ is the `CountMeasure` whose `referenceObject` is `Item` (*obj2*), and whose `Condition` is *(Place Order, completed)*, that is, the `Items` associated to the `Place Order` event will be taken into account. Since the measure specifies a condition representing an operation (**greater than**, `>`) implicitly, a `Derived measure` is being defined. If the mathematical operation were not necessary, this would not affect the relationship

between the rest of the elements. The relationship between the **Order** object and **Item** is defined through the *Order ~ Item traceabilityFunction* as described previously in this section.

3.3 Modeling of Measures

Based on the extended metamodel presented in Fig. 2, the measures mentioned in Sect. 3.1 have been modeled. Due to space restrictions they are not included in this paper, but are available online¹.

4 Discussion

Throughout this paper, we have tried to answer the research questions posed in Sect. 1 regarding the definition of performance measures taking into account the characteristics derived from an object-centric context. We started by analyzing the way in which performance measures are defined in a traditional context, in order to identify the relevant characteristics that should be taken into account when defining measures in the object-centric context. To answer the question about the characteristics of performance measurement in an object-centric context (RQ1), we focused on how performance measures are defined in a traditional context using the PPINOT metamodel as a reference, and then tried to define the same measures in an object-centric context, if possible. As a result of this phase, we identified 5 requirements that vary between the traditional and the object-centric context. As mentioned in Sect. 3, some of them affect more than one performance measure definition. Among them we can highlight:

- Take into account new attributes that must be considered in the measure definition. First, to specify the reference object on which the measure is defined (**referenceObject**), and second, to establish the relationship between process objects (**traceabilityFunction**) in case there is no direct relationship between them. For example, between **Order** and **Package** (Fig. 1b).
- Extend and/or modify, as appropriate, the application scope of the measures, to take into account the elements that are part of an object-centric event log: **ObjectTypes** and **Attributes** of objects and events. This does not mean that the original condition criteria are eliminated, since it is still necessary to take into account the time instants where events occur, for example.
- Allow the definition of a performance measure by reference to more than one object (object types).

The last point mentioned is closely related to the second question on how performance measures can be defined in the object-centric context. In general, all the measures proposed in the PPINOT metamodel for a traditional context can be defined in an object-centric context; however, adaptations must be made to include the requirements described in Sect. 3.1. In addition, since in an

¹ <https://github.com/Adartse/PerformanceMeasuresInAnObjectCentricContext>.

object-centric context it is useful to define measures taking into account several object types, it is necessary to introduce a new measure that takes into account the reference object on which any base measure is defined, and also to allow the application of aggregation functions on it taking another object as reference. Both this measure and the previous requirements were integrated into the PPINOT metamodel. In this work, we only address the modeling of performance measures from a conceptual point of view. However, one of the characteristics of the PPINOT metamodel is that it allows to facilitate the operationalization of the automatic calculation of PPIs. Therefore, with this analysis and initial proposal, we can lay the foundations for the operationalization of customized performance measures in an object-centric context. Although in this work we rely on PPINOT, the results of the analysis done could be used with other similar artifacts for the definition of process performance measures.

5 Conclusion and Future Work

In this paper, we present a novel approach for the definition of domain-specific performance measures in the object-centric context with the objective of optimizing the measurement of process performance. The analysis performed shows that the definition of measures in this context is not a trivial task since it is necessary to define and fulfill new requirements derived from the relationship of multiple objects and events. As future work, we propose to extend the set of test measures to identify new requirements (if any). Then, we propose to extend the formalization of the PPINOT metamodel so that performance measurement can be done automatically from an object-centric event log. We also plan to use it in a real context to evaluate its applicability.

References

1. Aalst, W.M.P.: Object-centric process mining: dealing with divergence and convergence in event data. In: Ölveczky, P.C., Salaün, G. (eds.) SEFM 2019. LNCS, vol. 11724, pp. 3–25. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30446-1_1
2. Adams, J.N., Van Der Aalst, W.M.: Precision and fitness in object-centric process mining. In: 2021 3rd International Conference on Process Mining (ICPM), pp. 128–135 (2021)
3. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, 2nd edn. Springer, Heidelberg (2018). <https://doi.org/10.1007/978-3-662-56509-4>
4. Estrada-Torres, B., Richetti, P.H.P., Del-Río-Ortega, A., et al.: Measuring performance in knowledge-intensive processes. *ACM Trans. Internet Tech.* **19**(1), 1–26 (2019)
5. Ghahfarokhi, A.F., Park, G., Berti, A., van der Aalst, W.M.P.: OCEL: a standard for object-centric event logs. In: Bellatreche, L., et al. (eds.) ADBIS 2021. CCIS, vol. 1450, pp. 169–175. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85082-1_16

6. Group, X.W.: IEEE standard for eXtensible event stream (XES) for achieving interoperability in event logs and event streams. IEEE Std. 1849-2016 pp. i–48 (2016)
7. Kronz, A.: Managing of process key performance indicators as part of the ARIS methodology. In: Corporate Performance Management, pp. 31–44. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-30787-7_3
8. Li, G., de Murillas, E.G.L., de Carvalho, R.M., van der Aalst, W.M.P.: Extracting object-centric event logs to support process mining on databases. In: Mendling, J., Mouratidis, H. (eds.) CAiSE 2018. LNBIP, vol. 317, pp. 182–199. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92901-9_16
9. Park, G., Adams, J.N., van der Aalst, W.M.P.: Opera: Object-centric performance analysis. CoRR abs/2204.10662 (2022)
10. Resinas, M., del Río-Ortega, A., Ruiz-Cortés, A.: PPINOT computer and ppinot4py. In: ICPM Demo Track 2021, pp. 51–52 (2021)
11. del Río-Ortega, A., Resinas, M., et al.: Using templates and linguistic patterns to define process performance indicators. *Enterp. Inf. Sys.* **10**(2), 159–192 (2016)
12. del Río-Ortega, A., Resinas, M., Cabanillas, C., et al.: On the definition and design-time analysis of process performance indicators. *Inf. Sys.* **38**(4), 470–490 (2013)

**10th International Workshop on
Declarative, Decision and Hybrid
Approaches to Processes (DEC2H 2022)**

10th International Workshop on Declarative, Decision and Hybrid Approaches to Processes (DEC2H 2022)

Rules and decisions define the behavioral constraints and factors determining the achievement of process goals. Business processes frequently involve rule-bound decisions – particularly knowledge-intensive processes, which operate in highly variable contexts and are thus flexible by nature. When describing such processes, variability and flexibility call for explicit statements of the underlying rules and decisions. While traditional notations such as BPMN excel at describing “happy paths”, they may fall short when modeling flexible and varying rules and decisions, wherein procedural models tend to clutter and become imprecise or impractical. Declarative modeling paradigms aim to directly capture the business rules or constraints underlying the process and thus tackle this challenge. To this end, several languages have been proposed, including DECLARE, Dynamic Condition Response (DCR) Graphs, Decision Modelling and Notation (DMN), fragment-based Case Management (fCM), Case Management Model and Notation (CMMN), Guard Stage Milestone (GSM), and Declarative Process Intermediate Language (DPIL). Recently, there has been a surge of interest in *hybrid* approaches, combining the strengths of declarative and procedural modeling paradigms.

In the workshop on Declarative, Decision and Hybrid Approaches to Processes (DEC2H), we are interested in the application and challenges of decision- and rule-based modeling in all phases of the Business Process Management lifecycle: identification, discovery, analysis, redesign, implementation, and monitoring.

This year, DEC2H reached the milestone of the tenth edition of the workshop – known as DeHMiMoP (Workshop on Declarative/Decision/Hybrid Mining and Modelling for Business Processes) till its sixth edition. DEC2H 2022 attracted eleven high-quality international submissions. Each paper was reviewed by at least three members of the Program Committee. Of all the submitted manuscripts, the top five were accepted for presentation.

The invited talk of Paolo Ceravolo, entitled “The quest for conflicting knowledge - How to stay sane while data gets messed up”, opened the workshop. In his thought-provoking presentation, he showed that typically science proceeds from observation to theory, from facts to the definition of models, but this does not hold in design science where observations of the artifact can bring the designer to change both the system or its goals. Similarly, when comparing the expected or observed behavior of a process the designer is in front of an epistemic dilemma: changing the specification of the system or modifying the way we are enforcing its behavior? He thus showed that such conflicts are not an accident nor an annoyance but a great opportunity to uplift knowledge if they are patiently turned into action-oriented behavior.

Julia Holz et al. analyzed the case-management languages CMMN, fCM, and PHILharmonic flow with the structured methods of functional comparison and user study. Maximilian König et al. illustrated a set of guidelines, also integrated in a

modeling tool, to support the creation of fragment-based case management models avoiding errors and misrepresentations. Roberto Casaluze et al. proposed a novel research line integrating statistical model checking with process mining to identify issues in the process model and suggest improvements in it. Joscha Grüger et al. investigated in a case study the adoption of the Arden Syntax for the conformance checking of treatment cases and medical guidelines. Axel Christfort et al. showed the improvement that the application of noise filtering brings to the quality of discovered DCR graphs from event logs.

We thank the authors for their noteworthy contributions and the members of the Program Committee for their invaluable help in the reviewing and discussion phases. We hope that the reader will benefit from reading these papers to know more about the latest advances in research about declarative, decision, and hybrid approaches to business process management.

September 2022

Claudio Di Ciccio
María Teresa Gómez-López
Tijds Slaats
Jan Vanthienen

Organization

Workshop Chairs

Claudio Di Ciccio
María Teresa Gómez-López
Tijs Slaats
Jan Vanthienen

Sapienza University of Rome, Italy
Universidad de Sevilla, Spain
University of Copenhagen, Denmark
KU Leuven, Belgium

Program Committee

Amine Abbad Andaloussi
Andrea Burattin
Alessio Cecconi
Carl Corea
João Costa Seco
Massimiliano de Leoni
Riccardo De Masellis
Chiara Di Francescomarino
Rik Eshuis

Amin Jalali
Krzysztof Kluza

Hugo A. López
Fabrizio Maria Maggi
Artem Polyvyanyy
Flavia Santoro
Stefan Schönig
Han van der Aa
Mathias Weske

University of St Gallen, Switzerland
Technical University of Denmark, Denmark
WU Vienna, Austria
University of Koblenz-Landau, Germany
Universidade NOVA de Lisboa, Portugal
University of Padua, Italy
Uppsala University, Sweden
Fondazione Bruno Kessler-IRST, Italy
Eindhoven University of Technology,
The Netherlands
Stockholm University, Sweden
AGH University of Science and Technology,
Poland
University of Copenhagen, Denmark
Free University of Bozen-Bolzano, Italy
University of Melbourne, Australia
UERJ, Brazil
University of Regensburg, Germany
University of Mannheim, Germany
HPI, University of Potsdam, Germany

The Quest for Conflicting Knowledge: How to Stay Sane While Data Gets Messed Up

Paolo Ceravolo

Università degli Studi di Milano (UNIMI), Milan, Italy
paolo.ceravolo@unimi.it

Extended Abstract

Process Mining (PM) is typically presented as the intersection between business process management and data mining [1]. Whereas significant differences exist in the *knowledge acquisition* process of PM and other procedures learning from data. For example, to validate predictive models in machine learning one can compare the output of an artifact, the predictive function learned, with ground truth, the known target values to be predicted. In contrast, PM tasks measure the appropriateness of two artifacts, the process model, a set of specifications, and the event log, example behavior about business process execution, without knowing any reference to measure their adherence to the reality, the system they aim to stand for.

This brings us to what we propose to call the *PM epistemic dilemma*. As illustrated in Fig. 1, published in [2], measuring the appropriateness of an event log over a model, for *process discovery* [3–5] or *conformance checking* [6, 7], provides us with results that cannot be considered explanatory by themselves. For example, in order to understand if the behavior observed in the event log but not in the model is dysfunctional, area 2, or unspecified, area 7, we need to know something about the system but this information is not part of the PM learning process.

What we call system (S) in the validation problem is not directly accessible to us and it is much more than just L and M . It also includes business rules (BR), prescriptions on the system, and world knowledge (W), constraints to use in interpreting the events. Using these sources of knowledge enriches our information but brings us in front of conflicting knowledge. For example, if we have a case included in the event log, $c_i \in L$, but not in the intersection between the log and the model, $c_i \notin (M \cap L)$, getting information about the conformance of this case with business rules and world knowledge, for example, $BR \cup W \models c_i$, brings us in front of a conflict if we expect that or model is also consistent with business rules and world knowledge, $BR \cup W \models M$. To solve the conflict we need to take a position about the relevance, the adherence to the system, of the different sources of knowledge we are collecting. We may decide to consider M incomplete and extend it by additional specifications to have $c_i \in M$ or we may consider updating our representation of BR in order to have $BR \cup W \not\models c_i$.

Many disciplines have addressed the problem of resolving conflicting knowledge [8–11]. In general, the strategies to be followed are two: (i) assessing the specificity or the priority of the different knowledge sources, to prefer one source over others;

(ii) searching for functionalities or specifications that can remove the conflict. In business process management these actions are often addressed by domain experts using non-formal tools. But it is also possible to establish systematic exploratory strategies. Let's define the notion of *event log segment* as a collection of cases that conform to a specific pattern, following the same set of constraints, for example, a set of business rules. Using it we can partition the event log by searching for conflicting and non-conflicting segments.

Comparing these partitions we can foster our knowledge acquisition process by observing associations, for example in terms of conditional probabilities, between business rules, or any other attribute, defining a segment and the existence of conflicts with M or W . Using strategies for resolving conflicts on the most relevant association we can improve our understanding of the system S , progressively reducing the difference between the areas 7 and 2 or 5 and 4 of Fig. 1. A mature evaluation of these associations implies adopting causal mining [12] approaches and statistical inference [13] to explore the many alternative ways of observing the association between two variables [14].

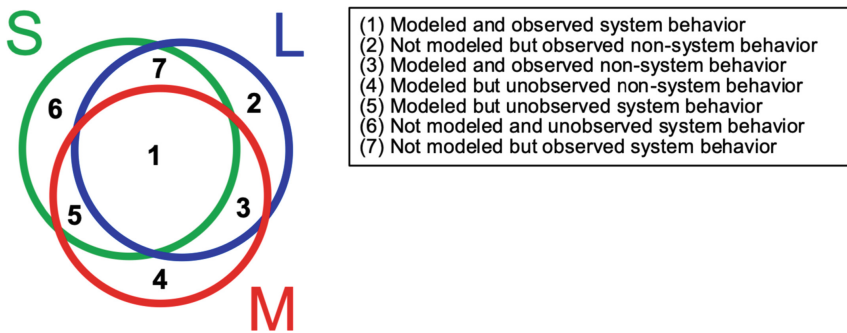


Fig. 1. Possible outcomes of measuring the appropriateness of a model M and event log L in relation to a system S [2]

References

1. van der Aalst, W. et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds). Business Process Management Workshops. BPM 2011. Lecture Notes in Business Information Processing, vol. 99. Springer, Heidelberg (2012). [10.1007/978-3-642-28108-2_19](https://doi.org/10.1007/978-3-642-28108-2_19)
2. van der Aalst, W.M.: Relating process models and event logs-21 conformance propositions. In: ATAED@ Petri Nets/ACSD, pp. 56–74 (2018)
3. Aalst, W.v.d., Carmona, J., Chatain, T., Dongen, B.V.: A tour in process mining: from practice to algorithmic challenges. In: Transactions on Petri Nets and Other Models of Concurrency XIV. pp. 1–35 Springer, Heidelberg (2019). [10.1007/978-3-662-60651-3](https://doi.org/10.1007/978-3-662-60651-3)

4. Buijs, J.C., van Dongen, B.F., van der Aalst, W.M.: Quality dimensions in process discovery: the importance of fitness, precision, generalization and simplicity. *Int. J. Cooperat. Inf. Syst.* **23**(01), 1440001 (2014)
5. Jr., S.B., Ceravolo, P., Damiani, E., Tavares, G.M.: Using meta-learning to recommend process discovery methods. *CoRR abs/2103.12874* (2021)
6. Van Der Aalst, W.: Process mining: Overview and opportunities. *ACM Trans. Manag. Inf. Syst. (TMIS)* **3**(2), 1–17 (2012)
7. Carmona, J., van Dongen, B., Weidlich, M.: Conformance Checking: Foundations, Milestones and Challenges. In: van der Aalst, W.M.P., Carmona, J. (eds) *Process Mining Handbook. Lecture Notes in Business Information Processing*, vol. 448. Springer, Cham. (2022). [10.1007/978-3-031-08848-3_5](https://doi.org/10.1007/978-3-031-08848-3_5)
8. Hussain, S., De Roo, J., Daniyal, A., Abidi, S.S.R.: Detecting and resolving inconsistencies in ontologies using contradiction derivations. In: 2011 IEEE 35th Annual Computer Software and Applications Conference, pp. 556–561, IEEE (2011)
9. Horty, J.F., Thomason, R.H., Touretzky, D.S.: A skeptical theory of inheritance in non-monotonic semantic networks. *Art. Intell.* **42**(2–3), 311–348 (1990)
10. Hsueh, N.L., Shen, W.H.: Handling nonfunctional and conflicting requirements with design patterns. In: 11th Asia-Pacific Software Engineering Conference, IEEE, pp. 608–615 (2004)
11. Branán, K.: Locality and anti-locality: The logic of conflicting requirements. *Linguist. Inq.* 1–52 (2021)
12. Bozorgi, Z.D., Teinmaa, 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), IEEE, pp. 129–136 (2020)
13. Held, L., Bové, D.S.: *Applied statistical inference*. Springer, Heidelberg, 10, 16 (2014). [10.1007/978-3-642-37887-4](https://doi.org/10.1007/978-3-642-37887-4)
14. Hernán, M.A., Clayton, D., Keiding, N.: The simpson's paradox unraveled. *Int. J. Epidemiol.* **40**(3), 780–785 (2011)



Design-Time Support for Fragment-Based Case Management

Kerstin Andree, Leon Bein^(✉), Maximilian König, Caterina Mandel, Marc Rosenau, Carla Terboven, Dorina Bano, Stephan Haarmann, and Mathias Weske

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
{kerstin.andree,leon.bein,maximilian.konig,caterina.mandel,
marc.rosenau,carla.terboven}@student.hpi.de,
{dorina.bano,stephan.haarmann,mathias.weske}@hpi.de

Abstract. Modeling business processes can be difficult, especially if they are flexible and the models consist of many interconnected parts. This is the case for hybrid process modeling approaches, such as fragment-based case management. Also, erroneous and low-quality models have adverse effects on the whole BPM lifecycle. To support creating fragment-based case models, we provide a set of guidelines and integrate them into a modeling tool called *fcm-js*. Our evaluation shows that *fcm-js* improves the quality of the process model as well as the user experience during modeling.

Keywords: Design-time support · Knowledge-intensive processes · Hybrid process modeling

1 Introduction

Business Process Management (BPM) enables organizations to design, analyze, and enact their business processes [20]. As a central artifact in BPM, *business process models* enable efficient communication between stakeholders and comprehensive analyses of business processes [3] emphasizing the importance of modeling in BPM [13, 20]. However, modeling processes, especially their rules and decisions, is difficult [2, 13]. It takes both experience and expertise to create flawless, accurate models [15] that are low enough in complexity to allow for quick understanding of the process [20].

Guidelines, i.e., “simple rules formulating desired properties of a model” [1], help to ensure consistency and integrity of process models [3]. A modeling tool that automatically checks guidelines improves design-time in terms of usability and comprehensibility of the process logic and structure [2]. For standard process modeling languages, such as Business Process Model and Notation (BPMN) [17], there exist modeling guidelines [1, 15, 16] which are already integrated into an automated verification tool [2]. In contrast, modeling approaches for flexible and knowledge-intensive processes, such as the hybrid approach *fragment-based Case Management* (FCM) [11], lack a valuable design time support, i.e., documentation and up-to-date tooling to create fCM models.

This paper presents a set of fCM modeling guidelines and the modeling tool *fcm-js* that integrates these guidelines. Developed as a human-centered modeling tool, *fcm-js* explicitly supports academic researchers (hereinafter referred to as designers or users) in exploring the fCM approach and creating models of high quality. The different modelers needed for the hybrid notation of the approach can be edited simultaneously, which allows for simple maintenance of process models as well as flexibility in the individual modeling approach. Additionally, users can easily examine decision goals, structures, and their connection with the business process while modeling. Based on the guidelines, an automated model-checking implemented on different levels of integration encourage a trial-and-error learning process through direct feedback, which significantly improves the user experience.

In the remainder of this paper, we introduce related work on design-time support in Sect. 2 and give an introduction to fCM in Sect. 3. Next, the requirements for valuable design-time support for fCM are discussed, before Sect. 4 introduces the modeling guidelines and their integration into the modeling tool *fcm-js*. Both are evaluated in Sect. 5. We conclude with a discussion in Sect. 6.

2 Related Work

Since many BPM tasks rely on process models, model quality is important. This is especially challenging for hybrid approaches that combine multiple models, which must be appropriately integrated. Guidelines and tools can help but are usually language-specific.

The DMN standard for decision modeling complements BPMN. However, inconsistencies between decision and process models can cause errors. Therefore, principles for integrated modeling have been proposed [8].

The data-centric MERODE approach [18] combines data models, event charts, and process models. The different parts depend on each other, for example, the data model constrains the order in which the process can create objects. Therefore, different cross-layer consistency criteria have been defined and incorporated into the MERODE's execution semantics [18].

BAUML is another hybrid approach. It combines a data model, state machines modeling the lifecycles of objects, and activity diagrams refining individual transitions. For BAUML different translations exist to logic and Petri nets to check for internal correctness [4].

For DCR graphs, a highlighter is proposed in [14]. It supports modeling DCR graphs from text to keep text and model consistent even when the model changes.

Many of the presented approaches [4, 14] have in common that they verify the model's behavior, and behavioral verification is computationally expensive. Furthermore, these methods are rarely embedded into modeling tools, i.e., do not support model change. We propose guidelines that include structural consistency and general model quality criteria for the hybrid approach fCM [11]. We integrate them into a modeling tool that provides immediate feedback to the designer and supports model change.

3 Motivation and Requirements

This section motivates the need for design-time support for fragment-based case management (fCM). We first introduce fCM using the example process of criminal justice inspired by [9] and then discuss the challenges of modeling with fCM. Based on this, requirements for valuable design-time support are deduced.

3.1 Fragment-Based Case Management

Fragment-based case management is an actively researched hybrid approach for case modeling [7, 11]. It supports semi-structured, knowledge-intensive processes by structuring them into static process parts (i.e., fragments), which can be flexibly combined during runtime. Knowledge workers determine the exact process execution path based on their own experience, expertise, and available data.

An example use case where fCM is applicable is a simplified version of a criminal justice process [9]. When a person commits a crime and is charged, a criminal trial occurs, and a sentence is announced. It either acquits the defendant or sentences them to imprisonment. Finally, after the sentence is served, the inmate is released and the process terminates. Figure 1 shows an fCM case model of this scenario consisting of four artifacts [6]: (1) a collection of *Fragments* (Fig. 1(a)), i.e., imperative sub-processes modeled using a subset of BPMN [17], (2) a *Data Model* (Fig. 1(b)), i.e., an extended UML class diagram comprising all relevant data classes and the associations between them, (3) *Object Lifecycles* (OLCs) for each data class (Fig. 1(c)), provided as state-transition diagrams, defining how data objects evolve during runtime, and (4) a *Goal State* (Fig. 1(d)), defining the process goal through a set of data objects that are in specific states using the disjunctive normal form.

Aside from the control flow known from BPMN, data flow plays an important role for fCM. In- and outgoing data objects in specific states represent conditions that have to be met before and after the execution of an activity. Combined with the concept of fragments, this allows for additional flexibility. Consider the following scenario: The activity “Pass sentence” has been executed. According to the respective fragment shown in Fig. 1(a), the data object *Sentence* was created, and it was decided whether the *Defendant* is *[innocent]* or *[sentenced to imprisonment]*. Once the data object *Defendant* is in state *[imprisoned]* and the *Sentence* is *[announced]*, the two fragments at the bottom are enabled. The knowledge worker has to decide which fragment to execute since it depends on the concrete case whether requesting a sentence reduction is sensible or not. This also means that enabled activities and fragments are not automatically mandatory and can be skipped. The knowledge workers are in full control of the process.

To ensure proper execution, the four artifacts must be consistent. Consider the data object *Sentence reduction*. By executing the activity *Decide on sentence reduction*, the object transitions from state *[requested]* to *[accepted | rejected]* where the symbol | indicates an exclusive OR. This behavior is also modeled in the respective OLC (Fig. 1(c)) which is assigned to the class *Sentence reduction* in the data model. However, using different modeling languages for the

respective artifacts makes maintaining consistency difficult. Designers often use different modeling tools for the different modeling languages. This hinders immediate verification, and thus keeping consistency between the artifacts becomes challenging.

The more complex and flexible the business process is, the more data classes, state transitions, and fragments need to be modeled. Hidden dependencies cause cognitive load, so designers can lose the overview [5], which results in an inefficient and ineffective verification [2]. Both domain-specific and process modeling expertise is required [1]. In addition, fCM has changed since its first publication [11] and new elements have been defined [6,7]. These are introduced in several papers, which makes modeling more difficult as a compact representation of rules and best practices is missing.

3.2 Requirements

Design-time support ought to improve the quality of a process model in terms of the model's correctness, complexity, and comprehensibility. In addition, it should provide a good user experience while modeling. The requirements for such design-time support for fCM are derived from the general characteristics of the modeling approach [7, 11] as well as from related work on design-time support [1, 2, 15, 16]. Except for the first three requirements, which apply specifically to fCM, each requirement listed below also applies to process modeling in general.

R1 Completeness of Case Model

The fCM case model consists of four explicitly modeled artifacts (Fragments, Object Lifecycles, Data Model, and Goal State) [11].

R2 Correctness within each Artifact

Each artifact is syntactically and semantically correct. This means that only the allowed set of model elements is used to correctly represent the process' content [11].

R3 Consistency between Artifacts

The artifacts are consistent with each other [7], e.g., the goal state is a subset of the states in the OLCs and data classes are used consistently in fragments, data models, and OLCs.

R2 Correctness within each Artifact

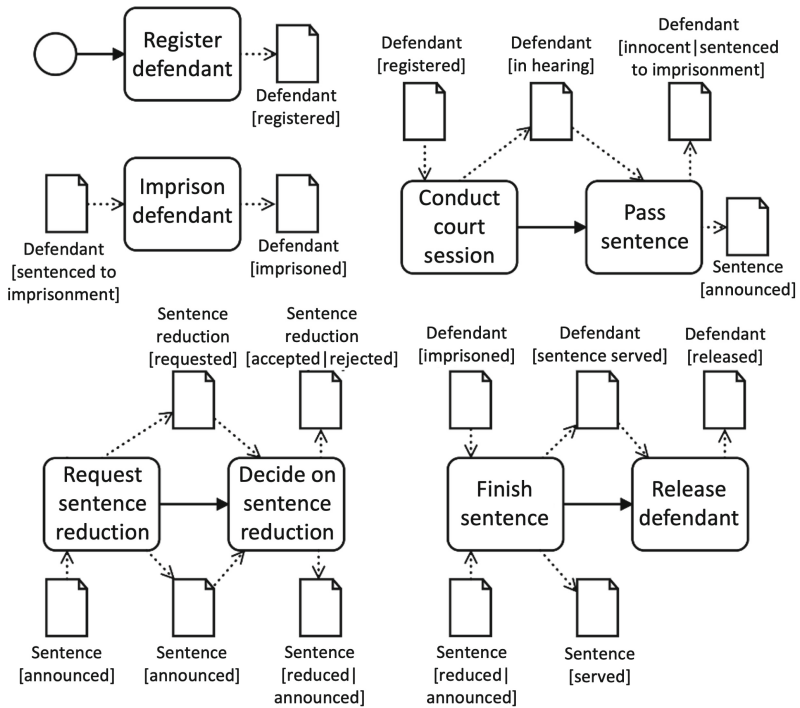
Completeness, correctness, and consistency are checked automatically, making modeling more efficient by relieving designers of the verification of the process model [2].

R5 Visual Modeling

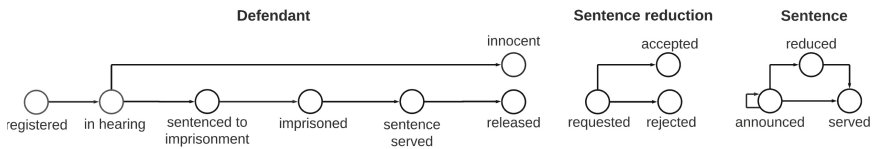
Designers can model visually. Apart from modeling capabilities, no skills (like programming) should be required to model a complete process model [12].

R6 User Flexibility

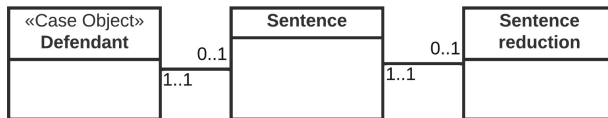
It is up to the designers to choose any modeling method introduced in [10]: process-first, object lifecycle-first, goals-first.



(a) Fragments



(b) Object Lifecycles



(c) Data Model

(Defendant [free]) v (Defendant [released] ∧ Sentence [served])

(d) Goal State

Fig. 1. FCM artifacts of the criminal justice use case

R7 Real-Time Modeling Support

Direct and permanent feedback is provided to support the users in modeling. Model quality issues are directly reported and therefore “guide users, preventing them from modeling defective processes” [19].

R8 Learning Process

Designers are provided with understandable rules and feedback to improve their modeling capabilities [2].

4 Design-Time Support

A high-quality process model is characterized by its precise yet comprehensible representation [20]. At design-time, designers can be supported by guidelines which provide suggestions for modeling. A tool integrating the guidelines can further improve a model’s quality by checking guidelines automatically.

In this section, we present (i) guidelines for fCM, (ii) a novel fCM modeling tool, and (iii) how it integrates the guidelines.

4.1 Guidelines

The 36 guidelines¹ are grouped in six categories: Fragments (12), Object Life-cycles (5), Data Models (7), Goal States (2), Consistency (7), and General Guidelines (3). The categorization preserves the flexibility of designers, since it allows for a precise lookup of guidelines depending on the artifact they currently work on. Each guideline consists of three required components (*ID*, *Name*, and *Description*) and three optional components (*Example*, *Motivation*, and *References*). Because of the ties of fCM to BPMN, this framework is based on [2]. We extend the approach by a motivation component that is also used in [16], since fCM often offers multiple variants for modeling that should be distinguished.

Table 1. General structure of a guideline illustrated with an example

mandatory	ID	F6
	Name	Use start events only in initial fragments
	Description	The designer should use start events to indicate initial fragments of the process which create the case class data object. These fragments can only be executed once within the same case
optional	Motivation	Originally, start events were used for each fragment [11]. However, that way they do not add any semantics to the process model because a fragment is enabled if the data preconditions are fulfilled
	Example	The process shown in Fig. 1(a) can only start with registering the defendant. Therefore, the case model has only one initial fragment
	References	[6]

¹ The guidelines are published on GitHub: <https://github.com/bptlab/fCM-design-support/wiki>.

Table 1 provides an example guideline dealing with start events². The *ID* is a unique identifier for each guideline. It consists of a letter indicating the respective category of guidelines and an ordinal number. In this case, F6 refers to the sixth guideline for fragments. A concise and descriptive *Name* gives a first impression of what the guideline is about, whereas the *Description* explains the content of the guideline in more detail. A *Motivation* explains the background of a guideline, which is especially helpful if fCM allows modeling the same behavior in multiple ways. The guideline shown in Table 1 notes that there was a change in the use of start events compared to the original version of fCM. An *Example* provides a short explanation or excerpts of an exemplary case model³ highlighting the relevant elements. Cited *References* can be used to obtain further information.

Since maintaining consistency between artifacts is one of the biggest challenges, we exemplarily discuss the concrete consistency aspect regarding the reachability of the goal state and explain how our guidelines support designers.

Table 2. Consistency guideline C1: make the goal state a subset of the OLCs

ID	C1
Name	Make the Goal State a Subset of the OLCs
Description	The designer should assure that all states included in the goal state are also included in the corresponding OLCs
Example	Consider the use case of criminal justice (cf. Fig. 1). Each state included in the goal state is also defined in the respective OLCs
References	[6]

In order to reach the goal state and to enable closing the case, certain data objects have to be in specific states, which is accomplished through executing activities in fragments. In addition, all state transitions performed by activities have to be modeled in the respective OLC. Thus, ensuring the reachability of the goal state requires consistency between three artifacts. This is covered by the guidelines C1 (see Table 2) and C3 (see Table 3). While C1 states that all data object states used in the goal state have to be defined in the OLCs, C3 establishes that each state transition modeled in fragments must have a corresponding transition in the OLCs.

4.2 Tool

The modeling tool *fcm-js*⁴ focuses on design-time support for fCM and is a web application mainly written in HTML and JavaScript, which allows a fast setup and good accessibility. Its key feature is the integration of movable visual

² <https://github.com/bptlab/fCM-design-support/wiki/Fragments#f6>.

³ We use the example use case of criminal justice to provide examples for the guidelines: <https://github.com/bptlab/fCM-design-support/wiki/Example-Use-Case---Criminal-Justice>.

⁴ Repository with source code and screencast at <https://github.com/bptlab/fCM-design-support>.

Table 3. Consistency guideline C3: use state labels and state transitions of data objects consistently in OLCs and fragments

ID	C3
Name	Use State Labels and State Transitions of Data Objects consistently in OLCs and Fragments.
Description	Each state label used for data objects in fragments should also be modeled as a state in the corresponding OLC. The designer should ensure that each state transition in the fragments corresponds to a state transition in the related OLC.
Example	The activity “Imprison defendant” performs a state transition on the data object <i>Defendant</i> from state <i>[sentenced to imprisonment]</i> into state <i>[imprisoned]</i> . This state transition is also encoded in the respective OLC (see Fig. 1).
References	[6, 11]

modelers for all four artifacts (cf. Fig. 2) which can be accessed at the same time. The modelers are based on the open-source projects bpmn-js⁵ and diagram-js⁶. A mediator component with access to all artifacts enables indirect communication between the modeling components and thus guideline checking across multiple artifacts. Lastly, we implemented a checking component with a unified guideline interface. Every guideline is checked automatically, and the user receives instant feedback as violating elements are highlighted (see Fig. 2), and quick fixes are proposed.

4.3 Integration

For the integration of guidelines into a software modeling tool, we define four levels targeting different requirements:

1. **No Integration:** The guidelines are not automatically checked but provided in an extra publication or catalog, which can be linked in the tool. This level includes highly case-dependent guidelines and fulfills the requirements *User Flexibility* (R6) by not imposing restrictions on the designer, and it supports the *Learning Process* (R8) by providing reading material.
2. **Low Integration:** Low integration includes automatically executed guideline checks (*Automated Verification* (R4)) which enable immediate feedback and direct interaction by highlighting violations and displaying violation messages which indicate what is wrong and how to fix it (*Real-Time Modeling Support* (R7) and trial-and-error *Learning Process* (R8)).
3. **Medium Integration:** Medium integration directly extends low integration by adding the suggestion and appliance of quick fixes, focusing on better support for R7 and R6.
4. **Full Integration:** Fully integrating a guideline means that models created with the tool inherently fulfill that guideline (R7).

For each guideline, we decided at which level it should be integrated, based on desirability and user study results described in Section 5 rather than technical possibility.

⁵ <https://github.com/bpmn-io/diagram-js>.

⁶ <https://github.com/bpmn-io/bpmn-js>.

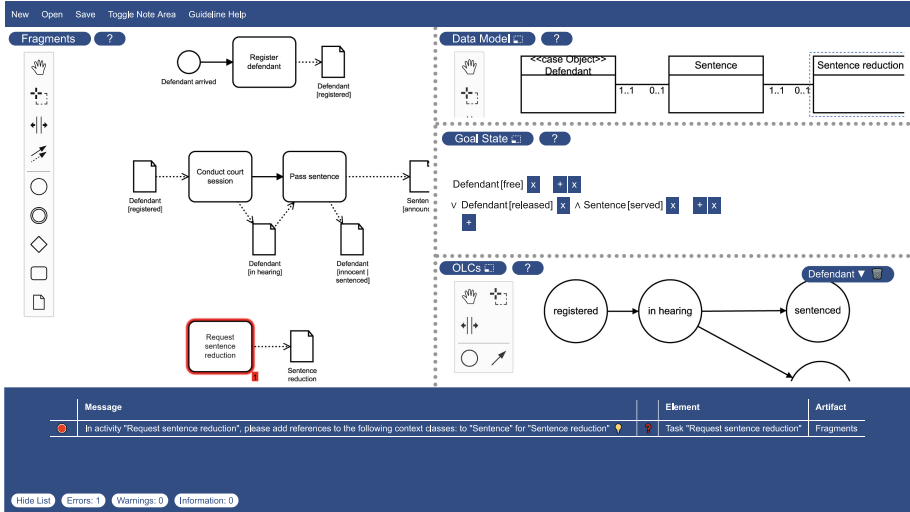


Fig. 2. Screenshot of *fcm-js* user interface

5 Evaluation

For optimal usage, the guidelines must be comprehensible and provide support. Integrated into a modeling tool, they must also significantly improve model quality and should enhance the user experience while modeling. This section presents the results from the evaluation of the guidelines and the modeling tool *fcm-js*.

Comprehensibility and Sufficient Support. A user study⁷ was conducted to obtain feedback from 14 participants with varying fCM expertise to assess the guidelines' comprehensibility and support for modeling. In addition, participants were asked about the specific integration of each guideline into a modeling tool according to their individual preference.

The results show that the guidelines are well understood: 74% of the responses indicate good understanding, 18% sufficient understanding. Only 8% were neutral or stated the guidelines as difficult to understand. Support for modeling was rated as excellent by 32% of the responses. 46% indicated good support, while 3% indicated little or no support, respectively. The desired integration level for each guideline is evenly distributed with no a clear connection between fCM expertise and desired level.

Model Quality. The impact of the guideline integration on process model quality was examined by comparing two groups of 5 competent fCM users, one

⁷ <https://forms.gle/MvJckrJ71zw59CJ6A>.

using *fcm-js* and the other using pen and paper. Both groups were asked to model an fCM case model based on a textual description. For each group, we checked the guideline coverage of the resulting models. Process models created with pen and paper covered only 23.6 of 36 guidelines on average (66% coverage) whereas models created using *fcm-js* covered 30.2 guidelines on average (84% coverage). Thus, model quality benefits from a good integration of guidelines.

User Experience. User experience refers to requirements R6, R7, and R8 and is an important indicator for dealing with complexity, especially in modeling approaches for flexible processes such as fCM. Think aloud sessions were conducted, in which five competent fCM users were observed while modeling an fCM case model using *fcm-js*. Following that, an additional interview provided insights into learning achievements.

The results show that participants were more confident in modeling with *fcm-js* compared to previous modeling procedures. According to participants, automatically performed consistency checks and a user interface specifically designed for fCM reduce complexity and provide a sense of certainty regarding model quality. Furthermore, they felt supported in maintaining hidden dependencies.

6 Discussion and Conclusion

This section discusses the guidelines, their integration into *fcm-js*, and the results of the evaluation concerning the requirements for valuable design-time support. Table 4 shows an overview of the requirements coverage and is discussed below.

Table 4. Examination of *fcm-js* based on requirements. Rating uses the following scale: Requirement not met/partially met/completely met (-/o/+).

Requirement	Rating	Explanation
R1 Completeness	+	<i>fcm-js</i> provides a modeling environment for each artifact
R2 Correctness	o	<i>fcm-js</i> only partially supports semantic correctness
R3 Consistency	o/+	<i>fcm-js</i> intensively supports consistency, but the final model is based on modeling decisions of the designer
R4 Automated verification	o	<i>fcm-js</i> automatically checks many guidelines, but the verification of behavioral characteristics is missing
R5 Visual modeling	+	each artifact can be modeled visually
R6 User flexibility	o/+	Process-first and OLC-first case modeling methods are fully supported, but the goals-first approach only partly
R7 Real-Time support	+	Support with four levels of integration
R8 Learning process	o/+	Great results in evaluation, but no long-term user study

In general, *fcm-js* allows modeling all artifacts (R1) visually (R5) in their respective modeling language. The guidelines cover correctness within each artifact (R2) in terms of its syntax and execution semantics. However, the latter is not fully supported by *fcm-js*, as we only verify the structure of the case model. Domain-specific semantics are neither covered by the guidelines nor our modeling tool. Nevertheless, consistency guidelines (R3) are supported by *fcm-js*.

The guidelines, as well as the integration, are currently up-to-date but have to be regularly revised in the future to ensure that R2 and R3 are still met, since fCM is a research approach likely to expect future updates. All guidelines that are integrated at the low level or higher are automatically checked (R4) and provide immediate feedback to the user (R7) or are fully integrated in a way that does not allow the user to create violations. However, the 20 currently integrated guidelines mostly consider structural model properties. Behavioral characteristics, such as the reachability of the goal state, are not yet integrated into *fcm-js* due to the increased complexity of monitoring behavioral properties compared to structural properties. Behavioral model checkers are used to handle this complexity and could be integrated into *fcm-js* with further research. Having all modelers accessible simultaneously enables designers to use the modeling method that best fits their use case and individual preferences (R6). The good comprehensibility of the guidelines supports designers in terms of extending their modeling knowledge (R8) and *fcm-js* allows trial-and-error learning.

In this paper, we presented requirements of valuable design-time support for fCM, a set of 36 guidelines for fCM modeling, and a web-based modeling tool for fCM called *fcm-js* in which 20 guidelines are integrated on different levels. While the guidelines and *fcm-js* are language specific, the general approach and architecture can be applied to other hybrid process modeling languages. Consistency and correctness criteria can be checked and enforced by a modeling tool, providing real-time support for process designers.

References

1. 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. Manage. J.* **27**(1), 1–23 (2021). <https://doi.org/10.1108/BPMJ-10-2019-0407>
2. Corradini, F., et al.: A guidelines framework for understandable BPMN models. *Data Knowl. Eng.* **113**, 129–154 (2018). <https://doi.org/10.1016/j.datak.2017.11.003>
3. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*, 2nd edn. Springer, Cham (2018). <https://doi.org/10.1007/978-3-662-56509-4>
4. Estañol, M., Sancho, M.-R., Teniente, E.: Verification and validation of UML artifact-centric business process models. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) *CAiSE 2015*. LNCS, vol. 9097, pp. 434–449. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19069-3_27
5. Green, T.R.G., Petre, M.: Usability analysis of visual programming environments: a ‘cognitive dimensions’ framework. *J. Vis. Lang. Comput.* **7**(2), 131–174 (1996). <https://doi.org/10.1006/jvlc.1996.0009>

6. Haarmann, S., Montali, M., Weske, M.: Refining case models using cardinality constraints. In: La Rosa, M., Sadiq, S., Teniente, E. (eds.) CAiSE 2021. LNCS, vol. 12751, pp. 296–310. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79382-1_18
7. Haarmann, S., Weske, M.: Correlating data objects in fragment-based case management. In: Abramowicz, W., Klein, G. (eds.) BIS 2020. LNBIP, vol. 389, pp. 197–209. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53337-3_15
8. Hasic, F., Smedt, J.D., Vanthienen, J.: Augmenting processes with decision intelligence: principles for integrated modelling. *Decis. Support Syst.* **107**, 1–12 (2018). <https://doi.org/10.1016/j.dss.2017.12.008>
9. van Hee, K., Serebrenik, A., Sidorova, N., Voorhoeve, M., van der Werf, J.M.: Modelling with history-dependent petri nets. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 320–327. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_23
10. Hewelt, M., Pufahl, L., Mandal, S., Wolff, F., Weske, M.: Toward a methodology for case modeling. *Softw. Syst. Model.* **19**(6), 1367–1393 (2019). <https://doi.org/10.1007/s10270-019-00766-5>
11. Hewelt, M., Weske, M.: A hybrid approach for flexible case modeling and execution. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNBIP, vol. 260, pp. 38–54. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45468-9_3
12. Kunze, M., Weske, M.: Signavio-oryx academic initiative. In: Proceedings of the Business Process Management 2010 Demonstration Track, Hoboken, NJ, USA, 14–16 September 2010. CEUR Workshop Proceedings, vol. 615. CEUR-WS.org (2010). <http://ceur-ws.org/Vol-615/paper6.pdf>
13. Leopold, H., Mendling, J., Günther, O.: Learning from quality issues of BPMN models from industry. *EMISA Forum* **36**(2), 120–123 (2016)
14. López, H.A., Debois, S., Hildebrandt, T.T., Marquard, M.: The process highlighter: from texts to declarative processes and back. In: Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018, Sydney, Australia, 9–14 September 2018, pp. 66–70 (2018). http://ceur-ws.org/Vol-2196/BPM.2018.paper_14.pdf
15. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven process modeling guidelines (7PMG). *Inf. Softw. Technol.* **52**(2), 127–136 (2010). <https://doi.org/10.1016/j.infsof.2009.08.004>
16. Moreno-Montes de Oca, I., Snoeck, M.: Pragmatic guidelines for business process modeling (2014). Available at SSRN 2592983
17. OMG: Business process model and notation BPMN v. 2.0 (2011)
18. Snoeck, M.: Enterprise Information Systems Engineering - The MERODE Approach. The Enterprise Engineering Series, Springer, Cham (2014). <https://doi.org/10.1007/978-3-319-10145-3>
19. Steinau, S., Andrews, K., Reichert, M.: A modeling tool for PHILharmonicFlows objects and lifecycle processes. In: Proceedings of the BPM Demo Track and BPM Dissertation Award 2017, Barcelona, Spain, 13 September 2017. CEUR Workshop Proceedings, vol. 1920. CEUR-WS.org (2017). http://ceur-ws.org/Vol-1920/BPM_2017_paper_196.pdf
20. Weske, M.: Business Process Management - Concepts, Languages, Architectures, 3rd edn. Springer, Cham (2019). <https://doi.org/10.1007/978-3-662-59432-2>



Process Mining Meets Statistical Model Checking: Towards a Novel Approach to Model Validation and Enhancement

Roberto Casaluca^{1,2}, Andrea Burattin³, Francesca Chiaromonte^{2,4},
and Andrea Vandin^{2,3}(✉)

¹ University of Pisa, Pisa, Italy

² Institute of Economics and EMbeDS, Sant'Anna School of Advanced Studies,
Pisa, Italy

andrea.vandin@santannapisa.it

³ DTU Technical University of Denmark, Kgs. Lyngby, Denmark

⁴ Department of Statistics and Huck Institutes of the Life Sciences,
Penn State University, Pennsylvania, USA

Abstract. We propose a novel research line integrating Statistical Model Checking (SMC), a family of simulation-based analysis techniques from quantitative formal methods, with Process Mining (PM), a collection of data-driven process-oriented techniques. SMC and PM are complementary. SMC focuses on performing the *right number* of simulations to obtain statistically-reliable estimations (e.g., the probability of success of an attack). PM focuses on reconstructing a model of a system using logs of its traces. Nevertheless, both approaches aim at providing evidence of issues/guarantees of the system, and at proposing enhancements.

We aim at enriching SMC by *explaining why* it produced specific estimates. This might help, e.g., identifying issues in the model (validation) or suggesting improvements (enhancement). Given that SMC uses statistics to decide what is the *correct* number of simulations (or *traces*), we avoid by-construction the complex issue of under-representation of system behavior in the logs crucial to many PM exercises.

This work-in-progress paper demonstrates the proposed methodology and its usefulness using a simple example from the security threat modeling domain. We show how PM helps highlighting both mistakes in the model, and possibilities for improvement.

Keywords: Process mining · Statistical model checking · Validation

1 Introduction

We present novel research integrating the simulation-based analysis technique from quantitative formal methods known as statistical model checking (SMC) [2], with the data- and process-driven techniques known as process mining (PM) [1].

Specifically, we aim at formulating a novel framework capable of making analyses results typical of SMC, and of simulation-based analysis in general,

more explainable and understandable. Our framework will empower modelers to actually *see* the unfolded behavior of their models, as opposed to just numerical aggregated values. This will pave the way to new possibilities in terms of debugging and validating the models of interest. Considering the widespread use of simulation models, having tools to debug and validate them will represent a potentially very impactful contribution among several disciplines.

2 Preliminaries and Related Work

2.1 Attack-Defense Trees

We briefly introduce the domain of risk modeling and analysis with so-called *attack-defense trees*. Attack-defense trees (ADT) and their variants [13, 16, 19] allow to represent security scenario by means of intuitive visual language constructs. These aim at providing means for specifying vulnerabilities and countermeasures, their interplay, together with quantitative aspects such as cost and effectiveness. The goal is to support policy- and decision-making in determining, e.g., the degree of vulnerability to specific attacks, based on the resources of attackers, or whether given defensive resources are cost-effective. Attack trees are widely used in several domains like, e.g., defense (e.g., their use is recommended by NATO [18]), aerospace [22], or safety-critical cyber-physical systems [12].

We present a simple running example for the risk assessment of a “bank robbery” scenario extrapolated from [5] as depicted in Fig. 1. We see that ADT are graphs whose nodes represent either attack goals or defensive measures, and sub-trees represent nodes’ refinements. The root (RobBank), is the threat under analysis. In order to achieve it, we shall first succeed in opening the vault (OpenVault), or blowing it up (BlowUp), or both. Refinements might also come with other Boolean conjunctions like *and*, *xor*, etc. The figure also shows another kind of refinement, denoted by a grey dashed edge: RobBank attempts can be mitigated by the countermeasure LockDown. This is a reactive defense that might be turned on by BlowUp attempts (dashed blue arrow).

Recently (e.g., [3, 4, 14]), ADT have been extended with *attacker profiles*, explicit attacker behaviours acting on the security scenario described by the ADT. This allows to assess systems against specific classes of attackers (e.g., large organizations with rich resources, or lone wolves). E.g., RisQFLan [4] allows one to specify *probabilistic attacker behaviors*. An example is shown in Fig. 2. We can see that the attacker can be in four *states*: Start, TryOpenVault, TryBlowUp, and

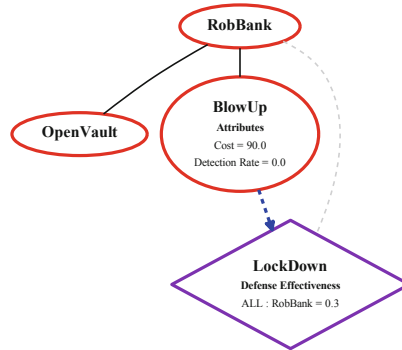


Fig. 1. Attack-defense Tree for a simple bank robbery scenario (Color figure online)

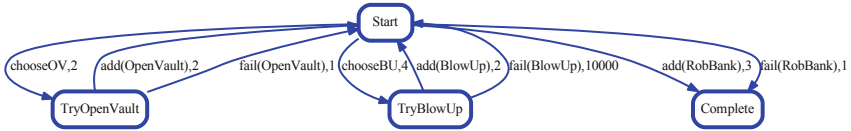


Fig. 2. Probabilistic attacker behaviour

Complete. When s/he is in state `Start`, s/he can decide to attempt `OpenVault` or `BlowUp` strategies by changing in states `TryOpenVault`, or `TryBlowUp`, respectively. This is done by performing *action* `chooseOV` with weight 2, or `chooseBU` with weight 4, respectively. Alternatively, if *allowed* by the ADT (i.e., if `OpenVault` or `BlowUp` attempts have already previously succeeded), the attacker can actually attempt to rob the bank by moving in state `Complete`. In particular, the attack will succeed (`succ(RobBank)`) with weight 3, or `fail(RobBank)` with weight 1. Likewise, from state `TryOpenVault` or `TryBlowUp`, the corresponding attacks will be attempted as dictated by the transitions with actions `succ` and `fail`. We note that the weight for `fail(BlowUp)` is particularly high, this will be instrumental to our analyses in Sect. 4.

The weights are used to compute the probability with which each transition is executed, allowing to obtain probabilistic simulations of the attacker behaviour modulo the constraints by the ADT. More details on the `RisQFLan` language, on the further quantitative attributes of nodes shown in Fig. 1, and on the simulation of `RisQFLan` models will be provided in Sect. 4.

2.2 (Black-Box) Statistical Model Checking

We introduce a family of simulation-based analysis techniques that comes under the name of (black-box) statistical model checking (SMC). SMC [2], consists in performing a *sufficient* number of probabilistic simulations of a model to obtain *statistically reliable estimations* of model properties. Black-box SMC (e.g., [21, 24, 25]) is a fragment of SMC where no assumption is made on the studied model, only that probabilistic simulations of it can be performed. E.g., MultiVeStA [10, 20, 24] is a black-box SMC tool that can be integrated with existing simulators to enrich them with automated statistical analysis techniques. The idea is simple: to each simulation we assign a real value, e.g., 0 or 1 if the attacker succeeds or fails, resp., in robbing the bank within the first 60 simulation steps. Technically, this is a random variable X in the interval $[0, 1]$ over simulations of the model. Notably, the expected value $p = E[X]$ of X corresponds to the probability of success of the event (the attacker succeeds within the first 60 steps of simulation). As discussed in [20, 24], MultiVeStA estimates such expected values $E[X]$ as the mean \bar{x} of n independent simulations, with n large enough to guarantee an (α, δ) *confidence interval* (CI) [15, Chapter 9]. In other words, MultiVeStA guarantees that $E[X]$, i.e., the studied probability, belongs to the interval $[\bar{x} - \delta/2, \bar{x} + \delta/2]$ with statistical confidence $(1 - \alpha) \cdot 100\%$. δ is a parameter chosen by the user that gives a sort of *precision* on the performed estimation. Instead, roughly, α is related to the probability that the studied probability actually belongs to the

computed interval. The interesting part is that the *correct* number of simulations to be performed for a given property and CI is chosen by MultiVeStA by using standard statistical machinery [15, Chapter 9].

Black-box SMC and MultiVeStA can estimate more complex properties. However, for the sake of presentation, here we focus on simple ones like the mentioned one. MultiVeStA has been successfully applied to a wide range of domains, including, e.g., security risk modeling [4], economical agent-based models [24], highly-configurable systems [5, 23], public transportation systems [8, 11], business process modeling [9]. Robotic scenarios with planning capabilities [6] and crowd steering scenarios [17]. This has been made possible by the fact that black-box SMC can be applied to virtually any simulation model. E.g., MultiVeStA only requires to implement a simulator-specific adaptor to perform basic operations such as `reset`: reset simulator to do a new simulation, providing a new random seed; `oneStep`: perform one step of simulation; and `eval`: evaluate an observation on the current simulator state. However, this high generality might come at the cost of low interpretability of results. Indeed, it might be difficult to provide *behavioral explanations* about *why* a property is estimated to a given value. SMC approaches like, e.g., [7], partially address this by providing a counterexample – an example of *problematic/relevant* simulation. However, to the best of our knowledge, no SMC technique has ever been integrated with rich support for describing and reasoning upon the *whole behavior* that led to a specific result. It is relevant to highlight that the methodology presented in this paper can be applied to any simulation model and SMC tool, since it does not rely on the internal mechanics of the analyzer or of the model, but exploits logs of the computed simulations. In the next section we present a process-oriented data-driven family of techniques known as process mining that we combine with SMC to offer rich *white-box behavioral* interpretability of the analysis results.

2.3 Process Mining

Process Mining (PM) is the scientific discipline bridging the gap between data science and process science [1]. It aims at using actual executions of a process to infer relevant information about how the underlying behavior is observed in action (as opposed to the intended behavior). PM consists of three main activities: (control-flow) discovery, enhancement, and conformance checking. *Discovery* aims to identify an abstract representation of the executed process by combining all the observed instances into a single model. Such a model can be enhanced with additional information (e.g., how often activities/paths are executed). This is called *enhancement*. Finally, *conformance checking* tries to understand the extent to which a normative model is violated in actual executions.

In this paper, we are interested in the discovery and enhancement activities since we aim at consuming the execution traces coming from SMC analyses to see whether the generated behavior is aligned with the original expectations.

3 Extending (Black-Box) SMC with Process Mining

In this section we discuss a novel methodology for white-box behavioral explanation of SMC analysis results by process mining.

3.1 Methodology

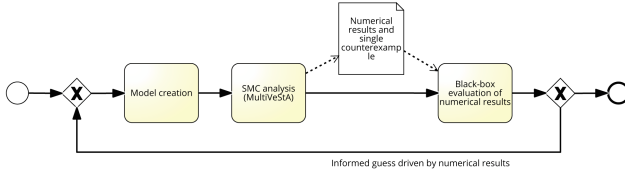
Figure 3a graphically depicts the typical methodology adopted for the construction and validation of simulation models based on SMC analysis. It starts with the creation of the model, which is then analyzed producing numerical results, plots, and possibly counterexamples for the studied properties. At this point, if the modeler spots unexpected numerical results, e.g., the probability of success of an attack is lower than expected, they might make an informed guess to hypothesize changes to be applied to the model to improve its performance and *fix* it. This can be seen as an *SMC-guided black-box validation* of the model (we use the term black-box, because there is no overview of the overarching behavior, but just numerical values or single counterexamples).

Figure 3b depicts a new methodology we propose which enriches (black-box) SMC with *SMC- and PM-guided white-box behavioral model validation*. The main idea is to augment SMC, e.g., MultiVeStA, with the ability to generate logs for the computed simulations, corresponding to one *trace* per simulation (here we use the term *trace* to indicate one sequence of observed events, all belonging to the same process instance). Once the analysis is concluded, we feed these logs to existing PM tools, enabling a behavioral interpretation of the analysis results. Indeed, the output of PM tools can be visually inspected to evaluate the behaviors originating from the collected instances of the attacker behavior (the simulations). Therefore, the result is not anymore given as raw numbers, which require an important cognitive step for *debugging* the model and discovering if and where something went wrong. Rather, PM provides interactive and navigable (via abstraction and filtering of traces and connections) graphical results which are closer and more comparable to the original model. This better supports and guides the identification of flaws or enhancement opportunities.

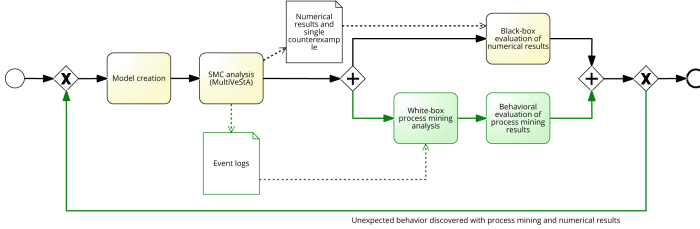
3.2 Operationalizing the Methodology

In order to enrich MultiVeStA with PM-oriented logging capabilities we have extended its interface with simulators, and in particular the RisQFLan's, with two methods, namely `createLogFile` (invoked once per analysis by MultiVeStA to create an empty log file); and `addLogRow` (invoked by MultiVeStA every time we want to note down an *event of interest*). In principle, MultiVeStA could be instructed in invoking `addLogRow` once or several times per simulation step, or only when specific events arise. This would give freedom to the modeler in deciding the coarseness-degree of the logs. Currently, we invoke the method `addLogRow` once per simulation step, recording:

- The incremental counter of steps (the *time stamp*);



(a) State-of-the-art life-cycle of SMC-analysed simulation models.



(b) Novel SMC- and PM-guided methodology for white-box behavioral model validation. Additions wrt Fig. 3a are shown in green.

Fig. 3. SMC- and PM-based methodology for white-box behavioral validation. (Color figure online)

- The unique random seed used by the current simulation (the *case ID*);
- The executed action (the *activity*);
- The target state of the executed transition (also part of the *activity*);
- Relevant additional information, e.g., the achievement of (sub)-attack attempts, the activation status of countermeasures, and further information on the obtained simulation state.

The set of information recorded are then structured in a file that can be used for the process mining analyses. Indeed, in order to analyse a log with process mining tools, it is necessary to have: (i) a case identifier (i.e., case ID) which is used to group together all the observations belonging to the same process instance (i.e., the simulation number); (ii) an activity name, indicating which action has been executed in each event; (iii) a timestamp (used for sorting the events and producing a sequence). Additional attributes can also be used in order to refine the analyses, for example, by applying some filters to focus only on successful attacks or to remove executions involving some paths/actions. For the actual process mining analyses we used Fluxicon Disco (<https://fluxicon.com/disco/>) which allows importing CSV files and provides very useful tools for interactive navigation and exploration of the results.

4 Experiments

Here we apply our proposed methodology to the discussed case study. In our experiments, we use Fluxicon Disco to study the generated event logs of the attacks. We investigate how the integration of SMC (MultiVeStA) and PM

(Disco) allows us to create two model refinements improving the *performances* of the attacker, and fixing *mistakes* in the model. We will use SMC to analyze the original model and the two devised refinements with respect to the following property:

What is the probability for the attacker to succeed in a robbing the bank within 60 steps of simulation?

The property is purposely simple. We chose 60 steps because we found it sufficient to cover the dynamics of the model. The results obtained by for the three variants are given in Fig. 4. In all the three variants, we used $\alpha = 0.1$, $\delta = 0.1$, while MultiVeStA decided to run 240 simulations.

We start discussing how the bank robbery case study can be encoded in RisQFLan.

4.1 RisQFLan Encoding of the Running Example

Figure 1 and 2 were generated by RisQFLan starting from a textual description which we detail here. The RisQFLan code for the tree-structure of the ADT itself from Fig. 1 is straightforward, therefore we do not discuss it here. The nodes in Fig. 1 contain quantitative information like *Cost*, *Detection rate*, and *Defense effectiveness*. These are given in *RisQFLan code blocks* shown in Fig. 5. The `attributes` block specifies quantitative attributes of nodes. In the example we defined only *Cost*, which specifies that the cost of `BlowUp` and `OpenVault` attempts is 90 and 0, resp. This means that every time `succ(BlowUp)` or `fail(BlowUp)` are executed, a counter *Cost* is increased by 90. This has a strong impact on the allowed simulations. Indeed, the block `quantitative constraints` imposes that, *no matter what*, the value of such counter will never be allowed to be greater than 100. Practically, this imposes that the attacker will only be allowed to attempt at most one `BlowUp` attempt per simulation. This is obtained as follows: at every simulation step, the weight (and therefore the probability of execution) of actions violating constraints is scaled down to 0. Block `attack detection rates` specifies that `BlowUp` attempts have probability 0 of being detected. This de facto imposes that the countermeasure `LockDown` will never be activated. We omit discussing *Defense effectiveness* because irrelevant to this paper.

<pre> 1 begin attributes 2 Cost = {BlowUp = 90, OpenVault = 0} 3 end attributes 4 5 begin quantitative constraints 6 { value(Cost) <= 100 } 7 end quantitative constraints </pre>	<pre> 1 begin attack detection rates 2 BlowUp = 0.0 3 end attack detection rates </pre>
--	--

Fig. 5. RisQFLan: (Left) Nodes’ attributes and quantitative constraints. (Right) Detection rates of attack nodes.

Model	Prob.
Original	0.17
1 st refinement	0.31
2 nd refinement	0.72

Fig. 4. Probability of bank robbery.

```

1  begin attacker behavior
2  begin attack
3  attacker = Thief
4  states = Start, TryOpenVault, TryBlowUp, Complete
5  transitions =
6  // If vault open or blown up, attacker can rob
7  Start - (succ(RobBank), 3, allowed(RobBank)) -> Complete,
8  Start - (fail(RobBank), 1, allowed(RobBank)) -> Complete,
9  // Strategy where the attacker can open the vault
10 Start -(chooseOV, 2) -> TryOpenVault,
11     TryOpenVault -(succ(OpenVault), 2) -> Start,
12     TryOpenVault -(fail(OpenVault), 1) -> Start,
13 // Strategy where the attacker tries to blow up the vault
14 Start -(chooseBU, 4) -> TryBlowUp,
15     TryBlowUp -(succ(BlowUp), 2) -> Start,
16     TryBlowUp -(fail(BlowUp), 10000) -> Start
17 end attack
18 end attacker behavior

```

```

1  begin actions
2  chooseOV
3  chooseBU
4  end actions

```

Fig. 6. RisQFLan: Probabilistic attacker behavior.

We now move our attention to the probabilistic attacker behavior in Fig. 2, given by the block in Fig. 6. We note an almost one-to-one correspondence among Fig. 2 and Fig. 6. Looking at the first transition, we see how those are given by a source (`Start`) and target state (`Complete`), the executed action (`succ(RobBank)`), the weight (3), and an optional *transition constraint* (a guard) that blocks the execution of the transition if not satisfied. In this case, `allowed(RobBank)` imposes that we can attempt `RobBank` attacks only if allowed by the constraints from the ADT. This models a sort of *smart* attacker that does not waste resources in attempting the attack until s/he knows to have chances of success. Indeed, the semantics of RisQFLan imposes that, without such guard, only `fail(RobBank)` (and not `succ(RobBank)`) would be allowed to execute if the constraints coming from the ADT are not satisfied [4]. Finally, in the figure we also see that the modeler can add model-specific actions on top of the `succ` and `fail` ones.

4.2 Analysis and Refinement of the Original Model

Behavioral Analysis of Results. As shown in Fig. 4, in the original model the attacker has only 0.17 probability in succeeding in its attack. By feeding the obtained logs into Disco (with filters set to 100% for both the activities and the path), we obtain the process in Fig. 7. We can see that the starting state is `Start-reset`, where `Start` is the initial state of Fig. 2, while `reset` is a special action implicitly execute by MultiVeStA when resetting the simulator before performing a new simulation. For example, we have a transition from the root state to `TryBlowUp-chooseBU`, labeled 167, to denote that 167 times out of 240 the attacker attempts a `BlowUp` attack. This is about twice the number of times (73) that `OpenVault` is chosen as first attack, coherently with the weights of the transitions in Fig. 6 with action `chooseOV` (2), and `chooseBU`, respectively.

Let us look at the process state `Complete-succ(RobBank)` at the bottom-left of Fig. 7. No other process state contains `succ(RobBank)`. Therefore, all and only the simulations reaching this configuration represent executions in which the attacker succeeded in robbing. The label 41 in the process state denotes that this happens 41 times. Indeed, $41/240 = 0.17$, the probability given in Fig. 4. At the bottom-left of Fig. 7 we also find state `Start-succ(OpenVault)`. Once we get into such state, we have succeeded in attacking `OpenVault` (the attacker successfully completed the left-most strategy in Fig. 2, and went back in state `Start`). According to the constraints expressed in the ADT in Fig. 1, the attacker could now attempt `RobBank` attacks. This is because the ADT requires that at least one among `OpenVault` or `BlowUp` is necessary.

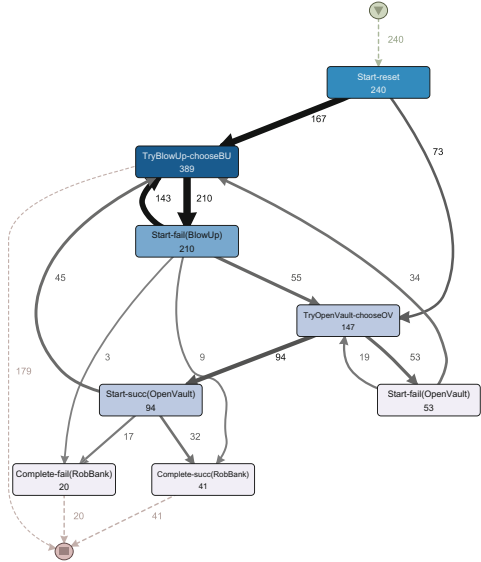


Fig. 7. Process model generated by Disco for the original model.

Indeed, the two downward process transitions outgoing from process state `Start-succ(OpenVault)` attempt the root attack, failing 17 times, and succeeding 32 times. Instead, the upward transition entering in `TryBlowUp-chooseBU` depicts that in many cases, 45, the attacker decided to first attempt also a `BlowUp` attack. This is permitted by the model and the ADT. However, it might be somehow unexpected or irrational by the attacker. Clearly, this will lead to higher expenses (`BlowUp` attempts have high cost), and lower success rate (due to `fail` actions possibly execute).

Model Refinement. Leveraging the analysis in Disco we discovered that the original model erroneously, or at least unexpectedly, describes a sort of *non parsimonious* attacker that attempts non-necessary `BlowUp` attacks even if they already succeeded in `OpenVault` ones. We can solve this issue by changing the transition with action `chooseBU` from Fig. 6 as shown in Fig. 8. The fix is simple: we only need to add a guard `!allowed(RobBank)`. As we shall see in the next

```

1 // We add the !allowed(RobBank)
2 Start -(chooseBU, 4, !allowed(RobBank)) -> TryBlowUp,
3 TryBlowUp -(succ(BlowUp), 2) -> Start,
4 TryBlowUp -(fail(BlowUp), 10000) -> Start

```

Fig. 8. First model refinement: we fix the *non-parsimonious attacker* problem.

section, this guarantees that the attacker will not attempt `BlowUp` attacks if they already met the ADT requirements to attempt the root goal.

4.3 Analysis of the First Refinement

Behavioral Analysis of Results. As shown in Fig. 4, in the first refinement, the probability of success of the attacker increased to 0.31. By feeding the obtained logs into Disco, we obtain the process in Fig. 9 (left). Here, we can see that we have solved the issue from Sect. 4.2: from state `Start-succ(OpenVault)` we only have the two downward transitions attempting `RobBank` attacks.

From the discovered process, we can see a further issue in the model. We can see that 3 process states have a grey dashed transition towards the bottom-most small grey circle with an inscribed square. These are three *endpoints*, meaning that simulations terminated in such states. The two bottom endpoints related to `Complete` are expected, because Fig. 6 dictates that we do not have outgoing transitions from state `Complete`. Instead, the third endpoint was not expected because `TryBlowUp` was not expected to be a terminal state, and therefore it can be considered as a *deadlock problem*. Notably, as shown in the label of the corresponding dashed transition, this issue has a particularly strong impact, as 138 simulations out of 240 terminate getting stuck in state `TryBlowUp`.

Using Disco, it is easy to *zoom in* specific behavior. Figure 9 (right) shows a process generated by Disco by focusing only on the 138 simulations getting stuck in `TryBlowUp`. We can see that the endpoint state is visited 276 times, twice per simulation. This clearly highlights a conflict among the cost of attempting `BlowUp` attacks (90), and the constraint on maximum allowed cost (100), both shown in Fig. 5 (left). What happens is that the second time the attacker gets in state `TryBlowUp`, they have spent already 90. Looking at Fig. 6, we can see that the two transitions outgoing from this state have action `succ/fail(BlowUp)`, modeling the success or failure of this attack, and target state `Start`. Unfortunately, both actions would lead to a cost of 180, violating the constraint of maximum cost. In other words, both transitions are disabled on second visit of `TryBlowUp`, making it a terminal state.

Model Refinement. This behavior was unintended. We would like the attacker to be able to always get back to state `Start` to attempt further attack strategies without remaining stuck in state `TryBlowUp`.

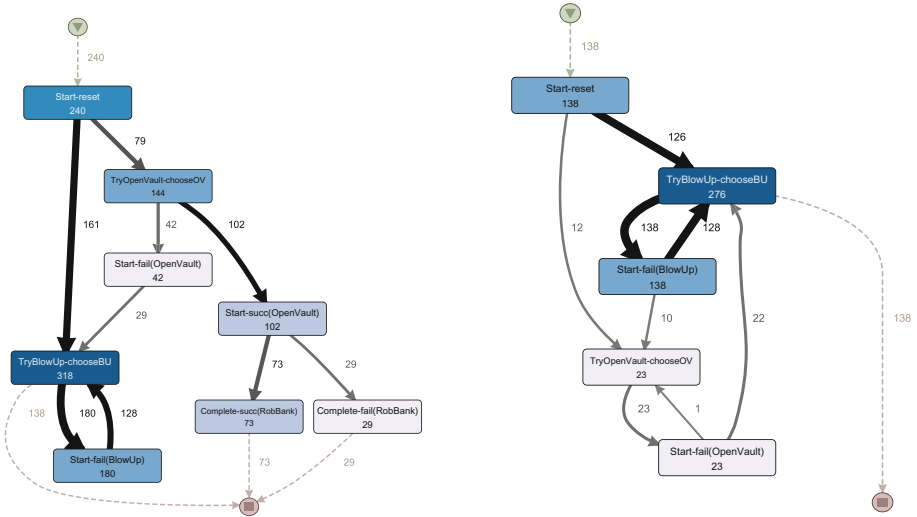


Fig. 9. Behavioral analysis of first refinement. (Left) PM process. (Right) Filtered process for end-point TryBlowUp.

<pre> 1 // Strategy where the attacker tries to blow up the vault 2 Start -(chooseBU, 4, !allowed(RobBank)) -> TryBlowUp, 3 TryBlowUp -(succ(BlowUp), 2) -> Start, 4 TryBlowUp -(fail(BlowUp), 10000) -> Start, 5 TryBlowUp -(goBack, 0.00001) -> Start </pre>	<pre> 1 begin actions 2 chooseOV 3 chooseBU 4 goBack 5 end actions </pre>
--	---

Fig. 10. Second model refinement: we fix the *deadlock* TryBlowUp.

This can be easily solved by adding a third outgoing transition from state TryBlowUp as shown in Fig. 10. Here, with a very low weight, we can take a transition with new action goBack to go back to state Start without attempting any attack. Being the weight so small, this transition will almost always be selected only in the cases that were creating a deadlock in the previous variant. As we shall see in the next section, this guarantees that we find only the two expected complete-related endpoints.

4.4 Analysis of the Second Refinement

Behavioral Analysis of Results. As shown in Fig. 4, in the second refinement, the success probability of the attacker increases considerably from 0.31 to 0.72. This is coherent with the fact that we expect to have improved on the 138 simulations that were getting stuck in state `TryBlowUp`. By feeding the obtained logs into Disco, we obtain the process in Fig. 11. The process generated by Disco confirms that we now have only the two expected endpoints. From the process we see we get in process state `Complete-succ(RobBank)` 173 times, with $173/240 = 0.72$ as indicated in Fig. 4.

Finally, we note that we never enter in a process state involving `succBlowUp`. This is expected from the very high weight (10K) of `succBlowUp`. This was on purpose, to allow for an easier presentation of the results instrumental for presenting the methodology.

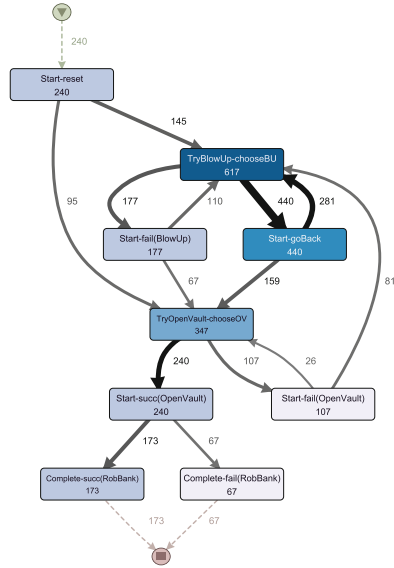


Fig. 11. PM behavior of 2nd refinement

5 Conclusions and Future Work

We proposed a novel methodology for validating and enhancing simulation models to make them more reliable. The methodology is based on the integration of simulation-based analysis techniques known as statistical model checking (SMC), with process-oriented data-driven techniques known and process mining (PM). A simulation in SMC corresponds to a trace in PM. To the best of our knowledge, this is the first integration of SMC and PM. We obtained a novel methodology for *SMC- and PM-guided white-box behavioral model validation and enhancement*.

We demonstrated our methodology on a toy example from the threat modeling domain. We have seen how PM can *discover* issues in the model by inspecting the simulations generated by SMC. Notably, given that SMC is able to decide the *correct* number of simulations necessary to perform an analysis, in some sense we also obtain statistically-reliable event logs for being used with PM.

As future work, we will consider more realistic models, from several domains. E.g., agent-based models from the social sciences, given that the SMC tool MultiVeStA has been recently redesigned and extended to tailor them [4]. Additionally, conformance checking techniques might become relevant in order to ensure that the simulations produced by SMC fulfill normative models. We also foresee

a richer integration of PM and SMC. Currently, we use PM *after* SMC completion. We might consider scenarios where streaming process mining techniques are performed *during* SMC in order to tailor the SMC analysis. In addition, PM might also be used *before* SMC. E.g., discovery algorithms might be applied to real data to synthesize attack-defense trees and/or attacker behaviors.



References

1. van der Aalst, W.M.: Process Mining, 2nd edn. Springer, Cham (2016)
2. Agha, G., Palmiskog, K.: A survey of statistical model checking. *ACM Trans. Model. Comp. Simul.* **28**(1), 6:1–6:39 (2018)
3. Aslanyan, Z., Nielson, F., Parker, D.: Quantitative verification and synthesis of attack-defence scenarios. In: Proceedings of CSF 2016, pp. 105–119. IEEE (2016)
4. ter Beek, M.H., Legay, A., Lafuente, A.L., Vandin, A.: Quantitative security risk modeling and analysis with RisQFLan. *Comput. Secur.* **109**, 102381 (2021)
5. ter Beek, M.H., Legay, A., Lluch-Lafuente, A., Vandin, A.: A framework for quantitative modeling and analysis of highly (re)configurable systems. *IEEE Trans. Softw. Eng.* **46**(3), 321–345 (2020)
6. Belzner, L., De Nicola, R., Vandin, A., Wirsing, M.: Reasoning (on) service component ensembles in rewriting logic. In: Iida, S., Meseguer, J., Ogata, K. (eds.) *Specification, Algebra, and Software*. LNCS, vol. 8373, pp. 188–211. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54624-2_10
7. Bulychev, P.E., et al.: UPPAAL-SMC: statistical model checking for priced timed automata. In: Proceedings of QAPL 2012, vol. 85, pp. 1–16 (2012)
8. Ciancia, V., Latella, D., Massink, M., Paškauskas, R., Vandin, A.: A tool-chain for statistical spatio-temporal model checking of bike sharing systems. In: Margaria, T., Steffen, B. (eds.) *ISoLA 2016*. LNCS, vol. 9952, pp. 657–673. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47166-2_46
9. Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F., Vandin, A.: A formal approach for the analysis of BPMN collaboration models. *JSS* **180**, 111007 (2021)
10. Gilmore, S., Reijnsbergen, D., Vandin, A.: Transient and steady-state statistical analysis for discrete event simulators. In: Polikarpova, N., Schneider, S. (eds.) *IFM 2017*. LNCS, vol. 10510, pp. 145–160. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66845-1_10
11. Gilmore, S., Tribastone, M., Vandin, A.: An analysis pathway for the quantitative evaluation of public transport systems. In: Albert, E., Sekerinski, E. (eds.) *IFM 2014*. LNCS, vol. 8739, pp. 71–86. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10181-1_5
12. Hu, J., Niu, H., Carrasco, J., Lennox, B., Arvin, F.: Fault-tolerant cooperative navigation of networked UAV swarms for forest fire monitoring. *Aerosp. Sci. Technol.* **123**, 107494 (2022)
13. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Foundations of attack-defence trees. In: Degano, P., Etalle, S., Guttman, J. (eds.) *FAST 2010*. LNCS, vol. 6561, pp. 80–95. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19751-2_6
14. Kumar, R., Ruijters, E., Stoelinga, M.: Quantitative attack tree analysis via priced timed automata. In: Sankaranarayanan, S., Vicario, E. (eds.) *FORMATS 2015*. LNCS, vol. 9268, pp. 156–171. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22975-1_11

15. Law, A.M.: *Simulation Modeling and Analysis*, 5th edn. McGraw-Hill, New York (2015)
16. Mauw, S., Oostdijk, M.: Foundations of attack trees. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 186–198. Springer, Heidelberg (2006). https://doi.org/10.1007/11734727_17
17. Pianini, D., Sebastio, S., Vandin, A.: Distributed statistical analysis of complex systems modeled through a chemical metaphor. In: HPCS, pp. 416–423 (2014)
18. Research and Technology Organisation of NATO: Improving Common Security Risk Analysis report. RTO Technical Report TR-IST-049 (2008)
19. Schneier, B.: Attack trees. *Dr. Dobb's J.* **24**, 21–29 (1999). <http://bit.ly/3tcfuoZ>
20. Sebastio, S., Vandin, A.: MultiVeStA: statistical model checking for discrete event simulators. In: 7th International Conference on ValueTools 2013, pp. 310–315. ICST/ACM (2013)
21. Sen, K., Viswanathan, M., Agha, G.: Statistical model checking of black-box probabilistic systems. In: Alur, R., Peled, D.A. (eds.) CAV 2004. LNCS, vol. 3114, pp. 202–215. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27813-9_16
22. U.S. Department of Defense: Defense Acquisition Guidebook, Section 8.5.3.3 (2009). <https://bit.ly/3NJjs07>
23. Vandin, A., ter Beek, M.H., Legay, A., Lluch Lafuente, A.: QFLan: a tool for the quantitative analysis of highly reconfigurable systems. In: Havelund, K., Peleska, J., Roscoe, B., de Vink, E. (eds.) FM 2018. LNCS, vol. 10951, pp. 329–337. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95582-7_19
24. Vandin, A., Giachini, D., Lamperti, F., Chiaromonte, F.: Automated and distributed statistical analysis of economic agent-based models. *J. Econ. Dyn. Control* **143**, 104458 (2022)
25. Younes, H.L.S.: Probabilistic verification for “black-box” systems. In: Etessami, K., Rajamani, S.K. (eds.) CAV 2005. LNCS, vol. 3576, pp. 253–265. Springer, Heidelberg (2005). https://doi.org/10.1007/11513988_25



A Systematic Comparison of Case Management Languages

Julia Holz¹, Luise Pufahl²(✉) , and Ingo Weber² 

¹ Technische Universität Berlin, Berlin, Germany
j.holz@campus.tu-berlin.de

² Technische Universität Berlin, Software and Business Engineering, Berlin, Germany
{luise.pufahl,ingo.weber}@tu-berlin.de

Abstract. Our world and many industries, such as healthcare or consulting, are becoming more digital and focused on knowledge workers. In consequence, flexible, knowledge-intensive business processes are increasingly relevant for organizations, and subject to increased interest. Case management languages support the management of knowledge-intensive processes with respect to documentation, analysis, execution, and monitoring. Besides the standard Case Management Model and Notation (CMMN), other languages have been developed, such as fragment-based Case Management (fCM) and PHILharmonic Flows. So far, no structured comparison of their functionalities and applicability has been undertaken. With this work, we compare CMMN, fCM, and PHILharmonic flow with two structured methods: a functional comparison and a user study. We found that fCM offers broad functionality, but CMMN was perceived to be easier for users.

Keywords: Case management · Comparison · User study

1 Introduction

With an increased automation of structured business processes, knowledge-intensive processes (KIPs) receive growing attention by business organizations [37]. KIPs are driven by knowledge workers who use their expertise and experience to drive a case based on its characteristics. Such processes are often emergent, knowledge and goal-oriented, event-driven, and possibly constraint and rule-driven [6]. Similar to structured business processes, knowledge-intensive processes need to be managed throughout the whole BPM life-cycle [18].

Case management refers to a process management approach that can support the flexible nature of knowledge-intensive processes. It provides concepts, methods, and techniques to manage KIPs' need for variability, adaptation, information, and compliance. Since the first approach to case management was developed by Van der Aalst et al. [34], several alternatives have been designed in research. *Data-oriented* languages, such as PHILharmonic Flows (PHIL) [15], capture mainly the relevant data objects of a case and their life cycle and center the modeling and execution of a case around it. *Constraint-oriented* languages,

such as the DECLARE language [25] or DCR (Dynamic Condition Response) graphs [31], focus on the constraints of a case. The knowledge worker is allowed to do everything as long as the constraints and rules are fulfilled. Finally, *stage-oriented* languages, such as fCM (fragment-based case management) [10] as well as the industry standard CMMN (Case Management Model and Notation) [23] divide a case into different stages that can be flexibly combined at runtime.

This broad range of languages led to the challenge for practitioners to select the “right” language for managing their knowledge-intensive processes. Thus, this paper aims to systematically compare case management languages. For this purpose, a selective set of case management modeling languages are investigated from a functional and understandability perspective. Thereby, we focus on two purposes: modeling and documenting KIPs.

In this work, we selected the industry-standard CMMN [23] and compare it to a representative of the data-oriented language, PHIL [1, 15], and a representative of the stage-oriented language, fCM¹ [8, 10], which both provide a modeling language and an execution engine for cases. Constraint-oriented languages are not considered in this work, because they use the declarative modeling approach. The understandability of declarative vs. imperative modeling was already studied in [25] and it was found that imperative modeling languages are more comprehensible. Declarative languages require a certain familiarity with the constructs to achieve user understandability [25]. However, in this work, we plan a user study with a limited training phase. In contrast to other research on the comparison of case management languages [6, 7, 18, 32], we provide a two-fold comparison with regards to their modeling functionality and understandability:

- Functionality [method: functional comparison]: similar to related work [6, 18, 32], we use a literature analysis to deduce a set of criteria based on which the languages are compared. Additionally, we have modeled three use cases in each language to assess which of the criteria are supported.
- Understandability [method: user study]: we evaluate model understandability of the three languages with a user study, where we test the interpretation effectiveness based on task fulfilment by the users.

The results indicate that the languages have different strengths: the analytical comparison shows that fCM offers the broadest functionality out of the set of languages, while the users in the study found CMMN more understandable. This paper is based on a master’s thesis [12]. In the remainder, related work is presented in Sect. 2. Then, the method for the criteria-based comparison is given and the results presented in Sect. 3, followed by the user study with its design and its results in Sect. 4. Finally, the results are discussed and a summary and outlook is given in Sect. 5.

2 Related Work

In research, different comparisons and assessments of process modeling languages including case management languages are made. Process modeling languages

¹ <https://github.com/bptlab/chimera>, accessed 2022-03-22.

Table 1. Research works comparing case management languages

Reference	Year	Analyzed languages	Method
Pichler et al. [25]	2011	BPMN, ConDec	User Study
Di Ciccio et al. [6]	2015	YAWL, ADEPT2, SmartPM, Declare, PHILharmonic Flows, ArtiFact - GSM, MailOfMine	Criteria-based comparison
Marin et al. [18]	2015	CMMN	Criteria-based assessment
Wiemuth et al. [38]	2017	BPMN, CMMN, DMN	Use case
Zensen and Küster [39]	2018	CMMN, BPMN	Use Case
Steinau et al. [32]	2019	Custom Case Handling Approach, GSM, PHILharmonic Flows	Criteria-based comparison
Gonzalez-Lopez et al. [7]	2021	fCM, fCM Landscape	User Study
Jalali [13]	2021	CMMN and DCR	User Study

have been compared, for example, regarding their (a) capabilities to capture certain characteristics (e.g., execution depends on knowledge) [6], (b) representational capabilities and expressive power with the help of an ontology-based theory of representation (e.g., a stable state) [26] or with the help of patterns (e.g., control flow patterns) [5], (c) understandability and usability (e.g., comprehension task efficiency or perceived usefulness) [20], and practical usage [22].

Table 1 gives an overview of research works which have compared case management languages with each other or with BPMN as well as works that performed a criteria-based evaluation. The table also lists the used method, and is sorted by year to highlight the progress of such kind of research.

Pichler et al. [25] compared an imperative approach in form of BPMN models to a declarative approach in form of ConDec models by investigating the understandability with the help of a user study. This research work focuses on the comparison of the declarative to the imperative modeling paradigm.

Di Ciccio et al. [6] analyzed different case management languages to define knowledge-intensive processes and their characteristics. Based on these characteristics, different case management modeling languages are compared and evaluated. Similarly, Marin et al. [18] surveyed different definitions to find characteristics and requirements of case management. However, they focused only the extent to which CMMN fulfills the characteristics and requirements. While technically not a comparison, we include the paper here due to relevance.

Wiemuth et al. [38] intended to combine business process modeling and adaptive case management in order to model a flexible and variable medical processes. Based on the use case, CMMN and DMN are compared to BPMN. The work is limited to one use case; other case management languages are not considered. Similarly, Zensen and Küster compare modeling a use case with the flexible elements of BPMN to CMMN case model.

Steinau et al. [32] conducted an exhaustive literature survey and analyzed different data-centric process modeling approaches, including a set of case man-

agement languages. Their comparison was based on a set of criteria. As the focus was on data-centric modeling approaches, CMMN was not examined.

Gonzalez et al. [7] extended the case management language fCM for case model landscapes to ease the readability of the case models. They comparatively evaluated the landscape against the original fCM notation with a user study on model understandability. Jalali [13] investigated the perceived usefulness and ease of use of CMMN and DCR in a user study and did not find strong differences between them.

In summary, the most frequently used method for comparison in the literature is checking for criteria coverage, where the criteria were typically derived from the literature. Both existing works with user studies had a different focus, and [25] is more than ten years old and pre-dates the release of CMMN. In this work, we complement the state of the art by (i) conducting a criteria-based comparison which includes CMMN, (ii) modeling different use cases, and (iii) performing a comparative user study. In particular, our comparison includes the CMMN industry standard and two research languages under active development, fCM and PHILharmonic Flows.

3 Criteria-Based Comparison

This section presents the results of the criteria-based comparison. We assume that the readers are familiar with the modeling languages; for details of those, see [8, 15, 23], or for a concise overview see [12, Ch. 2.3]. First, we present our method in Sect. 3.1 and then, we provide the criteria and the results in Sect. 3.2.

3.1 Method

The criteria for the functional comparison were defined based on a literature study with the goal to find relevant papers on criteria-based evaluations of case management languages. We conducted the literature analysis using the knowledge databases and search engines Primo (the main knowledge database of the Technical University Berlin) and Google Scholar. The search terms were ((“Case Management” OR “Case Modeling” OR “Case Handling”) AND (“process modeling” OR “comparison” OR “analysis” OR “assessment” OR “evaluation”)). We received 18.100 results. The retrieved papers were then cleansed of medical results, as case management is still strongly linked to the healthcare domain, and reduced with regard to duplicates. Papers were included that focus on case management and requirements, a comparison, an assessment, or an evaluation. From this analysis, we identified the 14 papers referenced in Table 2 that define relevant criteria for case management.

Furthermore, requirements for the selection of criteria were defined: A criterion must be (1) universally valid, attainable, based on the characteristics of case management, (2) not redundant, and (3) relate to the case design-phase. Overall, we obtained 96 criteria, from which we removed 55 duplicates and 22 criteria relating to the case execution phase, for example the criteria “Unanticipated

exceptions” and “Flexible execution” [6]. Furthermore, we removed seven criteria for being not attainable, too general or relating to a high-level requirement, such as “Advanced collaboration” [9] or “Implicit process description” [28].

In the criteria assessment, we followed a two step approach: we first modeled use cases in each language and assessed then the fulfillment of the criteria. By modeling the use cases, a more detailed understanding of the modeling languages, their characteristics, and features is created. For criteria which were only partial or not fulfilled, we re-read the description of the modeling language to rule out biases and possible limitations from the use cases.

Three use cases were selected and textually described [12, Appendix A-C]. The use cases originate from different case management domains: medical, administrative, and consulting. The medical and the administrative process were both derived from public event logs [17,36]. Specifically, we derived textual process description by analysing the most common variants and directly-follows graphs in the process mining tool Disco. The third use case, a consultancy project planning process, was elicited by six interviews in a software engineering and consulting firm, from which we created a detailed textual process description. With the textual descriptions, we modeled the three use cases in each of the three case management languages [12, Ch. 6.1] using MS Visio. MS Visio provides the flexibility to use all notational elements present in a modeling language.

Eventually, using the insights from the modeling, the support for each criterion selected before was assessed for CMMN, fCM, and PHILharmonic Flows (PHIL) on a three-value-scale: full support, partial/implicit support, and no support. If we observed partial or no support, we checked again the description of the modeling language to validate whether a criterion was really partially or not supported.

3.2 Criteria and Analysis

Based on the literature analysis, we derived 12 relevant criteria for comparing the case management languages. The criteria, a brief description, related references, and the fulfillment of each language are shown in Table 2. The first six criteria are related to the characteristics of case management. Following are six criteria related to modeling capabilities. We discuss the criteria and results next.

Case Management Criteria. *Knowledge-driven* describes the influence of data and its availability as well as decisions made by knowledge workers on the progression of the case [6,18]. Through the case progression, the process-related knowledge evolves [19]. All three examined case management languages fully support this criterion. In CMMN, the executing knowledge worker can influence the execution of the process by making decisions based on skills and expertise. An fCM model is driven by the availability of case data expressed by conditional start events of the process fragments. Enabled fragments are executed at the discretion of the executing worker. Knowledge-driven aspects of PHIL are expressed in the micro processes, one for each case data object. Those

Table 2. Selected criteria, references and fulfillment by the three case management languages. Scale: support exceeded expected levels ‘✓✓’, full support ‘✓’, partial/implicit support ‘(✓)’, and no support ‘×’.

Name	Description	References	CMMN	fCM	PHIL
Knowledge-driven	Execution depends on the availability of data and decisions by experts	[6, 11, 18, 21]	✓	✓	✓
Data modeling	Support for the modeling of data	[4, 6, 18, 28, 32, 38]	(✓)	✓	✓
Goal modeling	Support for the modeling of goals	[4, 6, 18]	(✓)	✓	(✓)
Data-driven activities	Support for the modeling of data-driven activities	[6, 18, 28, 34]	✓	✓	✓
External events	Support for the modeling of external events	[4, 6, 7, 18]	✓	✓	×
Resource or skill modeling	Support for the modeling of resources and skills of knowledge workers	[6, 18]	(✓)	×	×
Different modeling styles	Support for different modeling styles within the modeling environment	[6, 18, 32]	(✓)	✓	✓
Management of rules and constraints	Support for the formalization and management of rules and constraints	[6, 9, 18, 25, 31, 33]	✓✓	✓	(✓)
Management of roles	Support for the modeling of roles of knowledge workers	[9, 11, 31, 38]	✓	×	✓
Management of process granularity	Support for the management of process granularity and enforcement of granularity levels	[7, 32]	(✓)	✓	✓
Specification of case data behavior	Support for the modeling the behavior of case data	[32]	×	✓	✓
Specification of interactions	Support for the modeling of interactions between processes	[4, 7, 28, 32]	(✓)	(✓✓)	×
Sum	Exceeded/full/partial/no support		1/4/6/1	0/9/1/2	0/7/2/3

data objects and its properties may then influence the execution of the macro model synchronizing the case data.

Data modeling is supported when a case management language supports the specification of a data model or its elements [18]. Explicit support for data modeling is provided when data properties and relations between data types are included [21]. CMMN allows the modeling of data in form of case file items. Nevertheless, CMMN is classified as partial support, because relations between different case data types and their properties cannot be expressed. The modeling languages fCM and PHIL both feature individual data models. Hence, both are rated as full support.

Goal modeling is supported by a case management language, if the language allows the (explicit) definition of a process goal [4]. The process goal is a global goal, and usually represents the possible termination of a case. It may be data or decision-based [6]. CMMN is rated as supporting it partially, because a separate goal definition regarding the case is not part of a CMMN model. Nonetheless, milestones and the exit criterion can be perceived as an implicit definition of a case goal. In fCM, a goal state is specified explicitly. PHIL does not require an explicit definition of a goal state, but by highlighting the final stage, goal modeling is supported implicitly.

Data-driven activities. Activities in knowledge-intensive processes depend on the related data [34]. Hence, *data-driven activities* can be represented in terms

of data conditions, but also by the influence of a separately defined data model. Data may influence the ordering, start, and end of activities [18]. CMMN, fCM and PHIL fully support data-driven activities.

External events. If external events are supported, the case modeling language allows external triggers to influence the process progression. Such a trigger originates from the process's environment and may alter data states and the sequence flow [6]. External events are included as predefined elements in CMMN. The process fragments in fCM contain external events as well. Accordingly, external events are supported completely by both languages. However, PHIL provides no support for the specification of external events.

Resources and skills Resources and skills of process-related knowledge workers are critical for case management processes [18]. The criterion examines whether it is possible to represent resources and their skills by a notational element. CMMN supports a basic or implicit support for resource and skill modeling. In fCM and PHIL, resources and skills cannot be modeled.

Criteria Regarding Modeling Capabilities. In this part, we describe the criteria related to modeling capabilities of a case management language, such as different modeling styles, the management of rules and constraints, roles, and process granularity, as well as the specification of case data behavior and the interactions.

A case consists of several elements that partially represent knowledge, e.g., data objects, separately defined skills, or behavior. Those elements might have different *degrees of structuredness* [6]. To represent the aforementioned elements appropriately, different *modeling styles* can be required, for instance, the declarative or imperative modeling style. CMMN has a strong declarative flavor [30] in defining the relations between the stages, but it also allows defining imperative parts by having a process task that links to a BPMN diagram [18]. fCM and PHIL combine both styles in their modeling notation.

In case management, *rules and constraints* are integral elements to structure a case. Thus, according to [9], case management languages are supposed to support the explicit definition of rules and constraints by the process modeler. fCM supports rules and constraints via the definition of data constraints and the usage of BPMN events (e.g., for the definition of timer constraints) in the fragments and thereby supports this criteria fully. CMMN is even more flexible, thus we recorded the support to exceed expected levels: sentries of tasks and stages allow the definition of data constraints, or rules in any rule language by defining an expression and referring the language [23] PHIL provides also the definition of data constraints. It provides a partial support because certain constraints, e.g. timer constraints, cannot be expressed.

Case management requires a *definition of roles*. The role definition has no predetermined level of precision, it can range from complex role definitions to simple roles, using e.g., only skip permissions [9]. The definition and management of roles is provided by CMMN, thus offering full support. It restricts which role is allowed to perform tasks and modify the case plan model at runtime.

However, roles have no notational element, they are only specify as attributes. fCM provides neither a role nor a permission management. Roles in PHIL can be managed and are specified as permissions in the data model. A role can generally grant reading or writing rights, marking partial support.

The degree of detail in a case management process model is described by *process granularity* [7]. Supporting this criterion are case management languages that enforce or at least recommend particular levels of granularity [32]. CMMN provides partial support for the management of process granularity by allowing the clustering of case plan items into stages. The level of granularity and thus the level of detail in fCM and PHIL is managed through the different models. In fCM, the domain model, the object lifecycle model, and the process fragments each display a different level of granularity of a case. The same applies to all components of a PHILharmonic Flows model. Hence both fCM and PHILharmonic Flows are rated to fully support the process granularity.

Specification of case data behavior means that the case management language provides support to specify the allowed behavior at runtime of the data involved in a case [32]. CMMN does not support the modeling of data object behavior. The object lifecycle model of a fCM model provides a full support and fundamentally represents a behavior model and explicitly shows how a data object behaves during process execution. PHIL depicts the behavior within its micro processes, hence the rating of full support.

Finally, the last criterion concerns whether a modeling language allows the modeling of *interactions between processes*. It requires the inclusion and visibility of the connection in the model [35]. It is irrelevant for the evaluation of the criterion whether the interacting processes are modeled in the same modeling language. Interactions between processes can be modeled in CMMN using tasks that link BPMN or other CMMN models. However, a possible data exchange between processes and the precise connections cannot be modeled. For this reason, CMMN was ranked with partial support. Also rated with partial support is fCM where interactions between processes can be depicted with message events of the BPMN language, the modeling language used for the process fragments. PHIL does not support interactions between processes.

Summary of Observations. Overall, it can be observed that fCM has the highest number of criteria fully support followed by PHIL. fCM provides language concepts for the modeling of different case management characteristics, such as data, goals, and external events. CMMN rather indirectly supports certain aspects like data, goals and resources. PHILharmonic Flows provides like fCM no modeling concept for resources, and additionally, external events cannot be captured. Less of the modeling capabilities are supported by the languages. Whereas CMMN has its strength in the management of constraints and roles, fCM and PHIL support different modeling styles, the management of process granularity and the explicit definition of case data behavior. In contrast to fCM, PHIL has the capability of role management.

4 User Study

To evaluate the model understandability of the three chosen modeling languages, we conducted an experiment in a user study. For this purpose, the users' interpretation effectiveness of different case models is compared to identify the level of model understandability. The study subjects were asked to answer questions about the case models. The main measure gathered from the experiment was the interpretation effectiveness, representing the number of correct answers [3, 16].

4.1 Hypothesis and Experiment Design

In this experiment, we follow the guidelines for empirical evaluations of modeling languages, proposed by Burton-Jones et al. [3]. When planning the study, we also considered the guidelines for experimental design by Juristo and Moreno [14].

To compare the case management languages used for this experiment, we defined response variables. The dependent variables are *effectiveness* and *perceived difficulty*. The effectiveness is measured by the number of correct answers and may range between 0 and 15, as 15 questions per process model are to be answered. The perceived difficulty is rated by the participants directly and may range from 1 (very easy) to 5 (very hard). The independent variable is the modeling language. It is predefined and not modifiable by study subjects. As mentioned, CMMN is an industry standard and has undergone an exhaustive development process. Thus, we hypothesized that, comparison to the other two languages, CMMN performs higher in terms of measured effectiveness, and lower in perceived difficulty of CMMN. As such, we defined the following hypotheses and corresponding null hypotheses by using the dependent variables effectiveness and perceived difficulty:

- $H1_A$: The measured effectiveness of CMMN is higher than the measured effectiveness of fCM and PHIL.
- $H1_0$: There is no significant difference in the measured effectiveness of CMMN, fCM and PHIL.
- $H2_A$: The perceived difficulty of the CMMN is lower than the perceived difficulty of fCM and PHIL.
- $H2_0$: There is no significant difference in the perceived difficulty of CMMN, fCM and PHIL.

The experiment follows a crossover design [7]. Each subject receives three case models, each representing one of the previously modeled use cases [12, Ch. 6.1] (i.e., sepsis treatment, purchase handling and consultancy project planning) and one of the modeling languages (CMMN, fCM and PHILharmonic Flows (PHIL)). We aimed to mitigate possible object learning effects by presenting each process only exactly once to each subject, and technique learning effects by using each modeling language only exactly once per questionnaire. Furthermore, we changed the sequence in which the modeling languages are presented to the participants to minimize effects of tiredness. This results in the six possible combinations illustrated in Table 3. All case models were checked to be of

Table 3. Combinations

#	Sepsis process	Purchase handling process	Procurement process
1	CMMN	fCM	PHIL
2	CMMN	PHIL	fCM
3	fCM	CMMN	PHIL
4	fCM	PHIL	CMMN
5	PHIL	fCM	CMMN
6	PHIL	CMMN	fCM

comparable complexity by aligning the number of activities, events, and particularities, if necessary. Semantic equivalence was ensured by checking in the group of co-authors that each understanding question results in the same answer for all three languages. Few exceptions were in aspects that can not be expressed by the language, e.g. relation between case data in CMMN, roles in fCM, timer constrains in PHIL. In those cases, the indented right answer for the question is “I don’t know”. During the experiment, the models were constantly available, as proposed by Parson and Cole [24].

4.2 Experiment Implementation

The subjects were BSc and MSc students, PhD candidates and professionals who were invited to voluntarily participate in the anonymous experiment. The students were from the University of Potsdam and the Technische Universitaet Berlin. The students participating can be considered future users of business process management, including case management. The entirety of subjects in this experiment represents the target audience of case management as they have a basic knowledge of business process modeling and/or work in technical fields where process modeling is applied.

The questions, the provided material, and the models were solely available in English. Each subject was randomly assigned one of the combinations from Table 3, but with consideration of an equal distribution among the combinations.

The experiment was conducted online using Google Forms. The questionnaire was available for 2.5 weeks in May 2021. Due to COVID-19 restrictions, we were not able to conduct the experiment under laboratory conditions. All invited subjects received a link to the Google Forms questionnaire. The answers were automatically logged and checked for correctness by the platform. Only fully completed questionnaires were considered. We received 26 complete responses, distributed over the six combinations. Participants reported that they needed between 30 and 60 min for partaking in the study. The questionnaire and material was structured into four parts (1) demographic questions;(2) process modeling questions used to assess the participant’s experience in business process modeling and case management; (3) a brief overview and introduction to the used case management modeling languages in the form of specially produced videos;

Table 4. Number of User Types per combination with User Type A (no or basic knowledge in process modeling), User Type B (advanced knowledge in business process modeling) and User Type C (professional knowledge in business process modeling or prior knowledge in Case Management approach)

#	User type A	User type B	User type C	Σ
1	1	1	2	4
2	1	1	1	4
3	1	2	2	5
4	2	1	1	4
5	3	1	1	5
6	1	3	0	4
Σ	9	10	7	26

(4) understandability test consisting of 15 questions for each presented process model, and user feedback in form of rating the perceived difficulty and an open feedback question.

The questions in the fourth part were arranged in random order, and participants selected an answer out of the options “true”/“false”/“I don’t know”. The “I don’t know” option enabled participants to mark that an answer could not be given on the basis of the provided model, or to avoid guessing in case of uncertainty. The questions were formulated consistently with regard to case-related characteristics to ensure comparable difficulty. The effectiveness was then measured as the total score for each model separately. Out of the three options, exactly one was correct in each case, i.e., in some cases the “I don’t know” option was rated as correct if an aspect was not present in the corresponding model. A correct answer translates to one point, which means 15 points could be achieved per model. For each of the three models, participants rated the perceived overall difficulty and had an option to provide free-text feedback on the case model. The full questionnaire including all questions can be found in [12, pp. 73].

4.3 Experiment Results

After the experiment was concluded, the validity of the data was analyzed. The analysis of plausibility and consistency revealed that all responses are valid and were then used in the subsequent statistical analysis. All results presented in this section, and the related summaries and conclusions, relate solely to this experiment and do not claim general validity.

26 subjects participated with 15 students and 11 postgraduates. In Table 4, the number of participants per combination is shown, categorized by their knowledge on process modeling. In summary, we achieved a good distribution of participants with different knowledge regarding process modeling over the different questionnaire combinations. Only two subjects stated to have no knowledge in business process modeling, all other participants had at least basic knowledge. Also, only one of the participants had already worked with case management, all other users of type C had professional knowledge in BPMN. The data shows

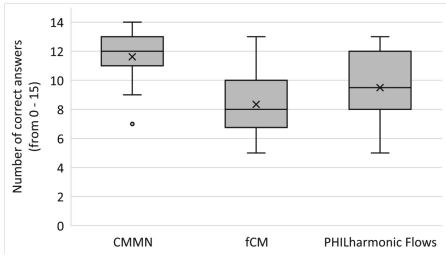


Fig. 1. Effectiveness (average marked with “X” and median with “-”).

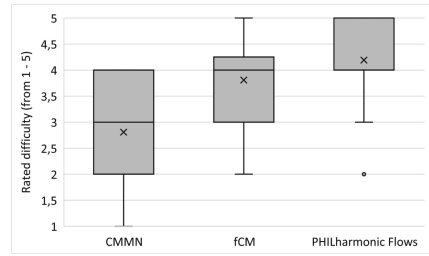


Fig. 2. Perceived Difficulty (average marked with “X” and median with “-”).

Table 5. Hypothesis testing results

Sub-hypothesis	Sig. (p)	Status	Hypothesis	Status
$H1_0^{CMMN-fCM}$	0.00	rejected	$H1_0$	rejected
$H1_0^{CMMN-PHIL}$	0.00	rejected		
$H1_0^{fCM-PHIL}$	0.03	rejected		
$H2_0^{CMMN-fCM}$	0.00	rejected	$H2_0$	rejected
$H2_0^{CMMN-PHIL}$	0.00	rejected		
$H2_0^{fCM-PHIL}$	0.08	not rej		

that, among our participants, BPMN is by far the most used and known process modeling language of the four considered, while CMMN is ahead of fCM and PHIL.

Figure 1 and 2 provide the resulting effectiveness and perceived difficulty for each of the case management modeling languages. The effectiveness is depicted in form of a score that can range from 0 to 15. It represents the number of correctly answered questions. The average score for CMMN is 11.62 with a median of 12. For fCM, the average score was 8.35 and the median 8. The average and median scores of PHIL are identical and equal 9.5². According to the analysis of the measured effectiveness in this experiment, CMMN is the most understandable modeling language, followed by PHIL and lastly fCM. However, PHIL has a higher scattering of the effectiveness data. Figure 2 implies that PHIL is perceived as the most difficult case modeling language, while CMMN appears to be the easiest of the modeling languages. Nevertheless, the median of the perceived difficulty of fCM and PHIL is equal at 4.

Following the analysis, the hypotheses introduced above were tested. In this experiment, the statistical analysis considered a 95% confidence. The data used for the hypothesis testing consists of paired measurements. For testing, the hypotheses were subdivided for pairwise comparison of languages. For instance, $H1_0^{CMMN-fCM}$ is the sub-hypothesis that states that no difference exists between

² Median is not an integer, which happened since the number of data points is even.

CMMN and fCM in terms of effectiveness. The paired measurements are the scores or perceived difficulty ratings for one process modeled in two modeling languages. The subjects are equivalent to the participants, which provides the independence of subjects. As the measured differences of the data are non-normally distributed, a non-parametric alternative to the t-test is applied: the Wilcoxon signed rank test.

Table 5 showing the results of the pair-wise testing demonstrates that the hypothesis testing provided significant evidence for differences in effectiveness between CMMN, fCM and PHIL. From analyzing the data and intermediate results from the Wilcoxon signed rank test, we derived that CMMN has a higher understandability than fCM ($T = (-)0$) and PHIL ($T = (-)13$). The analysis further indicates that PHIL has a higher effectiveness and therefore a better understandability than fCM ($T = (+)67,5$). From the results for the sub-hypotheses, we can deduce that H_{10} can be rejected as well. Hypothesis testing provides significant statistical evidence to reject the assumption that CMMN is perceived as difficult as fCM and PHIL. Conversely, by further analysis we found that CMMN is perceived as easier than fCM ($T = (+)20$) and PHIL ($T = (+)5,5$) by the participants. Finally, hypothesis $H_{20}^{\text{fCM-PHIL}}$ cannot be rejected by hypothesis testing, indicating that no significant difference in the perceived difficulty of the two exists. However, this does not factor into H_{20} . Given the results for sub-hypotheses $H_{20}^{\text{CMMN-fCM}}$ and $H_{20}^{\text{CMMN-PHIL}}$, we also reject H_{20} .

From the comments given for the different models, we were able to gather feedback for a brief qualitative analysis. The comments on CMMN state that the modeling language is “more readable in comparison” and questions were considered “easier to answer”. Nevertheless, one participant commented that the model was not very informative due to only one type of connector, a lack of phasing, and a general lack of time dependencies. Secondly, the comments on fCM focus on the complexity and associated difficulty of readability. Also, obtaining an overview of the case model is considered difficult. However, it was noted that the process fragments were easy to understand due to the BPMN notation. Finally, PHIL was repeatedly called the hardest of the three in the comments. To one participant the connection of the different micro processes and macro process of PHIL stayed unclear.

5 Discussion and Outlook

Main Observations. Based on the two-fold comparison, the main observations for the different case modeling languages are the following. *fCM* performs strongest out of the three languages in the analytical comparison with the highest full coverage of the selected case management and modeling capability criteria. However, it has the lowest interpretation effectiveness in the comparison and performs similar to PHILharmonic Flows (PHIL) regarding the perceived interpretation difficulty. Therefore, Gonzalez et al. [7] have developed a case landscape for fCM to improve the model interpretation effectiveness of users. In contrast, *CMMN* supports less of the functional criteria fully, but has both a

higher interpretation effectiveness and a lower perceived difficulty. Based on the insight from literature [2], this could be explained with less modeling concepts of CMMN in contrast to fCM and PHIL, which could improve the interpretation effectiveness of people. The data-oriented case management approach *PHIL* supports fewer of the case management criteria, but supports more criteria as fCM regarding the modeling capabilities. Subjects perceived *PHIL* as more difficult, but the interpretation effectiveness of the language was higher than fCM. A possible reason could be that *PHIL* focuses on the case data and their life cycles, whereas fCM uses multiple model types representing the process fragments on the one hand, and the involved data on the other hand.

Threats to Validity. In the analytical comparison, criteria were selected based on a structured literature search. Still, they have not been checked for completeness and practical relevance, which could be done in future. When assessing the criteria, we took certain measures to reduce the bias. First, case models based on real-world scenarios were modelled. When certain criteria were not or only partially supported, we checked again the respective description of the modeling language to rule out limitations from the use cases.

Regarding the experiments: domain knowledge of participants could influence the results which should be investigated in future works. Alternatively, the possibility of domain knowledge could be eliminated by providing business process models labeled with abstract symbols, like letters, instead of task descriptions. Furthermore, comparable difficulty of statements could be ensured more precisely by not only using guidelines for the formulation but by pre-tests. Comparable process model complexity could have been determined by different methods than the model metrics used here.

Concerning the implementation, it was not possible due to COVID-19 restrictions to conduct the experiment under laboratory conditions. In a laboratory experiment, the proposed variables' interpretation effort and efficiency could be included, as a meaningful measurement of time would be possible [3]. The number of participants is sufficient to perform significance analyses. Still, the significance of the results would likely be higher with more subjects participating in the experiment. The influence of prior knowledge of business process modeling, familiarity with case management, and the individual modeling languages on the results should be examined and evaluated in further analysis through statistical analyses. Overall, all results from this study should be validated in further experiments, ideally taking the described threats to validity into consideration.

Summary and Outlook. In this work, we conducted a structured comparison of the case management languages CMMN, fragment-based Case Management (fCM), and PHILharmonic Flows (PHIL) with regards to their modeling functionality and applicability. This comparison extends existing research in that it not only analyzes functional aspects but also the understandability. In the context of our study, the results show that CMMN provides better comprehensible

case models in contrast to the other two languages, but it provides less functional support. Also, there is no broad usage in case engines yet [29]. It might be useful as an intermediate for business users and could be translated into other case management languages, such as fCM or PHIL; to generate models which could then be used to verify or execute the cases. The method we established in this research work could also be used as a framework to compare the modeling capability of case management languages. Currently, it focuses on CMMN, fCM, and PHIL. In future, it could be used to evaluate further existing languages, such as DCR graphs. The modeling capabilities have been evaluated regarding their coverage of requirements of knowledge-intensive processes, taken from the literature. Still, in the future pattern-based or ontological-based comparison regarding the representational capabilities [27] could be conducted. Furthermore, the focus of this comparison is on modeling and can be extended to execution, monitoring and analysis capabilities of case management languages in the future.

References

1. Andrews, K., Steinau, S., Reichert, M.: Enabling runtime flexibility in data-centric and data-driven process execution engines (2021). IS 101, 101447
2. Bajaj, A.: The effect of the number of concepts on the readability of schemas: an empirical study with data models. *Requirements Eng.* **9**(4), 261–270 (2004)
3. Burton-Jones, A., Wand, Y., Weber, R.: Guidelines for empirical evaluations of conceptual modeling grammars. *J AIS* **10**(6), 495–532 (2009)
4. de Man, H.: Case management: a review of modeling approaches. *BPTrends*, January (2009)
5. van Der Aalst, W.M., Ter Hofstede, A.H., Kiepuszewski, B., Barros, A.P.: Workflow patterns. *Distrib. Parallel Databases* **14**(1), 5–51 (2003)
6. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. *JoDS* **4**(1), 29–57 (2015). <https://doi.org/10.1007/s13740-014-0038-4>
7. Gonzalez-Lopez, F., Pufahl, L., Munoz-Gama, J., Herskovic, V., Sepúlveda, M.: Case model landscapes: toward an improved representation of knowledge-intensive processes using the fCM-language. *SoSym* **20**, 1–25 (2021)
8. Haarmann, S., Holfter, A., Pufahl, L., Weske, M.: Formal framework for checking compliance of data-driven case management. *JoDS* **10**, 1–21 (2021)
9. Hauder, M., Pigat, S., Matthes, F.: Research challenges in adaptive case management: a literature review. In: 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW) (2014)
10. Hewelt, M., Weske, M.: A hybrid approach for flexible case modeling and execution. In: La Rosa, M., Loos, P., Pastor, O. (eds.) *BPM 2016*. LNBIP, vol. 260, pp. 38–54. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45468-9_3
11. Hinkelmann, K., Pierfranceschi, A.: Combining process modelling and case modelling. In: *MeTTeG14*, pp. 83–93. Universitas Studiorum, Mantova (2014)
12. Holz, J.: A Systematic Comparison of Case Management Approaches. Master's Thesis, TU Berlin (2021). <https://doi.org/10.13140/RG.2.2.22243.68649>
13. Jalali, A.: Evaluating perceived usefulness and ease of use of CMMN and DCR. In: Augusto, A., Gill, A., Nurcan, S., Reinhartz-Berger, I., Schmidt, R., Zdravkovic, J. (eds.) *BPMDS/EMMSAD -2021*. LNBIP, vol. 421, pp. 147–162. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79186-5_10

14. Juristo, N., Moreno, A.M.: *Basics of Software Engineering Experimentation*. Springer, Boston (2001)
15. Künzle, V., Reichert, M.: PHILharmonicFlows: towards a framework for object-aware process management. *J. Softw. Evol. Process.* **23**(4), 205–244 (2011)
16. Malinova, M.: *A Language for Designing Process Maps*. Ph.D. thesis, WU Vienna (2016)
17. Mannhardt, F.: Sepsis cases - event log. <https://doi.org/10.4121/UUID:915D2BFB-7E84-49AD-A286-DC35F063A460>
18. Marin, M.A., Hauder, M., Matthes, F.: Case management: an evaluation of existing approaches for knowledge-intensive processes. In: Reichert, M., Reijers, H.A. (eds.) *BPM 2015. LNBP*, vol. 256, pp. 5–16. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42887-1_1
19. Marjanovic, O., Freeze, R.: Knowledge intensive business processes: theoretical foundations and research challenges. In: 44th HICSS. IEEE, Piscataway (2011)
20. Mendling, J., Strembeck, M., Recker, J.: Factors of process model comprehension-findings from a series of experiments. *DSS* **53**(1), 195–206 (2012)
21. Motahari-Nezhad, H.R., Swenson, K.D.: Adaptive case management: overview and research challenges. In: 2013 IEEE 15th Conference on Business Informatics (2013)
22. Muehlen, Michael zur, Recker, Jan: How much language is enough? Theoretical and practical use of the business process modeling notation. In: Bubenko, J., Krogstie, J., Pastor, O., Pernici, B., Rolland, C., Solvberg, A. (eds.) *Seminal Contributions to Information Systems Engineering*, pp. 429–443. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36926-1_35
23. Object management group: case management model and notation (CMMN): specification. version 1.1 (2016)
24. Parsons, J., Cole, L.: What do the pictures mean? Guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques. *Data Knowl. Eng.* **55**(3), 327–342 (2005)
25. Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., Reijers, H.A.: Imperative versus declarative process modeling languages: an empirical investigation. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *BPM 2011. LNBP*, vol. 99, pp. 383–394. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_37
26. Recker, J., Rosemann, M., Indulska, M., Green, P.: Business process modeling—a comparative analysis. *J AIS* **10**(4), 1 (2009)
27. Recker, J., Rosemann, M., Krogstie, J.: Ontology-versus pattern-based evaluation of process modeling languages: a comparison. *Commun. Assoc. Inf. Syst.* **20**(1), 48 (2007)
28. Reijers, H.A., Rigter, J.H.M., van der Aalst, W.M.P.: The case handling case. *Int. J. Coop. Inf. Syst.* **12**(03), 365–391 (2003)
29. Routis, I., Bardaki, C., Dede, G., Nikolaidou, M., Kamalakis, T., Anagnostopoulos, D.: CMMN evaluation: the modelers’ perceptions of the main notation elements. *SoSym* **20**, 1–21 (2021). <https://doi.org/10.1007/s10270-021-00880-3>
30. Slaats, T.: Declarative and hybrid process discovery: recent advances and open challenges. *JoDS* **9**(1), 3–20 (2020). <https://doi.org/10.1007/s13740-020-00112-9>
31. Slaats, T., Mukkamala, R.R., Hildebrandt, T., Marquard, M.: Exformatics declarative case management workflows as DCR graphs. In: Daniel, F., Wang, J., Weber, B. (eds.) *BPM 2013. LNCS*, vol. 8094, pp. 339–354. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_28

32. 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. *SoSym* **18**(4), 2679–2716 (2019). <https://doi.org/10.1007/s10270-018-0695-0>
33. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative workflows: balancing between flexibility and support. *CSRD* **23**(2), 99–113 (2009). <https://doi.org/10.1007/s00450-009-0057-9>
34. van der Aalst, W.M., Weske, M., Grünbauer, D.: Case handling: a new paradigm for business process support. *Data Knowl. Eng.* **53**(2), 129–162 (2005)
35. van der Aalst, W.M., Weske, M., Wirtz, G.: Advanced topics in workflow management: issues, requirements, and solutions. *JIDPS* **7**(3), 49–77 (2003)
36. van Dongen, B.F.: BPI challenge (2019). <https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1>
37. White, M.: Case mgmt.: combining knowledge with process. *BPTrends* (2009)
38. Wiemuth, M., Junger, D., Leitritz, M.A., Neumann, J., Neumuth, T., Burgert, O.: Application fields for the new object management group (OMG) standards case management model and notation (CMMN) and decision management notation (DMN) in the perioperative field. *IJCARS* **12**(8), 1439–1449 (2017). <https://doi.org/10.1007/s11548-017-1608-3>
39. Zensen, A., Küster, J.: A comparison of flexible BPMN and CMMN in practice: a case study on component release processes. In: *EDOC*, pp. 105–114. IEEE (2018)



Declarative Guideline Conformance Checking of Clinical Treatments: A Case Study

Joscha Grüger^{1,2(✉)}, Tobias Geyer^{2(✉)}, Martin Kuhn²,
Stephan A. Braun^{3,4(✉)}, and Ralph Bergmann^{1,2(✉)}

¹ Business Information Systems II, University of Trier, Trier, Germany

² German Research Center for Artificial Intelligence (DFKI), SDS Branch Trier,
Trier, Germany

{grueger,bergmann}@uni-trier.de, {tobias.geyer,martin.kuhn}@dfki.de

³ Department of Dermatology, University Hospital Münster, Münster, Germany

⁴ Department of Dermatology, Medical Faculty, Heinrich Heine University,
Düsseldorf, Germany

Stephanalexander.Braun@ukmuenster.de

Abstract. Conformance checking is a process mining technique that allows verifying the conformance of process instances to a given model. Thus, this technique is predestined to be used in the medical context for the comparison of treatment cases with clinical guidelines. However, medical processes are highly variable, highly dynamic, and complex. This makes the use of imperative conformance checking approaches in the medical domain difficult. Studies show that declarative approaches can better address these characteristics. However, none of the approaches has yet gained practical acceptance. Another challenge are alignments, which usually do not add any value from a medical point of view. For this reason, we investigate in a case study the usability of the HL7 standard Arden Syntax for declarative, rule-based conformance checking and the use of manually modeled alignments. Using the approach, it was possible to check the conformance of treatment cases and create medically meaningful alignments for large parts of a medical guideline.

Keywords: Declarative modeling · Arden syntax · Process mining · Conformance checking · Alignments · Guideline compliance

1 Introduction

Clinical guidelines are systematically developed statements that reflect the state of evidence-based medical knowledge at the time the guidelines are created. They are intended to support physicians and patients in the decision-making process for appropriate medical care in specific clinical situations [1]. Thus, they represent the documentation of evidence-based medicine and are an important tool for the scientifically sound treatment of patients. However, it is not yet possible to say to what extent guideline knowledge is applied in treatment [2, 3]. Although some approaches were presented, no holistic standard for the representation and

verification of guideline compliance in clinical treatment has yet become established. The investigation of guideline compliance or non-compliance is of medical interest though and holds potential insights for research. For this purpose, the process perspective can provide essential information, e.g., to research observed behavior such as differences between the executed activities and the activities recommended in the guidelines [4] and to help to understand the corresponding reasons and implications, which is in the interest of physicians [5].

The detection of deviations and the evaluation of the conformity of process instances is part of process mining. Process mining is an emerging field of research and fills the gap between data mining and business process management [6]. One technique of process mining is conformance checking, whose approaches focus on measuring the conformance of a process instance to a process model. The results of the measurement can usually be output in the form of metrics or alignments [7], i.e., corrective adjustments for process instances.

Previous research attempted to convert guideline knowledge into imperative representations of a data Petri net and use this to check conformance [8,9]. The result is that the high degree of variability and flexibility of medical processes makes the complete imperative modeling of all variants almost impossible [9].

One way to address the stated difficulties is to use declarative process models. Declarative process modeling languages are considered to be particularly suitable for representing processes in the medical context [10,11]. Unlike imperative approaches, declarative modeling allows the definition of rules for the execution of processes without excluding non-modeled process behavior [12]. This reflects well the nature of medical processes and guidelines and takes into account the unpredictability of events over the course of treatment.

Both the imperative and declarative conformance checking approaches provide alignments that are technically appropriate. However, they are not meaningful from a medical perspective. From the perspective of domain experts, these alignments include adjustment instructions that are inappropriate for the use and evaluation of the results. For example, it does not make sense to delete patients' diseases or change the guideline as part of the alignment simply because this can optimize the costs of the alignment [9].

In this case study, we investigate to which extent the mentioned problems of imperative process models trying to cover all contingencies and, more generally, the medical meaningfulness of alignments can be addressed in a declarative approach. For this purpose, we use the Arden Syntax for Medical Logic Modules (MLMs), which is widely used in medicine and HL7 standardized, to formalize medical knowledge [13]. Besides the classical use of Arden syntax, we show how it can be used as a declarative description of process models by defining rules. The formalism enables rule-based declarative representation of guideline knowledge. So far, the standard Arden Syntax has found application in decision support [14–16]. Despite the expressive power of the syntax, there are no approaches to conformance checking. The challenge of meaningful alignments is addressed via manually developed alignment steps that are integrated in the respective MLM. The presented approach is applied and evaluated on real patient data.

The remainder of the paper is organized as follows. Section 2 provides background information on the components of our approach. Section 3 describes the

research method for the model generation, the conformance checking approach as well as the alignments and shows how the event log from the patient data is created. Section 4 presents the implementation and the results. In Sect. 5, the findings are discussed and Sect. 6 concludes the paper.

2 Fundamentals

2.1 Conformance Checking and Alignments

Conformance checking describes the process of identifying discrepancies between the desired behavior of the process depicted by the process model and the real behavior of the process depicted by the event log [4]. Standard conformance checking only considers the control-flow perspective [17] but event logs often contain information, that go beyond the ordering of events such as time or data related information [17]. Since multiple perspectives can be considered, deviations are not only possible in the ordering of the events but also in other perspectives. One type of conformance checking is *local conformance checking*. This type describes the process of checking the conformance by using a set of independent rules regarding the process. Therefore, only specific parts of the process are checked and not the process as a whole. These rules are often defined in linear temporal logic or in declarative modeling languages [17].

Most state-of-the-art conformance checking techniques are using *alignments* [18]. An alignment can be seen as mapping of the process model capturing the desired behavior and the event log recording the behavior that occurs in reality [6]. Also, the concept of alignments is not process modeling language specific and thus can be defined for nearly all modeling languages [6].

For the creation of an alignment, the events recorded in the event log are mapped to process steps in the process model. Alignments are defined by moves. A *log move* is executed when an event is recorded in the event log that does not occur in the model. A *model move* is executed when an event that is required by the process model is missing in the event log. If the mapping is correct from a control flow perspective but violates a condition from a data perspective, an *incorrect synchronous move* is executed. If the control flow and data perspective match, it is called a *synchronous move*. Using these alignment moves, it is possible to identify the events that violate the process execution [19–21].

2.2 Medical Logic Modules and Arden Syntax

Medical Logic Modules (MLMs) were developed with the aim of presenting medical knowledge in self-contained units, readable by humans and interpretable by computers, and transferable to other clinics [22,23]. The Arden Syntax is a declarative, HL7-standardized, open implementation of MLMs [13,24]. It was developed for the exchange of medical knowledge. In the following, we interpret the term MLM as MLM in the Arden Syntax. MLMs are text files arranged in discrete slots. Each slot has a name (e.g., `version`) followed by a colon and the

body (e.g., 2.0) and ends with a semicolon. Depending on the slot, the body can contain free text, code, or structured data. To improve readability, MLMs are divided into three categories: *maintenance*, *library*, and *knowledge* [23].

Maintenance contains slots not directly related to medical knowledge, rather information such as title, author, and version. The library category contains slots related to the source of the modeled knowledge, keywords and textual explanations. Knowledge category slots describe what the module does. It is divided into three sections. First, the *data* slot, which can be used to retrieve patient data from a database. Second, the *evoke* slot, which specifies conditions when a module is activated. Last, the *logic* slot contains the actual medical rule or medical condition to be evaluated [22]. Therefore, the Arden Syntax provides extensive operators. Such operators as **before**, **after**, **within same day or n days before/after** natively allow addressing the time perspective [25]. *Action* slot defines what to do when the logic slot is evaluated against **true** [22]. For example, an MLM could be evoked when a new patient is admitted. In the logic slot, it could be checked whether the patient has a food intolerance. If the condition was evaluated as **true**, a request for nutritional counseling could be entered into the hospital information system (HIS) database.

3 Materials and Methods

The methodology is outlined in Fig. 1. In the first step, the guideline knowledge to be used is selected and then transformed into the declarative model. The focus is on the complete and correct transformation of the guidelines' medical knowledge formulated in free text into the declarative, computer-interpretable representation. In parallel, the event log is created from the patient data. Finally, conformance checking is performed where alignments are applied until the trace is conforming to the model, and the degree of guideline compliance can be evaluated based on the results.

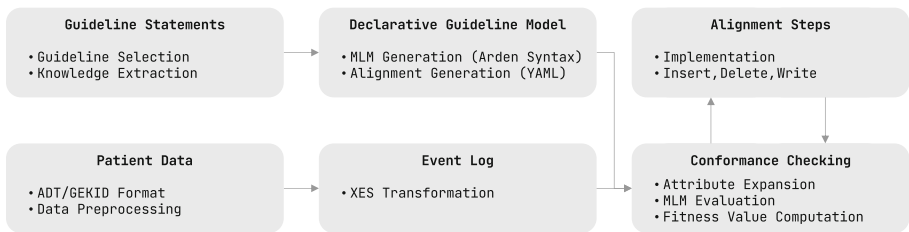


Fig. 1. Methodology outline.

3.1 Generation of the MLM Rule Base

The declarative approach of MLMs and Arden Syntax is used to represent the guideline knowledge as a declarative rule-based model for conformance checking.

For this purpose, an MLM is created for each of the guideline statements. Thus, the guideline is represented by a set of MLMs. MLMs offer the advantage that, despite their expressive power, they are readable in parts even for domain experts after a short training [13]. In the context of clinical guidelines, the expressiveness results from the already existing fields of the MLM structure covering the majority of the information fields of a guideline statement, the extensibility of these by new data fields, and the usability of the Arden Syntax and SQLite.

In principle, MLMs are designed to be evoked by an external call, an event or an evoke condition (*evoke slot*). During an evocation the logical part is evaluated (*logic slot*) and depending on the result an action can be executed (*action slot*) [13]. The approach of this case study uses the expressive power of MLMs to formulate rules for detecting deviations based on the events of the log. Therefore, the evoke slot is used to define and subscribe to events of the event log that trigger the respective MLM. Since the data perspective also can trigger MLMs in addition to the control flow perspective, data attributes are also interpreted as independent events, so-called *attribute writes*, in the evocation process. In the present conformance checking approach, the use of logic and action slots is redefined to check the compliance of treatment cases to a given guideline statement and return the appropriate alignment. For this purpose, the entire conformance with the guideline statement is checked in the logic slot and the result is returned. A distinction is made between conforming and non-conforming traces. Conforming traces only return the describing metadata for the examined statement. Non-conforming traces return previously by domain experts defined alignment steps (see Sect. 3.4). In the action slot, the conformance is returned as a boolean value together with the alignment generated in the logic slot.

The process of transforming clinical guideline knowledge into explicit, computer-interpretable formalizations is complex and fraught with responsibility. Misinterpretations can have serious consequences for patient health [26]. For correct interpretation of guideline statements, implicit background knowledge of different fields of expertise is required. To ensure the correct transformation of guideline statements into MLMs, the CGK4PM framework was applied [27]. Following this, an interdisciplinary team of doctors and computer scientists was assembled. These worked together in the iterative approach of formalization and validation described in CGK4PM. Formalization took place in workshops, in which 5 statements were discussed in each session. Validation was done firstly by the doctors in workshops and secondly technically, with test patients representing compliant and non-compliant patients. After the joint formalization workshops, the cases were transferred to the Arden Syntax by the computer scientists and are validated using unit tests.

Exemplary, a part of the German clinical guideline for the treatment of malignant melanoma of the Association of the Scientific Medical Societies (AWMF)¹ was transferred into MLMs. The focus was on Chap. 4 (Diagnostics and Therapy in Primary Care) and Chap. 6 (Diagnostics and Therapy in Locoregional Metastasis). For this purpose, 39 guideline statements were randomly selected

¹ <https://www.awmf.org>.

and each statement was transferred into an MLM. Five of these statements were optional and therefore not relevant for conformance and not transformed. Nine statements were not transitioned due to lack of data in the underlying database, like structured information about uncertainties and suspicions.

The model results in 25 individual and independent MLMs and thus represents as many guideline statements. In the knowledge category, the data, evoke, logic and action slots are used for the actual evaluations. In the data slot, all data needed to check compliance is retrieved from the database and provided to the evoke, logic, and action slots for evaluation. Moreover, the events relevant for MLM evocation are defined here. The data query step turned out to be the most difficult and error-prone step, since at this point the information required for the logic part must be specifically queried using SQL syntax and Arden Syntax.

3.2 Event Log

To maintain comparability, the same dataset as in [9] was used. The dataset contains data on five patients with malignant melanoma and their treatment at Münster University Hospital. Patients were informed about the use of data and gave their written consent. For privacy reasons, the data were made available to the research in anonymized form. The data are in the uniform XML format of cancer registries in Germany, ADT/GEKID². Among other things, it contains data on treatment, diagnosis, histology, and cancer staging. The major advantage of using data in the format of the basic data set is that it is used by all German cancer registries and results are thus transferable and comparable. In addition, all entries contain a timestamp, which enables procedural use.

For conformance checking, the data is converted to the XES event log format [28]. For this purpose, a generic XML to XES converter was implemented in Python and configured to convert ADT/GEKID data to XES. During the pre-processing in [9], an extensive filtering of activities according to relevance and irrelevance for guideline compliance (e.g., psychological counseling) was necessary because conformance checking with an imperative model was used. Contrarily, in the present approach, conformance checking is done on the whole log without filtering since the declarative method by itself considers only relevant data. However, the naming of attributes and events in the event log had to be aligned with the model.

3.3 Conformance Checking

MLMs subscribe to events, these can relate to both control flow level events and data level write operations. Therefore, the trace is extended so that attribute writes are also interpreted as events for evocation and thus conformance checking is also evoked for each write. It is important to note that the MLMs subscribe to the writing of a specific attribute value, e.g., for the attribute ICD-Code with the value C43.9 the event `write_icd_code_c43.9` would be triggered.

² <https://www.gekid.de/adt-gekid-basisdatensatz>.

In the evoked MLMs, conformity is first calculated locally and expressed as *conform* (1) or *non-conform* (0). Then, a global computation of a fitness value is performed across all MLMs. It should be noted that not every MLM is evoked for every trace. Therefore, the calculation is performed including only the evoked MLMs. With respect to the fitness dimension of conformance checking, we quantify the conformance of a trace σ from Log L with an MLM model M . Therefore, we introduce the fitness function $fitness(\sigma, M)$. This returns a value between 0 and 1, quantifying the degree of conformance of trace σ with the model M . Here, 1 represents optimal fitness and 0 represents very poor fitness.

Definition 1 (Fitness). *Let MLM be the universe of all MLMs, $M \subseteq MLM$ be the declarative model, and σ be a trace. Then $M_\sigma \subseteq M$ is the short form of all MLMs evoked for sigma. The function $eval : M \times \Sigma \rightarrow \{0, 1\}$ evaluates whether a trace conforms to an MLM or not or was not evoked. The fitness is defined as:*

$$fitness(\sigma, M) = \frac{\sum_{i=1}^n eval(\sigma, M'_i)}{|M'_\sigma|}$$

3.4 Alignment

In [9], it is shown that from both a medical and guideline compliance use case perspective, it is not reasonable to align the model. Therefore, the alignment is done under the assumption that the model is correct and only the trace is adjusted. Thus, a log move is always handled with a DELETE operation in the event log, a model move with an INSERT in the event log, and an incorrect synchronous move is handled with a data WRITE in the event log. For this purpose, alignment steps have been manually defined for each MLM. Each alignment step consists of a set of m alignment operations. Each operation consists of an operator, a positional relation, a value, at least one timestamp as position information and optional parameters concerning the operation.

Definition 2 (Alignment Steps). *Let $O = \{INSERT, DELETE, WRITE\}$ be the set of all alignment operations, $R = \{BEFORE, AFTER, AT, BETWEEN\}$ be the set of positional relations, $S = \{ATTRIBUTE, EVENT\}$ be the set of the subjects involved, V be the universe of all possible attribute values, and TS be the universe of all possible timestamps. Then an alignment proposition is a quadruple (s, o, v, p) with $s \in S, o \in O, v \in V$ and $p \in R \times TS^*$.*

In the MLMs, the alignment steps are YAML encoded. The alignment engine (AE) then resolves the received alignment steps and performs the operations described in them. For this purpose, based on the position information in the form of one or two timestamps and the positional relation for each INSERT, a matching position is searched for in the trace. Either the earliest conforming position, the last position or a random position can be selected. In case of a DELETE or WRITE, the exact event must be identified by timestamp and activity name. If no conforming position can be found, the alignment is aborted. If the

alignment was successful, the trace is run through again to check the effect of the alignment on previous events and their conformity.

Figure 2 shows an outlined example of an alignment calculated with an MLM. In this MLM, event C should occur after event B. The MLM is evoked since event B occurs in the trace. Since event C does not occur, the logic of the MLM concludes to **false** and the alignment operation defined in the **else** block is executed. Thus, an event is generated with the value of C and is inserted after the B event. To find the correct position for inserting, the timestamps of the events are used. The end result is the insertion of event C in the trace and the detection of the guideline violation due to the initial absence of C.

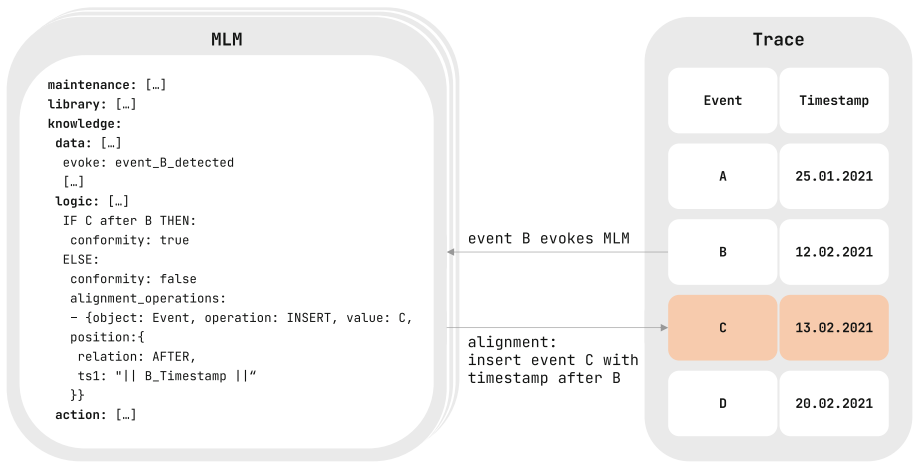


Fig. 2. The outlined example shows a trace that violates the MLM, which states that event B must be followed by event C. The alignment step modeled manually in the MLM indicate that C is to be inserted after B.

4 Implementation

The implementation was split into the central management of the rule-based model and events in a server component (*MLM server*) and the local implementation of the rules in the client (*Alignment Engine (AE)*). The interaction of the components is depicted in Fig. 3. In essence, the MLM server triggers the MLMs relevant to an event and returns the results to the AE. For this purpose, the MLM server manages subscriptions of events by the MLMs, receives traces and events from the AE, triggers the MLMs subscribing to the received event, summarizes the results of all evaluations and returns them to the AE. The server component was implemented in Python, using the Arden2ByteCode compiler.

The AE expands traces so that attributes are transformed to attribute write events. Sequentially, the AE calls the MLM server for each of the events. For this purpose, the trace is transferred to a temporary SQLite database. In case

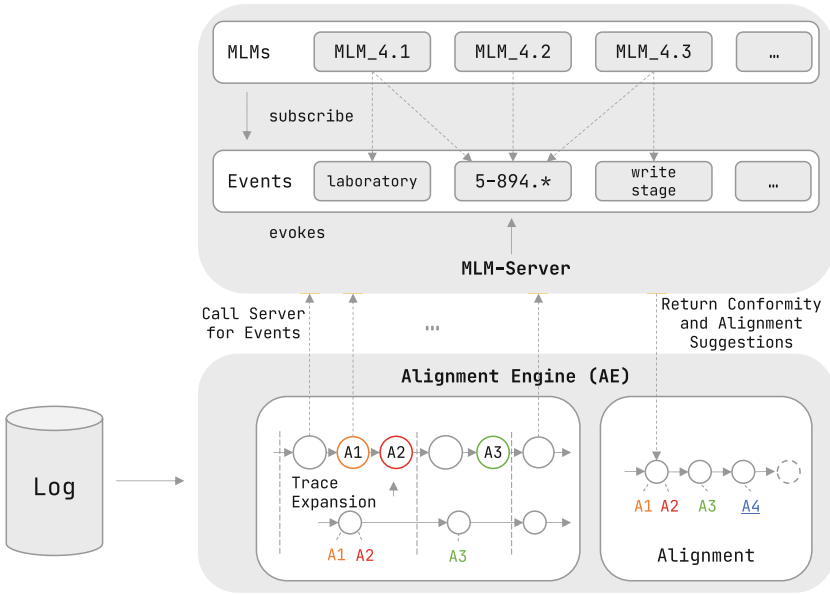


Fig. 3. The AE expands the traces from the log and generates an event for each write operation of an attribute. For each event, the MLM server is called, which then evokes MLMs subscribed to the respective event. Based on the returned manually modeled alignment steps, the AE generates a new compliant trace without changing the model.

of an MLM violation, the alignment steps provided by the server are converted into YAML. To do this, the AE resolves the positional relations to the timestamps given in the alignment steps and thus identifies the possible position(s) in the trace to implement the alignment. The alignment process is restarted with the aligned trace to consider also implications that have arisen through the implementation of the alignment.

The approach was evaluated based on five real world treatment cases (see Sect. 3.2). Each trace consists of 29 events on average, and each event has eight attributes on average. In total, counting only the first evocations per MLM per trace, 47 evocations were performed. Of these, 21 checks failed and 26 were compliant. All alignments were successfully applied by the AE. A total of 21 alignments were executed, 15 deletes and six inserts. No verification failed on the four MLMs that write attributes based on alignments. On average, the traces have a fitness of 0.478, while the worst fitness is 0.2 and the best is 0.66.

5 Discussion

Considering the results, significant advantages could be identified in the use of the declarative modeling approach. In a qualitative comparison with the imperative approach from [9], working on the same guideline and same data set, most

limitations mentioned there could be addressed with the declarative approach. However, mapping ambiguities remain a challenge [9]. This means that it is not always possible to determine whether the event was executed because of a particular attribute, or whether that attribute is just an unnoticed byproduct. The Arden Syntax natively allows inclusion of the time perspective and implementation of softer temporal constraints like delays. The declarative nature of the approach made it possible to handle high variability of processes. In addition, the semantic meaningfulness of the alignments could be improved by pre-modeled steps. Limitations still exist in the alignment of events, especially in processes running in parallel in one instance, e.g., the treatment of comorbidities during a hospital stay. Furthermore, explicit mapping of repetitive activities to activities in the model remains a challenge. In addition, the fitness score is still not very meaningful, as it does not take into account the severity of the deviation.

The transformation of the guideline to the rule-based model proved to be complex. This is due to the specific knowledge to be modeled and to the lack of verification tools that, e.g., search for contradictions or interdependencies in the rule base. The advantage of modeling with Arden Syntax was that after some practice, the domain experts were also able to read and edit parts of the MLMs.

The use of manually modeled alignment steps required a high initial effort. From the user's perspective, however, these addressed the deviations semantically correct contrary to other algorithmic approaches, e.g., the one used in [9]. However, one limitation related to alignments is the runtime of the alignment implementation in the trace. Since each alignment can also affect previous events, the entire trace must be checked again for conformance.

6 Conclusion

In this paper, we focused on the MLM-based declarative modeling of guidelines and a conformance checking approach that meaningfully considers guideline characteristics and medical context. The declarative modeling approach excelled in its expressiveness and its characteristic of dealing with the variability of medical processes. The case study showed promising initial results by considering medically meaningful logic and individually defined alignment steps. These novel features are the key difference from conventional conformance checking algorithms and contribute to medical application of declarative process mining. However, there are still severe limitations in the scope of the expressions (e.g. before/after), which only refer to the timestamp and not to the activities.

We plan to add more guideline statements to the model for further evaluation and identification of further challenges. Moreover, it should be investigated to what extent ArdenML (an XML schema for expressing MLMs) and its possibility to connect to other XLM-based HL7 standards can be beneficially integrated into the architecture. Furthermore, we want to improve the conformance checking approach and tackle the discussed limitations such as parallel and repetitive activities. Since the algorithm can currently only evaluate if an activity is compliant or not, we plan to extend it with a fuzzy function. Thus, we hope to

make more meaningful statements about the treatments by differentiating more precisely between guideline violations such as temporal deviations. These details that distinguish the patients with regard to treatment outcomes can be highly relevant for physicians in their reasoning process.

Acknowledgments. The research is partially funded by the German Federal Ministry of Education and Research (BMBF) in the project DaTreFo under the funding code 16KIS1644.

References

1. Lohr, K.N., Field, M.J.: Clinical practice guidelines: directions for a new program. Volume 90–08 of Publication IOM. National Academy Press, Washington (1990)
2. Forsner, T., Hansson, J., Brommels, M., Wistedt, A.A., Forsell, Y.: Implementing clinical guidelines in psychiatry: a qualitative study of perceived facilitators and barriers. *BMC Psychiatry* **10**, 8 (2010)
3. Landfeldt, E., et al.: Compliance to care guidelines for Duchenne muscular dystrophy. *J. Neuromuscul. Dis.* **2**(1), 63–72 (2015)
4. Rovani, M., Maggi, F.M., de Leoni, M., van der Aalst, W.M.: Declarative process mining in healthcare. *Expert Syst. Appl.* **42**(23), 9236–9251 (2015)
5. Gatta, R., et al.: Clinical guidelines: a crossroad of many research areas. challenges and opportunities in process mining for healthcare. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) *BPM 2019. LNBIP*, vol. 362, pp. 545–556. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37453-2_44
6. van der Aalst, W.M.P.: *Process Mining: Data Science in Action*, 2nd edn. Springer, Heidelberg (2016)
7. Van der Aalst, W., Adriansyah, A., van Dongen, B.: Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisc. Rev.: Data Min. Knowl. Discov.* **2**(2), 182–192 (2012)
8. Geleijnse, G., et al.: Using process mining to evaluate colon cancer guideline adherence with cancer registry data: a case study. In: American Medical Informatics Association Annual Symposium, AMIA 2018, San Francisco, California, USA, 3–7 November 2018. American Medical Informatics Association (2018)
9. Grüger, J., Geyer, T., Kuhn, M., Braun, S.A., Bergmann, R.: Verifying guideline compliance in clinical treatment using multi-perspective conformance checking: a case study. In: Munoz-Gama, J., Lu, X. (eds.) *ICPM 2021. LNBIP*, vol. 433, pp. 301–313. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98581-3_22
10. Bottrighi, A., et al.: A hybrid approach to clinical guideline and to basic medical knowledge conformance. In: Combi, C., Shahar, Y., Abu-Hanna, A. (eds.) *AIME 2009. LNCS (LNAI)*, vol. 5651, pp. 91–95. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02976-9_12
11. Burattin, A., Maggi, F.M., Sperduti, A.: Conformance checking based on multi-perspective declarative process models. *Expert Syst. Appl.* **65**, 194–211 (2016)
12. Pestic, M., van der Aalst, W.M.P.: A declarative approach for flexible business processes management. In: Eder, J., Dustdar, S. (eds.) *BPM 2006. LNCS*, vol. 4103, pp. 169–180. Springer, Heidelberg (2006). https://doi.org/10.1007/11837862_18
13. Samwald, M., Fehre, K., de Bruin, J., Adlassnig, K.P.: The Arden syntax standard for clinical decision support: experiences and directions. *J. Biomed. Inform.* **45**(4), 711–718 (2012)

14. Anand, V., Carroll, A.E., Biondich, P.G., Dugan, T.M., Downs, S.M.: Pediatric decision support using adapted Arden syntax. *Artif. Intell. Med.* **92**, 15–23 (2018). Special Issue on Arden Syntax
15. Schuh, C., de Bruin, J.S., Seeling, W.: Clinical decision support systems at the Vienna general hospital using Arden syntax: design, implementation, and integration. *Artif. Intell. Med.* **92** (2018). Special Issue on Arden Syntax
16. Adlassnig, K.P., Fehre, K., Rappelsberger, A.: Fuzzy-Arden-syntax-based, vendor-agnostic, scalable clinical decision support and monitoring platform. *Stud. Health Technol. Inform.* **216**, 1111 (2015)
17. Caron, F.: Business process analytics for enterprise risk management and auditing. Ph.D. thesis, Katholieke Universiteit Leuven, Belgium (2013)
18. Dunzer, S., Stierle, M., Matzner, M., Baier, S.: Conformance checking: a state-of-the-art literature review. In: Betz, S. (ed.) *Proceedings of the 11th International Conference on Subject-Oriented Business Process Management*, pp. 1–10. Association for Computing Machinery, New York (2019)
19. Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B.F., van der Aalst, W.M.P.: Alignment based precision checking. In: La Rosa, M., Soffer, P. (eds.) *BPM 2012. LNBP*, vol. 132, pp. 137–149. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36285-9_15
20. de Leoni, M., van der Aalst, W.M.P.: Aligning event logs and process models for multi-perspective conformance checking: an approach based on integer linear programming. In: Daniel, F., Wang, J., Weber, B. (eds.) *BPM 2013. LNCS*, vol. 8094, pp. 113–129. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_10
21. Mannhardt, F.: Multi-perspective process mining. Ph.D. thesis, Technische Universiteit Eindhoven (2018)
22. Arkad, K., Gill, H., Ludwigs, U., Shahsavar, N., Gao, X.M., Wigertz, O.: Medical logic module (MLM) representation of knowledge in a ventilator treatment advisory system. *Int. J. Clin. Monit. Comput.* **8**(1) (1991)
23. Hripcsak, G.: Writing Arden syntax medical logic modules. *Comput. Biol. Med.* **24**(5), 331–363 (1994)
24. Vetterlein, T., Mandl, H., Adlassnig, K.P.: Fuzzy Arden syntax: a fuzzy programming language for medicine. *Artif. Intell. Med.* **49**(1), 1–10 (2010)
25. Jenders, R.A., Haug, P., Adlassnig, K.P.: HL7 Arden syntax: implementation guide. Technical report, Health Level Seven International, Ann Arbor, USA (2019)
26. Patel, V.L., Allen, V.G., Arocha, J.F., Shortliffe, E.H.: Representing clinical guidelines in GLIF: individual and collaborative expertise. *JAMIA* **5**(5), 467–483 (1998)
27. Grüger, J., Geyer, T., Bergmann, R., Braun, S.A.: CGK4PM: towards a methodology for the systematic generation of clinical guideline process models and the utilization of conformance checking. *BioMedInformatics* **2**(3), 359–374 (2022)
28. IEEE: Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams. *IEEE Std 1849-2016*, pp. 1–50 (2016)



Improving Declarative Process Mining with *a Priori* Noise Filtering

Axel Kjeld Fjelrad Christfort¹(✉), Søren Debois^{2,3}() and Tijs Slaats¹()

¹ Department of Computer Science, University of Copenhagen,
Copenhagen, Denmark

`axel@christfort.dk`, `t.slaats@di.ku.dk`

² Department of Computer Science, IT University of Copenhagen,
Copenhagen, Denmark

`debois@itu.dk`

³ DCR Solutions A/S, Copenhagen, Denmark

`debois@dcrsolutions.net`

Abstract. In this paper, we report the results of an exploratory study into the efficacy of noise filtering in improving the accuracy of declarative process mining. We apply the double-granularity mixed-dependency filtering algorithm as introduced by [9], to the DisCoverR declarative miner [1], and parameter optimise it to only perform coarse-grained filtering. However, while noise filtering appears promising on the surface, one might worry that the outlier behaviour allowed by declarative models may be wrongly classified as noise and removed. To test the efficacy of noise filtering from both perspectives, we applied DisCoverR with noise filtering to two data sets: the process log collection used in the PDC2020 process discovery contest, emulating “real-world” scenarios; and a synthetic set of logs known to exhibit (non-noise) outlier behaviour. We find that on the contest data sets, noise filtering significantly increases accuracy (on average 23% points), obtaining exploratory evidence that noise filtering may improve declarative miner performance; on the synthetic logs we showcase examples where noise is filtered, while outlier behaviour remains untouched.

Keywords: Noise filtering · Process discovery · Declarative process models · DCR Graphs

1 Introduction

A key challenge in process mining is the soundness of input data. Logs are often extracted from legacy systems that were not designed specifically for logging process runs and various sources of *noise* can occur. For example, system malfunctions or bugs in program code may cause events to be duplicated, removed, or stored in the wrong order. In other cases the process may not always have

Work supported by the Innovation Fund Denmark project *EcoKnow* (7050-00034A), Digital Research Centre Denmark and DCR Solutions A/S.

© Springer Nature Switzerland AG 2023

C. Cabanillas et al. (Eds.): BPM 2022, LNBP 460, pp. 286–297, 2023.

https://doi.org/10.1007/978-3-031-25383-6_21

been executed successfully, but logged nonetheless. If the database does not indicate that the process failed, then such “bad” traces may creep into our data and be mined as if they are expected good behaviour of the system.

Noise differs from *infrequent behaviour* in that while both concepts can result in rare dependencies between events, infrequent behaviour truthfully represents the process. As such, filtering intentional infrequent behaviour will decrease the accuracy of the mined model, while filtering noise will increase the accuracy.

When dealing with noise filtering for process discovery, two approaches can be discerned. The first are process discovery algorithms which have some form of noise filtering ingrained. For example the log skeleton miner [11] can filter out noise as part of their core algorithm. The second approach is to filter the log in an *apriori* manner, i.e. the log is first filtered for noise by a filtering algorithm and afterwards mined by a discovery algorithm. This second approach allows the discovery algorithm to assume noise-free input data.

One example of the second approach is the DisCoveR miner [8], which uses the assumption of noise-free input data to mine perfectly fitting and highly accurate models with a remarkably high run-time performance [1]. It was particularly successful in the Process Discovery Contest 2019¹, where it achieved a 96.1% accuracy [1]. While the 2019 contest already contained some forms of noise, these did not appear to affect the accuracy of the miner significantly. However in 2020, the addition of a new form of noise significantly reduced the accuracy of the miner. Algorithms were scored based on the harmonic mean of *true positive rate* and *true negative rate* of the models they produced, and DisCoveR scored an average of 78% on across all logs without noise, and only 21% across the logs with added noise.

This paper reports on our attempts to improve the results of the DisCoveR miner through the application of apriori noise filtering. In Sect. 2 we present the chosen noise-filtering method, originally introduced in [9], which performs double granularity (on both events and traces) filtering based on mixed dependency. In Sect. 3 we show how the algorithm has been parameter optimised to only perform coarse-grained filtering. Furthermore we show, using synthetic logs, how the algorithm successfully distinguishes between infrequent behaviour and noise. In Sect. 4 we show how this method of noise-filtering significantly improved the score of DisCoveR for the 2020 Process Discovery Contest, from 50% on all logs to 73%. Finally we conclude in Sect. 5.

Related Work

Several ways of diminishing the impact of noise in process discovery have been proposed in the last decade. These can be split into two categories, those that aim to detect and counteract noise during process mining, and those that aim to preprocess logs to remove noise independently of any miner. Of the first category, several miners, such as the Heuristics Miner [12] and Fodina [10], claim a build-in tolerance for noise [3]. Other miners such as the Inductive visual Miner [6],

¹ <https://icpmconference.org/2019/process-discovery-contest>.

Directly Follows visual Miner [5], and Log Skeleton Miner [8] have noise filtering as an input parameter.

Of the second category, a recent approach proposed in [3] constructs an automaton that initially accepts all traces in a given log. The arcs of this automaton are then filtered based on relative frequency, and finally fine-grained filtering is applied by replacing each trace in the log with the longest replayable subtrace on the filtered automaton. As the algorithm requires the NP-hard identification of the minimum anomaly-free automaton, we did not consider it suitable for our purposes.

2 Noise Filtering Algorithm

In this section the noise filtering algorithm we investigate will be presented. This was introduced in [9], however, as will be shown, we deviate from the original in minor details, due to discoveries found during the investigation. The approach consists of both fine-grained filtering of events in traces, and coarse-grained filtering of entire traces. The terms needed for each will be described in Sects. 2.1 and 2.2, followed by them being combined to form the algorithm in Sect. 2.3.

2.1 Fine-Grained Filtering

The approach uses two key concepts to detect noise: the *Directly Follows Degree (DFD)* and the *Predecessor/Successor Density (D_{pre} and D_{suc})*. These are in turn used to define *Local- and Global Dependency*, which when combined into *Mixed Dependency* become a metric capable of fine-grained distinguishing of noise and infrequent events.

DFD is a measure of how closely two events are related in the log, and as such events with high *DFD* should not be filtered. However, infrequent events will always have low *DFD* as they occur infrequently in the log. Therefore D_{pre} and D_{suc} are introduced. These are measures of the average *DFD* between all *predecessors* and *successors* respectively, and while infrequent events will always occur infrequently, and thus have low D_{pre} and D_{suc} , events that occur infrequently due to being noise will occur frequently elsewhere, and thus have higher D_{pre} and D_{suc} . So by favoring high *DFD* and low D_{pre} and D_{suc} , noise should be filtered, while infrequent events are ignored.

Mathematically this can be defined as follows. Consider an event log \mathcal{L} comprising traces σ_i for $0 \leq i < n$ over event set \mathcal{E} . For $e, f \in \mathcal{E}$ we say that f *directly follows* e iff there exists a trace $\sigma_k \in \mathcal{L}$ s.t. f appears directly after e in σ_k , i.e. with no event in between. The measure $DFD(e, f) \in \mathbb{N}$ is defined as the number of times f *directly follows* e in \mathcal{L} .

If f *directly follows* e we say that e is a predecessor to f and conversely that f is a successor to e . The predecessor set of an event e , $U_{pre}(e)$, is then defined as the set of all predecessors to e , and the successor set of e , $U_{suc}(e)$ is defined as the set of all successors to e . With this terminology, we define predecessor density

as the average *DFD* between all predecessors and e and successor density as the average *DFD* between f and all successors.

$$\begin{aligned}
 D_{pre}(f) &= \frac{\sum_{e \in U_{pre}(f)} DFD(e, f)}{|U_{pre}(f)|} \\
 D_{suc}(e) &= \frac{\sum_{f \in U_{suc}(e)} DFD(e, f)}{|U_{suc}(e)|}
 \end{aligned} \tag{1}$$

From this, local dependency between two events, $e, f \in \mathcal{E}$, is defined as seen in (2). As can be seen from the equation, this is normalised and tends to 1 as $DFD(e, f)$ tends towards ∞ . Furthermore, (2) consists of two terms that are each subtracted from 1. The first of these will decrease as the *successor density* of e decreases, likewise the second link will decrease as the *predecessor density* of f decreases.

$$\begin{aligned}
 Dep_{local}(e, f) &= \\
 &1 - \frac{1}{(1 + \exp((DFD(e, f) - D_{suc}(e)) \cdot \frac{4}{D_{suc}(e)})) \cdot 2} \\
 &\quad - \frac{1}{(1 + \exp((DFD(e, f) - D_{pre}(f)) \cdot \frac{4}{D_{pre}(f)})) \cdot 2}
 \end{aligned} \tag{2}$$

Global dependency is defined in (3). Whereas local dependency compares the dependency of two events to their successors and predecessors, global dependency compares their dependency to that of all event pairs in the graph. [9] suggests using a ζ of 0.02, which also provided the best results in our experiments.

$$\begin{aligned}
 Dep_{global}(e, f) &= 1/(1 + e^{(1-DFD(e,f)/\theta)}) \\
 \theta &= \text{Max}(DFD(e_x, e_y)) \\
 \text{s.t. } e_x, e_y &\in \mathcal{E} \wedge \frac{DFD(e_x, e_y)}{\sum_{e_k, e_t \in \mathcal{E}} DFD(e_k, e_t)} < \zeta
 \end{aligned} \tag{3}$$

These can then be combined to form mixed dependency as seen in Eq. 4, where α denotes the weighing of the two forms of dependency used and is given as 0.5 in [9].

$$Dep_{mixed}(e, f) = \alpha \cdot Dep_{local}(e, f) + (1 - \alpha) \cdot Dep_{global}(e, f) \tag{4}$$

Using this, a mixed dependency matrix is constructed, which contains the mixed dependencies for all events in the log. This mixed dependency matrix will play a role in both defining the coarse-grained filtering, and in the algorithm itself.

2.2 Coarse Grained Filtering

Fine grained filtering, however, cannot detect all forms of noise. Consider noise in the form of events being removed from a trace. We cannot hope to remedy

this by removing more events from the trace. Therefore a way of coarse-grained filtering is introduced, which when the estimated level of noise in a trace becomes too high, the entire trace is filtered. This is done in the form of an abandon factor $f_{abandon}$ that is initialised to 1 for each trace. Each time an event f is filtered due to $Dep_{mixed}(e, f)$ being too low, this is altered by the penalty function, as seen in (5), where f_{punish} denotes the punishment factor. If the abandon factor of a trace falls below a certain abandon threshold $\mathcal{T}_{abandon}$ the entire trace is filtered.

$$\mathcal{P}(f_{abandon}, Dep_{mixed}(e, f)) = f_{abandon} \cdot f_{punish}(1 + 2 \cdot (f_{punish}^{-1} - 1) \cdot Dep_{mixed}(e, f)) \quad (5)$$

2.3 The Algorithm

With the core concepts introduced, we can present the algorithm as shown in Algorithm 1, which takes a log \mathcal{L} , an abandon threshold $\mathcal{T}_{abandon}$, a dependency threshold $\mathcal{T}_{dependency}$, and a punishment factor f_{punish} . Note that we have taken the liberty of reformulating the pseudo-code of [9], to make bridging the gap between pseudo-code and actual implementation for our experiments a little bit easier. The core semantics of the approach should, however, remain unchanged.

It starts by computing the mixed dependency matrix as described above. Then it iterates the traces in the log, always adding the first event in each trace, and initialising $f_{abandon}$ to 1. Next it iterates the events in the trace, checking if the mixed dependency between them is higher than the dependency threshold. If this is the case, the event is added, and if not, the abandon factor is updated and checked against the abandon threshold. Note that while f changes to the next event every iteration, e is always the last event added to the filtered trace. This ensures that whenever the dependency of a new event is checked, the predecessor is not the event that was before it in the trace, but the last event that was not filtered.

Furthermore, we propose a *filter threshold* of 50% to ensure the filtering has been valid. This is based on the fact that for few of our training logs, more than 50% of the log had been filtered, yielding significantly worse results than with no filtering. As such, we believe that the fact that over 50% of a log has been filtered is a sign of a failed filtering attempt, and thus, the original log should be returned.

2.4 Parameter Optimisation

Since there are two constants not given in the paper, the abandon threshold $\mathcal{T}_{abandon}$ and the punishment factor f_{punish} , we parameter optimised this algorithm on the PDC2020 data-set. However, due to the fact that it seems that there is a clear effect in adjusting the dependency threshold $\mathcal{T}_{dependency}$, this will also be parameter optimised. The chosen values for each parameter can be seen below in Fig. 1. For f_{punish} and $\mathcal{T}_{abandon}$ almost the full spectrum is

Algorithm 1 Double granularity filtering based on mixed dependency

```

1:  $A_{MDM}^L = \text{buildMixedDependencyMatrix}(\mathcal{L})$ 
2: for all Trace  $\sigma$  in  $\mathcal{L}$  do
3:    $e = \sigma.\text{getFirstEvent}()$ 
4:    $e_{\text{end}} = \sigma.\text{getLastEvent}()$ 
5:    $\sigma_{\text{filter}}.\text{addEvent}(e)$ 
6:    $f_{\text{abandon}} = 1$ 
7:    $f = e.\text{nextEvent}()$ 
8:   while  $f \neq \text{NULL}$  do
9:     if  $A_{MDM}^L(e, f) \geq \mathcal{T}_{\text{dependency}}$  then
10:       $\sigma_{\text{filter}}.\text{addEvent}(f)$ 
11:       $e = f$ 
12:     else
13:        $f_{\text{abandon}} = \mathcal{P}(f_{\text{abandon}}, \text{Dep}_{\text{mixed}}(e, f))$ 
14:       if  $f_{\text{abandon}} < \mathcal{T}_{\text{abandon}}$  then
15:         continue to next trace
16:       end if
17:     end if
18:      $f = f.\text{nextEvent}()$ 
19:   end while
20:    $\mathcal{L}_{\text{filter}}.\text{addTrace}(\sigma_{\text{filter}})$ 
21: end for

```

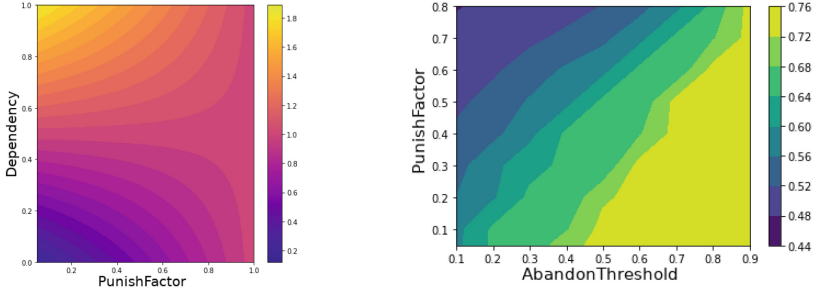
$\mathcal{T}_{\text{abandon}} :$	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
$f_{\text{punish}} :$	[0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
$\mathcal{T}_{\text{dependency}} :$	[0.1, 0.2, 0.3, 0.4, 0.5]

Fig. 1. Parameter space tried during parameter optimization

tried, though adjusted slightly since too many values proved too computationally demanding. For $\mathcal{T}_{\text{dependency}}$ a beneficial effect was only seen when lowering it, and as such only half of the spectrum is tried.

From these a $\mathcal{T}_{\text{dependency}}$ of 0.2 seemed optimal, however there were many combinations of $\mathcal{T}_{\text{abandon}}$ and f_{punish} that yielded the same score. These can be seen in Fig. 2b. Here we note that the yellow area consists of all optimal combinations with the center of mass at $\mathcal{T}_{\text{abandon}} = 0.9$ and $f_{\text{punish}} = 0.05$. Now looking at Fig. 2a we see that for $f_{\text{punish}} = 0.05$ even a dependency of 0.2 will yield $\mathcal{P}(1) \approx 0.6$, which is well below an optimal $\mathcal{T}_{\text{abandon}}$ of 0.9. This means that with these optimal combinations, when the first event is filtered, the trace will also be filtered, essentially degrading the double-granularity filtering to only being coarse-grained.

Since the parameter optimisation resulted in parameters that actually had semantic meaning, *i.e.* the algorithm only performing coarse-grained filtering, we believe that the combination of f_{punish} and $\mathcal{T}_{\text{abandon}}$ that result in this effect, can be considered generally useful parameters. This is also supported by the results of PDC2021, in which DisCoveR scored 95.8% without *a priori* filtering



(a) A heatmap of \mathcal{P} for an $f_{abandon}$ of 1. (b) A heatmap of the score with combinations of $\mathcal{T}_{abandon}$ and f_{punish} , and $\mathcal{T}_{dependency}$ locked to 0.2.

Fig. 2. Results of parameter optimisation

and 96.2% with filtering. This shows that even for a data-set with which the algorithm had no issue with noise, a beneficial effect was still seen with this approach and these parameters.

3 Experiment 1: Synthetic Data

To better understand how and whether the proposed method distinguishes infrequent events from noise, we present in this section the application of the method to two synthetic logs: one in which filtering should occur, and one in which it should not.

Data-Set

We construct data sets to model two distinct situations: one (**infrequent-X**) in which an event X occurs at random with very low frequency; and one (**early-A**) in which an event A occurs near the end of the trace with high probability; and also, but very seldomly, at the beginning of the trace. The intuition here is that the X in **infrequent-X** should *not* be filtered, because low frequency and uniform distribution is not necessarily indicative of noise. Conversely, the A in **early-A** *should* be filtered, but only when occurring early, because very low-frequent deviations from an otherwise clear pattern is indicative of noise.

Both synthetic logs have been generated based on the model shown in Fig. 3. This model specifies that all events collected in the boxes Parts 1, 2, and 3 must be executed at least once; that all events in Part 1 must be executed before any in Part 2 are; and likewise, that all events in Part 2 must be executed before any in Part 3 are. Once Part 3 commences, events from Parts 1 and 2 can no longer be executed. Once all events in Part 3 have been executed at least once, A can execute. Finally, events L , X , and M can occur at any point, with X being an infrequent event, occurring at a rate of approximately $1/20.000$.

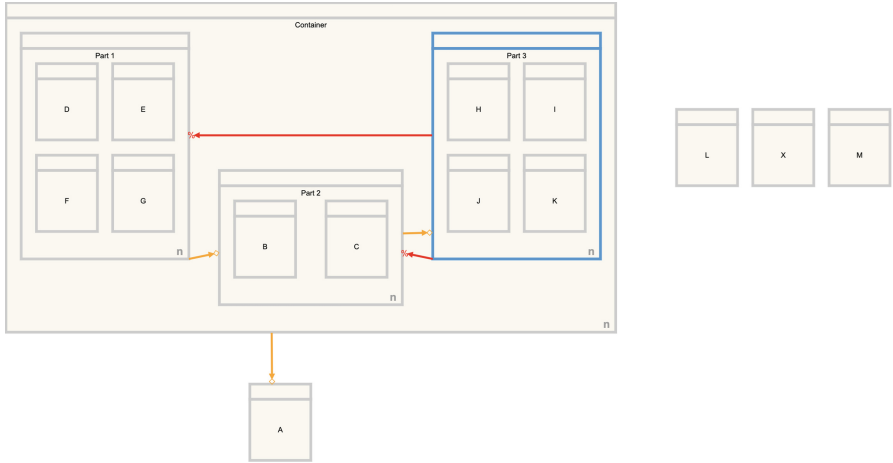


Fig. 3. The underlying model for the synthetic logs expressed in DCR [4] notation.

The log `infrequent-X` is generated directly from this model with no noise added, while the log `early-A` has noise added in the form of A being moved towards the beginning of a few traces.

Our hypothesis is that the traces featuring X, which is strictly infrequent, will not be filtered, while the traces featuring the early A will be filtered, since A occurs frequently towards the end of traces.

Results

Before filtering, both logs contained 950 traces, and afterwards `early-A` contained 857 traces and `infrequent-X` contained 860 traces, corresponding to 9.79% and 9.47% being filtered respectively. Notably, for `infrequent-X`, none of the traces that contained X was filtered, while in `early-A`, all the traces that contained an early A were filtered.

Visualisations of the logs can be seen in Fig. 4. Here we note that for the `infrequent-X` log in Figs. 4a and 4b, the infrequent X's marked with red arrows have not been filtered, and the structure remains similar. For the non-filtered `early-A` log in Fig. 4c, we see the large area of infrequent paths marked in red, starting at the A's marked with arrows. In the filtered log in Fig. 4d, we see that these have all been filtered, and the structure now more closely resembles that of the other logs generated by the same model.

This strongly indicates that while some filtering occurs no matter what, infrequent behaviour is not targeted for filtering simply by being infrequent, while infrequent behaviour that occurs frequently elsewhere is targeted.

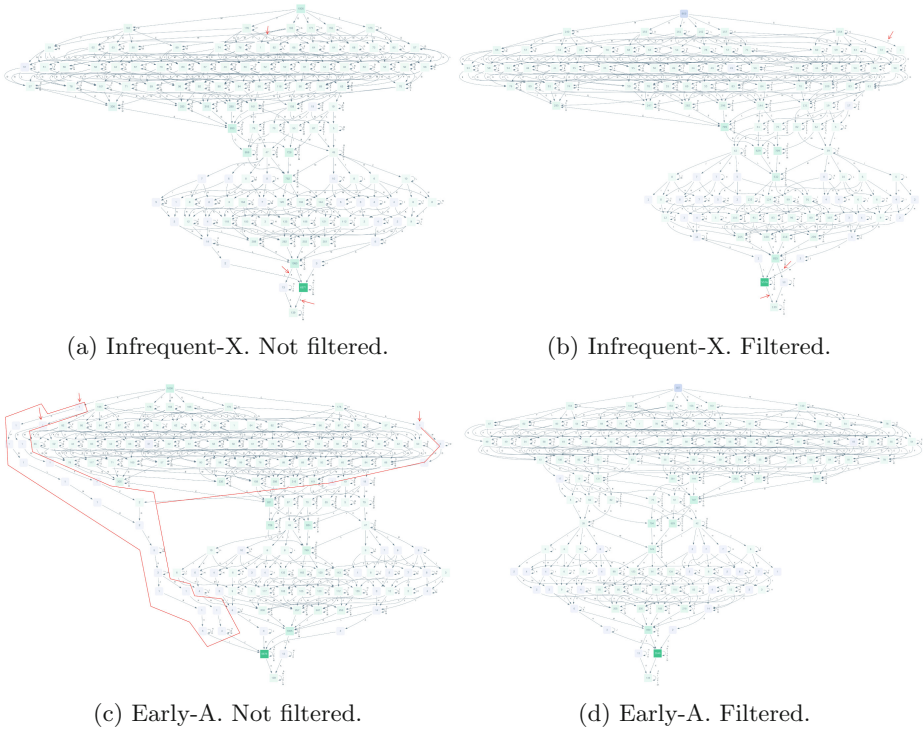


Fig. 4. The effect of filtering on the two synthetic logs. We draw the readers attention to the structure of the graphs, as opposed to the individual nodes and edges. The infrequent X’s and early A’s have been marked with red arrows, as well as the large area of deviant behaviour resulting from the early A’s being marked with red lines. (Color figure online)

4 Experiment 2: Process Discovery Contest 2020

Next we tested the noise filtering algorithm on the PDC 2020 logs, which initially gave rise to our desire to perform apriori noise filtering.

Data-Set

The data set for the contest consists of 192 training logs with the naming convention PDC_2020_ABCDEFG, where each letter corresponds to a single pattern to occur, with the legend seen in Table 1. Most importantly, the last bit is 0 for no noise added, and 1 for approximately 20% noise added. Furthermore, for each training log, there is a corresponding test log, with about 50% noise, and a ground truth log.

Table 1. Legend for the training logs in the PDC 2020 data set, as seen in the README in the data set.

A	Dependent tasks, also known as long-term dependencies. Possible values are 0 for No and 1 for Yes. If Yes then all transitions that bypass the dependent tasks are disabled
B	Loops. Possible values are 0 for No, 1 for Simple, and 2 for Complex. If No, then all transitions that start a loop are disabled. If Simple, then all transitions that are a shortcut between the loop and the main flow are disabled
C	OR constructs. Possible values are 0 for No and 1 for Yes. If No, then all transitions that only take some inputs for an OR-join and all transitions that generate only some outputs for an OR-split are disabled
D	Routing constructs, also known as invisible tasks. Possible values are 0 for No and 1 for Yes. If Yes, then some transitions are made invisible
E	Optional tasks. Possible values are 0 for No and 1 for Yes. If Yes, then some invisible transitions are added to allow skipping of some (visible) transitions
F	Duplicate tasks, also known as recurrent activities. Possible values are 0 for No and 1 for Yes. If Yes, then some transitions are relabeled to existing labels
G	Noise. Possible values are 0 for No and 1 for Yes. If Yes, then noise is introduced in approx. 1 out of 5 traces. Noise is introduced by deleting one event (40%), moving one event in the trace (20%), or copying one event in the trace (40%)

Results

The metric used for evaluating performance on this data-set will be a harmonic mean of true positive rate, and true negative rate, as this was the metric of the PDC2020 automated discovery contest. We will denote this metric Harmonic Accuracy, as seen in (6), with tp , fp , fn , and tn denoting the *true positives*, *false positive*, *false negative*, and *true negatives* of the confusion matrix respectively.

$$\begin{aligned}
 \text{TPR} &= \frac{tp}{tp + fn} \\
 \text{TNR} &= \frac{tn}{tn + fp} \\
 \text{Harmonic Accuracy} &= 2 \cdot \frac{\text{TPR} \cdot \text{TNR}}{\text{TPR} + \text{TNR}}
 \end{aligned} \tag{6}$$

The results can be produced by running the DisCoveR miner on each training log, computing a model for each one. The corresponding test log is then run on this model producing a classified log, in which each trace is classified as either positive (the model accepts the trace) or negative (the model rejects the trace). These classified logs are then held up against the ground truth logs resulting in a confusion matrix for each log.

Based on these confusion matrices, the harmonic accuracy is computed for each log, and then averaged across all logs to yield a total measure.

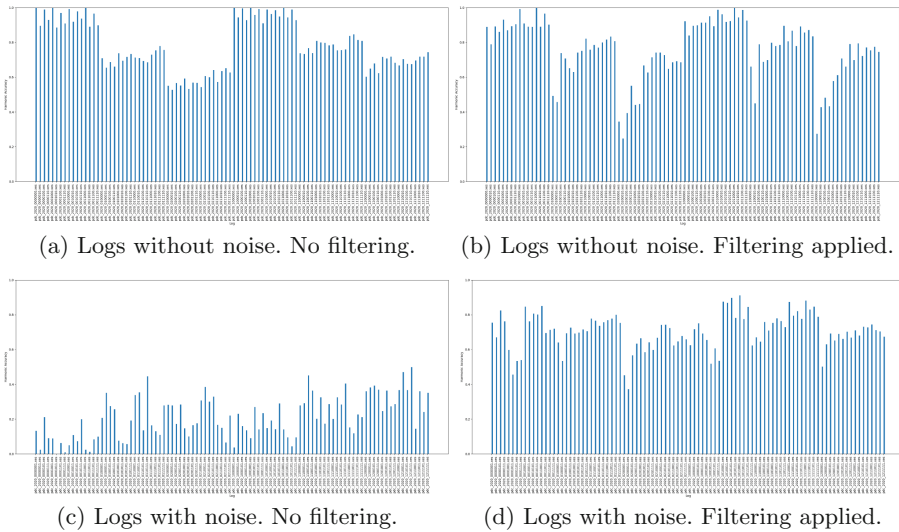


Fig. 5. Results of PDC2020 with and without filtering. Plots the harmonic accuracy for each individual log in the dataset.

With this measure, the DisCover miner scored 50% without filtering and 73% with a priori filtering applied, *i.e.* an increase of 23% points. The scores for the individual logs can be seen in Fig. 5. Here it plainly shows that while the accuracy across logs without noise suffer slightly from applied filtering, the accuracy across logs with noise is severely increased.

5 Conclusion

In this paper we have implemented and tested an algorithm for *a priori* noise filtering [9], for the purpose of improving the accuracy of declarative process discovery. While the algorithm implements both fine- and coarse-grained filtering, it has been parameter optimised to only perform coarse-grained filtering. It was optimised for the PDC2020 data-set, but also improved the accuracy in the currently ongoing PDC2021 contest, and as such we believe the parameters found might prove generally useful.

The parameter optimised algorithm has been applied to synthetic logs, where we showcase examples of how infrequent behaviour in a declarative process is not filtered, while noise in the same log is filtered. Furthermore, we have tested the optimised algorithm on the PDC2020 data-set with the DisCover miner [1], where applied filtering raised the score from 50% to 73%, *i.e.* an increase of 23% points.

In future work we will consider additional noise filtering algorithms, combine the approach with additional declarative miners, such as the Declare miner [7] and MINERful [2], and experiment on additional public real-life event logs.

References

1. Back, C.O., Slaats, T., Hildebrandt, T.T., Marquard, M.: Discover: accurate and efficient discovery of declarative process models. *Int. J. Softw. Tools Technol. Transfer* **24**, 563–587 (2021)
2. Ciccio, C.D., Mecella, M.: On the discovery of declarative control flows for artful processes. *ACM Trans. Manag. Inf. Syst.* **5**(4), 24:1–24:37 (2015). <https://doi.org/10.1145/2629447>. <http://doi.acm.org/10.1145/2629447>
3. Conforti, R., Rosa, M.L., ter Hofstede, A.H.M.: Filtering out infrequent behavior from business process event logs. *IEEE Trans. Knowl. Data Eng.* **29**(2), 300–314 (2017). <https://doi.org/10.1109/TKDE.2016.2614680>
4. Debois, S., Hildebrandt, T.T., Marquard, M., Slaats, T.: The DCR graphs process portal. In: *Proceedings of the BPM Demo Track 2016. CEUR Workshop Proceedings*, vol. 1789, pp. 7–11. CEUR-WS.org (2016)
5. Leemans, S.J., Poppe, E., Wynn, M.T.: Directly follows-based process mining: exploration a case study. In: *2019 International Conference on Process Mining (ICPM)*, pp. 25–32 (2019). <https://doi.org/10.1109/ICPM.2019.00015>
6. Leemans, S., Fahland, D., van der Aalst, W.: Process and deviation exploration with inductive visual miner. In: Limonad, L., Weber, B. (eds.) *BPM Demo Sessions 2014*, pp. 46–50. CEUR Workshop Proceedings, CEUR-WS.org (2014)
7. Maggi, F.M., Mooij, A.J., van der Aalst, W.M.P.: User-guided discovery of declarative process models. In: *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pp. 192–199 (2011). <https://doi.org/10.1109/CIDM.2011.5949297>
8. Nekrasaite, V., Parli, A.T., Back, C.O., Slaats, T.: Discovering responsibilities with dynamic condition response graphs. In: Giorgini, P., Weber, B. (eds.) *CAISE 2019. LNCS*, vol. 11483, pp. 595–610. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_37
9. Sun, X., Hou, W., Yu, D., Wang, J., Pan, J.: Filtering out noise logs for process modelling based on event dependency. In: *2019 IEEE International Conference on Web Services (ICWS)*, pp. 388–392 (2019). <https://doi.org/10.1109/ICWS.2019.00069>
10. van den Broucke, S.K., De Weerd, J.: Fodina: a robust and flexible heuristic process discovery technique. *Decis. Support Syst.* **100**, 109–118 (2017). <https://doi.org/10.1016/j.dss.2017.04.005>. <https://www.sciencedirect.com/science/article/pii/S0167923617300647>. *Smart Business Process Management*
11. Verbeek, H.M.W., de Carvalho, R.M.: *Log skeletons: a classification approach to process discovery* (2018)
12. Weijters, A., Aalst, W., Medeiros, A.: *Process Mining with the Heuristics Miner-Algorithm*, vol. 166 (2006)

**1st International Workshop on Natural
Language Processing for Business
Process Management (NLP4BPM 2022)**

1st International Workshop on Natural Language Processing for Business Process Management (NLP4BPM 2022)

Natural language can play a variety of roles in the context of business process management and analysis. Among others, it can be used to describe processes in a comprehensible manner, define the meaning of events and activities, and it can provide support for the conduct of process analyses themselves, e.g., as an interface for process mining or modeling.

In this context, the goal of the NLP4BPM workshop is to bring together researchers and practitioners to present, discuss, and evaluate how natural language processing (NLP) can be used to establish new or improve existing methods, techniques, tools, and process-aware systems that support the different phases of the BPM life-cycle. Furthermore, we aim to promote an exchange on the advances, challenges and barriers researchers encounter, and establish an environment where collaborations can naturally emerge.

In this first edition of the workshop, we received a total of six submissions, consisting of three regular, one idea, and two resource papers. The submissions were reviewed by at least three members of the Program Committee. From these submissions, the top three were accepted for the workshop: two regular papers and one resource paper. The three papers were presented in-person at the BPM Conference in Münster, Germany, attracting a large audience. After this presentation session, we also hosted a discussion panel on the challenges and opportunities of NLP in BPM, with Chiara Ghidini, Jan Mendling, and Hajo Reijers as panelists. Below, we briefly summarize the three accepted papers.

Stein Dani et al. present an approach for the extraction of event logs from databases. This task is typically a manual and highly time-consuming task, which often represents a hurdle for the application of process mining altogether. Their approach takes a new angle to tackle this by using an existing process model as a starting point. They automatically identify to which database tables the activities of the considered process model relate by using NLP techniques. Based on the resulting mapping, an event log can then be extracted in an automated fashion. The results of the evaluation show that the approach has the potential to successfully support event log extraction based on matching.

Cabrera et al. employ NLP in the context of predictive process monitoring (PPM). As process context can add valuable information to a predictive model, recent PPM techniques often incorporate it to improve process predictions. However, techniques so far barely consider unstructured textual attributes for this, such as process-related comment fields, emails, or documents. Cabrera et al. bridge this gap by proposing a text-aware PPM technique using contextualized word embeddings to predict the next activity and the next timestamp of running process instances. An experimental evaluation with a text-enriched real-life event log shows that the technique presented can

outperform text-aware PPM approaches relying on non-contextualized word embeddings in terms of predictive performance.

Bellan et al. present PET, a novel resource to support researchers that target the task of process model extraction from textual descriptions. Various approaches have been developed for this challenging task, yet there is a lack of gold-standard corpora of annotated textual process descriptions to be used for their evaluation. Due to this, it is currently hard to compare the results obtained by extraction approaches in an objective manner, whereas the lack of annotated texts also prevents the application of data-driven information extraction methodologies, typical of the NLP field. The PET dataset bridges this gap as the first corpus of textual process descriptions annotated with activities, gateways, actors, and flow information, which is complemented with various baselines to benchmark the difficulty and challenges of business process extraction from text.

The organizers wish to thank all the people who submitted papers to the NLP4BPM 2022 workshop, the many participants creating fruitful discussions, the panelists, and the NLP4BPM Program Committee members for their valuable work in reviewing the submissions. We are looking forward to future editions of the NLP4BPM workshop.

October 2022

Han van der Aa
Manuel Resinas
Adela del Río-Ortega
Henrik Leopold

Organization

Organizing Committee

Han van der Aa	University of Mannheim, Germany
Manuel Resinas	University of Seville, Spain
Adela del Río-Ortega	University of Seville, Spain
Henrik Leopold	Kühne Logistics University, Germany

Program Committee

Patrizio Bellan	Fondazione Bruno Kessler, Italy
Jordi Cabot	Open University of Catalonia, Spain
Josep Carmona	Universitat Politècnica de Catalunya, Spain
Fabiano Dalpiaz	Utrecht University, The Netherlands
Mauro Dragoni	Fondazione Bruno Kessler, Italy
Walid Gaaloul	Télécom SudParis, France
Chiara Ghidini	Fondazione Bruno Kessler, Italy
Daniela Grigori	University Paris-Dauphine, France
Wolfgang Kratsch	FIM, Germany
Hugo A. López	University of Copenhagen, Denmark
Fabrizio Maria Maggi	Free University of Bozen-Bolzano, Italy
Jan Mendling	Humboldt-Universität zu Berlin, Germany
Lluís Padró	Universitat Politècnica de Catalunya, Spain
Adrian Rebmann	University of Mannheim, Germany
Hajo A. Reijers	Utrecht University, The Netherlands
Lucinéia Heloisa Thom	Federal University of Rio Grande do Sul, Brazil
Karolin Winter	Technical University of Munich, Germany



Text-Aware Predictive Process Monitoring with Contextualized Word Embeddings

Lena Cabrera^{1(✉)}, Sven Weinzierl^{2(✉)}, Sandra Zilker^{2(✉)}, and Martin Matzner²

¹ Maastricht University, Paul-Henri Spaaklaan 1, Maastricht, The Netherlands
l.cabreraperez@student.maastrichtuniversity.nl

² FAU Erlangen-Nürnberg, Fürther Straße 248, Nürnberg, Germany
{sven.weinzierl,sandra.zilker,martin.matzner}@fau.de

Abstract. Predictive process monitoring (PPM) is the discipline of exploiting event logs of business processes to construct predictive models for anticipating different properties of running business processes. The event logs used contain control flow information of past process executions and, often, additional information about the context in which a process ran. As the process context can add valuable information to a predictive model, recent PPM techniques often incorporate it to improve process predictions. While most techniques incorporate context information as well-structured numerical and categorical context features, only a few utilize unstructured text from process-related comment fields, emails, or documents. The few existing text-aware PPM approaches are limited in capturing semantic information, as different meanings of the same word occurring in different contexts, i.e., sentences, are ignored. This paper addresses this limitation by proposing a text-aware PPM technique using contextualized word embeddings to predict the next activity and the next timestamp of running process instances. An experimental evaluation with a text-enriched real-life event log shows that our technique can outperform text-aware PPM approaches relying on non-contextualized word embeddings in terms of predictive performance.

Keywords: Business process management · Predictive process monitoring · Machine learning · Deep learning · Natural language processing

1 Introduction

Nowadays, ever-changing customer needs and rapid technological progress force organizations to continuously adapt their business processes [14]. Timely anticipation of incremental or radical changes to business processes required in the future is therefore critical to the success of organizations [14]. Predictive process monitoring (PPM) provides a set of techniques developed to predict how ongoing process executions will unfold up to their completion [4]. Predicting properties such as future process behavior (e.g., next activities) or a process outcome

(e.g., a process-related performance indicator) enables process stakeholders to act proactively and take corrective actions in fast-changing environments [9].

As a data-driven business process management approach, PPM exploits event logs of process executions recorded by process-aware information systems to construct predictive models. At the core of recent PPM techniques are often machine learning (ML) algorithms that automatically discover underlying structures in data and capture those within predictive models [1]. Event logs typically contain control flow information of past process executions, attributed with a process identifier and a series of events, each associated with an activity and a timestamp [12]. Existing PPM techniques often extend the control flow perspective of event data by considering additional recorded information about the context in which a process is running [9]. Whenever the process context holds information crucial to the prediction task, purely control flow-oriented approaches are limited in delivering accurate predictions [12]. Thus, current PPM techniques typically utilize context information to improve process predictions [9].

While most techniques consider well-structured numerical and categorical context features, only a few techniques utilize unstructured text from different process-related sources such as comment fields, emails, or documents; despite existing evidence of text adding valuable information to predictive models [12,17]. The few existing text-aware PPM solutions transform the text into vector representations that can be processed by ML algorithms using traditional vectorization approaches (e.g., Bag of words (BoW) [12] or Doc2Vec [12,17]). These approaches are limited in capturing semantic information as polysemous words, which have different meanings depending on the context of the sentence in which they occur, are always represented by the same vector representation [6]. Thus, a word's contextual meaning is ignored.

Natural language processing (NLP) research has addressed the problem of ambiguous words through contextualized word embeddings pre-trained on large corpora [13]. Contextualized word embeddings capture the context in which a word occurs by learning sentence-level semantics, that is, they consider all words comprising a sentence when generating a word's vector representation [13]. However, to the best of the authors' knowledge, no attempt has yet been made to leverage contextual word representations in PPM. Against this background, the contribution of our research is threefold:

1. We propose a text-aware PPM technique using contextualized word embeddings to predict the next activity and the next timestamp of running process instances.
2. We show that the technique leads to improvements in predictive performance regarding both prediction tasks.
3. We evaluate our approach on a real-world event log.

The paper is structured as follows: Sect. 2 introduces preliminaries of PPM in general, presents further concepts relevant for our work, and reviews related work. Section 3 describes the developed text-aware PPM technique. In Sect. 4, the procedure of the experimental evaluation is described. Section 5 presents the results of this work. Section 6 describes limitations and future research directions.

2 Background and Related Work

2.1 Preliminaries

PPM techniques build predictive models from event log data. An event log is structured into processes instances, called cases, where a case consists of a series of events.

Definition 1 (Event, Case, Event Log). *An event is a tuple $e = (c, a, ts, d_1, \dots, d_n)$, where c is the process instance id or case id, a is the activity, ts is the timestamp, and d_n is the n^{th} context feature assigned to the event e . A case is a non-empty sequence $\sigma = \langle e_1, \dots, e_{|\sigma|} \rangle$ of events such that $\forall i, j \in \{1, \dots, |\sigma|\}$ $e_i.c = e_j.c$ with $i > j$, where $.$ denotes the process instance id c of the event e_i or e_j . A case referring to a process instance can also be represented as a sequence of vectors $\langle x_1, \dots, x_{|\sigma|} \rangle$, where $x \in \mathbb{R}^m$ is a vector with size m . A vector can store all event information or a part of it (e.g., information belonging to the event’s activity and its n^{th} context feature). An event log \mathcal{L} is a set of cases $\{\sigma_1, \dots, \sigma_{|\mathcal{L}|}\}$.*

Definition 2 (Prefix, Label). *A prefix is a non-empty sub-sequence of a case $\sigma = \langle x_1, \dots, x_k, \dots, x_{|\sigma|} \rangle$ with a length k . It is defined as $f_{pre}^{(k)}(\sigma) = \langle x_1, \dots, x_k \rangle$, with $0 < k < |\sigma|$. For instance, possible prefixes for $\sigma_1 = \langle x_1, x_2, x_3 \rangle$ are $\langle x_1 \rangle$ or $\langle x_1, x_2 \rangle$. A label is an annotation for a prefix (i.e., the next activity or the next timestamp) of a case $\sigma = \langle x_1, \dots, x_k, \dots, x_{|\sigma|} \rangle$ with a length k . It is defined as $f_l^{(k)}(\sigma) = a_{k+1}$ or $f_l^{(k)}(\sigma) = ts_{k+1}$, with $0 < k < |\sigma|$, where a_{k+1} and ts_{k+1} include features describing the activity and the timestamp of the next event x_{k+1} , respectively. For instance, possible labels for $\sigma_1 = \langle x_1, x_2, x_3 \rangle$ are a_2 or a_3 only storing information of the next events’ activities, or ts_2 or ts_3 specifying the next events’ timestamps.*

2.2 Contextualized Word Embeddings

Representing words and documents as mathematical entities that can be read, reasoned, and manipulated by computational models is a fundamental challenge in NLP [7]. Current leading approaches represent discrete words as fixed-length vectors in a continuous real-valued space, where similarities in the vector space correlate with semantic similarities between words [7]. Typically, these approaches rely on language modeling, where a language model (LM) is a probability distribution over sequences of words that can be used to predict the next word given a number of previous words (i.e., a word’s “left” context) [8].

While distributional word vectors are widely used in modern NLP systems, they only consider a single global representation for each word, ignoring different contexts [8]. In contrast, *contextualized* word embeddings move beyond fixed word embeddings in that each word or token (i.e., a sub-word) is associated with a representation that is a function of the entire sentence it occurs in [13]. First contextualized word vectors were learned from the internal states of a deep bidirectional recurrent neural network (RNN) considering preceding and succeeding words (i.e., a word’s “left” and “right” context). However, more recent

approaches have focused on the Transformer [19], an attention-based encoder-decoder architecture using multi-headed self-attention instead of recurrent layers, to identify the context that confers meaning to each element in a sequence. Arguably, one of the most influential models based on the Transformer is Bidirectional Encoder Representations from Transformers (BERT) [3].

BERT *pre-training* involves two unsupervised tasks: masked language modeling (MLM) and next sentence prediction (NSP). MLM is done to prevent bidirectional conditioning where the model could trivially predict the target word in a multi-layered context. Moreover, to learn sentence relationships, BERT is pre-trained for a binarized NSP task, labeling two input sentences as actual consecutive or random sentences from the corpus.

For BERT *fine-tuning*, the model is first initialized with pre-trained parameters which are fine-tuned using labeled data from another supervised learning task. Compared to pre-training, fine-tuning is relatively inexpensive. However, fine-tuning is not the only way to use BERT. Similar to other models, one can use a pre-trained BERT model to create contextualized word embeddings and feed these embeddings to separate existing models. Devlin et al. [3] show that this process achieves results not far behind fine-tuning BERT.

2.3 Related Work

The broad body of the past work on the next activity or the next timestamp prediction resorts to deep learning (DL) approaches using deep neural network (DNN) architectures. A popular choice of architecture are RNN with long short-term memory (LSTM) cells. As one of the first, Evermann et al. [5] suggest an LSTM architecture for predicting next activities utilizing an embedding layer to encode events. Tax et al. [16] propose a multi-task learning LSTM to predict not only the next activity but also the next timestamp. Camargo et al. [2] propose a multi-task learning LSTM for predicting next activities, timestamps, and related resources.

Regarding context-aware PPM, most techniques consider well-structured categorical and numerical context features (e.g., [18,21]). With respect to textual context data, only a few document- or text-aware PPM solutions have been proposed so far. Weinzierl et al. [20] developed a concept for a document-aware business process prediction technique. They consider context-aware business process prediction techniques coupled with automated information extraction. Yeshchenko et al. [22] present a novel approach to leverage context external to the business processes, namely the sentiments of online news, for the task of remaining time prediction. Teinemaa et al. [17] propose a PPM framework for binary outcome prediction, exploiting text from emails and comments. The authors derive word representations from text documents using four different vectorization techniques, namely n-grams, latent Dirichlet allocation (LDA), Naïve Bayes, and Doc2Vec. As an underlying classifier, they used a random forest. Pegoraro et al. [12] developed a text-aware multi-task process prediction model based on LSTMs. In addition to n-gram, LDA, and Doc2Vec, the authors investigate the BoW model.

3 Text-Aware Process Prediction with BERT

Our proposed technique utilizes BERT to create contextualized word embeddings for text-enriched event logs as it is one of the most important language models existing in the field of NLP. We call our technique TAPPBERT, short for text-aware process prediction with BERT. Figure 1 provides an overview of TAPPBERT. The figure comprises three parts: (a) depicts an excerpt of the raw *event log* used in our experiments, (b) illustrates the *preprocessing* performed to transform the event log into feature inputs for the *prediction model* representing a neural network, and (c) shows its architecture.

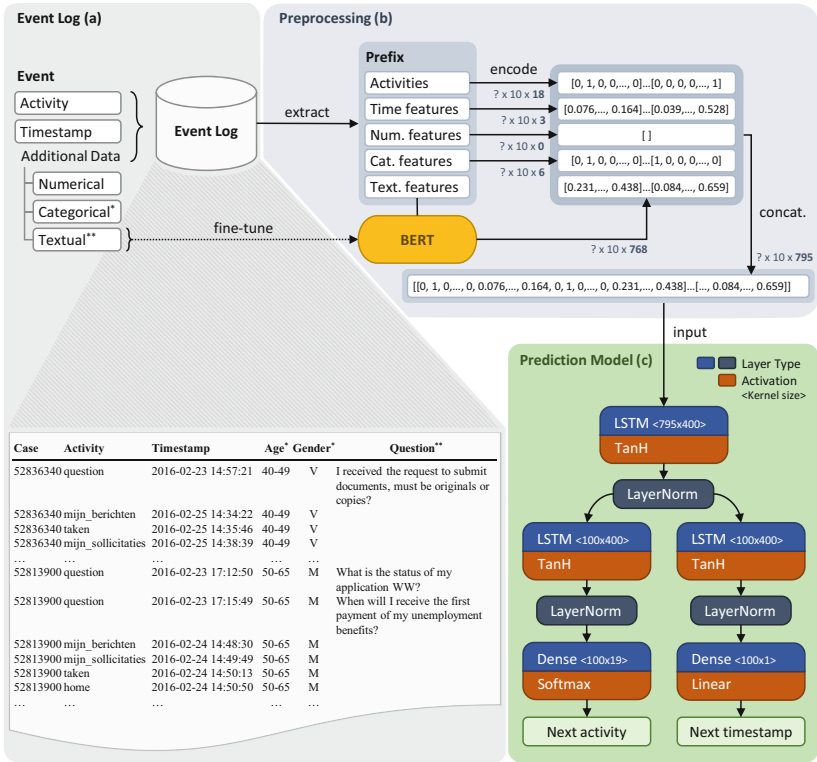


Fig. 1. The proposed technique TAPPBERT.

In the following, we focus on the two components (b) and (c) by describing the illustrated preprocessing steps and then providing a description of the neural network architecture of the prediction model and its implementation. The depicted event log (a) is described in detail in Sect. 4.1.

3.1 Preprocessing

The prediction model of TAPPBERT processes sequences of vectors representing prefixes from the event log. More specifically, each input is a 3-dimensional (3D) prefix tensor I with a shape $P \times E \times F$. P specifies the number of prefixes the model is trained on before its internal parameters are updated (also known as batch size). E is equal to the number of events in the longest case in the event log. F represents a 1D event feature vector comprising concatenated control flow and process context features of a single event. TAPPBERT can utilize numerical, categorical, and textual event features.

Given an event $e = (c, a, ts, d_1, \dots, d_n)$, its activity a is represented by a one-hot encoded vector. The timestamp ts of the event is used to compute time features that capture time-related correlations within a prefix. We convert each timestamp into three time features t_i that are min-max normalized according to $f_{\text{mm}}(t_i) = \frac{t_i - \min(t_i)}{\max(t_i) - \min(t_i)}$. If t_i is not bounded conceptually, the minimum and maximum value of t_i , occurring in an event log, are used for scaling.

For each event, we obtain a time feature vector $\hat{t} = (f_{\text{mm}}(t_1), f_{\text{mm}}(t_2), f_{\text{mm}}(t_3))$, where t_1 is the time passed since the previous event, t_2 is the time within the day (since midnight), and t_3 is the time within the week (since Monday) in seconds.

Regarding context, every feature d_i of e is encoded into a vector using different encoding techniques for numerical, categorical, and textual features. Numerical features are min-max normalized, categorical features are one-hot encoded, and textual features are encoded into a fixed-length vector using BERT. The concatenation of all event features — that is, the one-hot encoded activity vector, the time vector, one-hot encoded categorical context vectors, and the text vector — for all events of prefixes in a batch is I .

At the heart of TAPPBERT is the language model BERT, generating contextualized word embeddings for textual event features. In preparation for encoding text with BERT, we preprocess it with a number of normalization steps, namely conversion to lower case, tokenization, lemmatization, and stop word removal. The preprocessed text is then fed to BERT. We tested different versions of BERT and compared their predictive performance during our experimental evaluation.

Pre-trained BERT. First, we used the pre-trained *BERT base model (uncased)*¹. This model consists of 12 Transformer encoder layers with 12 self-attention heads and a hidden size of 768. The model was trained on the BookCorpus (800M words) and English Wikipedia (2,500M words). In order to be able to generate word embeddings with BERT, we added special tokens to the start and end of texts, namely a [CLS] token for classification and [SEP] to separate sentences. During pre-training, the model uses these special tokens to learn MLM and NSP. The [CLS] token plays an important role since “the final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks” [3]. TAPPBERT uses the [CLS] token representation of

¹ <https://huggingface.co/bert-base-uncased>.

the last layer of the model as the contextualized representation of textual event features.

Pre-trained and fine-tuned BERT. Second, we used *BERT base for Sequence Classification*². This model is the above-described BERT base model (uncased) with an additional linear layer on top of the pooled output to perform sequence classification (e.g., next activity prediction) or regression (e.g., next timestamp prediction³). We fine-tuned this pre-trained BERT base model on both the next activity (BERT+Act) and the next timestamp (BERT+Time) prediction task. Moreover, we experimented with concatenating the word embeddings generated from both of these fine-tuned BERT models (BERT+Act+Time). With regard to tuning the hyperparameters of the model, the authors of the paper, initially proposing BERT, state that the optimal hyperparameter values of BERT are task-specific [3]. However, they state that a range of possible values for batch size, learning rate, and number of epochs works well across various tasks. Acting upon their recommended ranges for hyperparameters, we performed a grid search to identify the optimal hyperparameters for fine-tuning, including 2, 4, 8, and 16 fine-tuning epochs, a learning rate of 2e-5 and 5e-5, and a batch size of 32.

BERT Trained “from Scratch”. Third, we trained BERT from scratch (*Bert base for Pre-training*⁴) using the text from the *question*-feature from the *Customer Journey* event log used during experiments (see Sect. 4.1 for further details). When training BERT “from scratch”, BERT is trained right away on our data set with a random initialization. Thus, subsequent fine-tuning of BERT toward our data set becomes obsolete. Also, the vector size can be varied. In our experiments, we tested a vector size of 36 and 768.

3.2 Prediction Model Architecture

The architecture of the prediction model used in TAPPBERT originates from the multi-task LSTM neural network proposed by Tax et al. [16] which simultaneously predicts the next activity and the next timestamp (see Footnote 3). It consists of an input layer and a shared LSTM layer followed by a specialized LSTM layer and a fully connected output layer, one for each target, respectively. We apply layer normalization after each LSTM layer to stabilize the learning process and apply dropout during training to avoid overfitting. The prediction

² https://huggingface.co/docs/transformers/master/en/model_doc/bert#transformers.BertForSequenceClassification.

³ Strictly speaking, the model learns to predict $f_{mm}(t_1)$ given the timestamp ts_k of the last event in the prefix and the next timestamp ts_{k+1} in the case. Therefore, the next timestamp ts_{k+1} can be calculated by adding t_1 to the timestamp ts_k of the last event in the prefix.

⁴ https://huggingface.co/docs/transformers/master/en/model_doc/bert#transformers.BertForPreTraining.

model receives as input the preprocessed prefixes I to learn to predict the next activity and the next timestamp following each prefix.

4 Evaluation

TAPPBERT is evaluated through a comparison with a set of baseline techniques using a real-life event log. We implemented all techniques in Python using the DL frameworks TensorFlow and PyTorch. The pre-trained BERT models were taken from the Hugging Face library. The source code and event log used to train and evaluate our models as well as the obtained results are available⁵. We conducted all experiments on a workstation with 12 CPUs, 128 GB RAM, and a single Quadro RTX 6000 GPU. In the following, we detail our experimental setup.

4.1 Data Set

The performance of TAPPBERT and the baselines was evaluated on a publicly available data set. The data set describes customer journeys of the Employee Insurance Agency commissioned by the Dutch Ministry of Social Affairs and Employment⁶. It contains anonymized click and phone call data from the agency’s official website and call center, respectively. The event log contains 55,220 events, of which 18 are distinct activities distributed over 15,001 cases with 1,001 different variants and a maximum case length of 10 events. The maximum time gap between consecutive events is 2 d 23 h 59 min and 59 s. While customer *questions* were used as a textual context feature, additional non-textual categorical context features include *gender* and *age group*. Before normalization of the *questions*, there are 247,010 words with a vocabulary size of 1,203. After normalization, 101,076 words with a vocabulary size of 817 remain. Exemplary instances of the different event features can be seen in Fig. 1 (a). To be able to predict the termination of cases, it is common practice to add an artificial *end* event after each case during preprocessing [15].

4.2 Experimental Setting

We ordered the cases of the event log by the timestamp of the first event in each case without shuffling the data. Based on this sorting of the cases, we split the event log into a training (64%), validation (16%), and test (20%) set of cases which are then transformed into prefixes. Aiming to report a reliable estimate of the model’s predictive performance, we repeated the validation procedure three times and calculated the average and the standard deviation per metric across the three sets of results.

⁵ <https://github.com/fau-is/tappbert>.

⁶ <https://research.tue.nl/en/datasets/bpi-challenge-2016>.

We report four different evaluation metrics, two for each type of prediction target, averaged across three runs. The next activity prediction is a classification task. Therefore, we report the *accuracy*, which measures the proportion of correct classifications in relation to the total number of predictions made. Further, we report the weighted *F1-score*, the weighted harmonic mean of precision and recall. These chosen metrics are well-established in the PPM community [10]. The next timestamp prediction (see Footnote 3) problem is a regression task. First, we report the *mean absolute error (MAE)* as it is a commonly used metric in PPM to evaluate the performance of regression models [11]. However, the MAE does not penalize time gaps of large size between two events present in the *Customer Journey* event log as much as the *mean squared error (MSE)*. Therefore, we also report the MSE to evaluate the performance for the regression task.

We compare our technique with a set of baselines in terms of predictive performance. All baselines share the same model architecture as described in Sect. 3.2. However, they differ in their ability to consider text and their method of encoding textual event log features into vectors. The first baseline does not utilize any text features and is therefore not text-aware. Regarding text-aware baselines, we evaluate techniques using four different text encoding methods as proposed by Pegoraro et al. [12]. These methods include BoW, bag of n-gram (BoNG), Doc2Vec, and LDA with three different vector sizes, respectively.

We trained TAPPBERT and the baselines with a maximum number of 25 epochs and a patience of 10 epochs to optimize categorical cross entropy loss and MAE loss for the next activity and the next timestamp prediction, respectively. We used the NAdam optimizer with a learning rate of 0.001, a batch size of 32, a dropout rate of 0.2, and 100 units in each LSTM layer. Regarding the different versions of BERT introduced before, we only report the results of the ones configured with a batch size of 32, a learning rate of $5e-5$ (with Adam optimizer), and 16 epochs in case of fine-tuning. The results of all experiments conducted, including different parameter settings, are documented in the code repository (see Footnote 5).

5 Results

Table 1 presents the results for TAPPBERT and the baselines. The results show that TAPPBERT (4), where BERT was fine-tuned toward predicting the next activity, outperforms all baselines for both prediction tasks. For predicting next activities, it achieves an average accuracy of 0.4933 and an average F1-score of 0.4387. This is a 0.0037 average accuracy improvement and a 0.0030 average F1-score improvement over the prior state of the art. TAPPBERT (4) outperforms the baseline without text by a substantial margin of 0.2 and 0.37 average improvement in accuracy and F1-score, respectively. For predicting next timestamp, it achieves an average MAE of 0.1773 and an average MSE of 0.2782. On average, these results are lower by a margin of 0.0036 (MAE) and 0.0127 (MSE) compared to the prior state of the art. Regarding the baseline without text, the improvement is even more significant (improvement of 0.014 for MAE and 0.052

for MSE). Other versions of TAPPBERT, for instance fine-tuned toward predicting the next timestamp or trained from scratch, achieved no improvement.

Table 1. Predictive performance of TAPPBERT and the baselines (Average across three runs with standard deviation. MAE and MSE values are in days. Bold font highlights the best results.).

	Language model	Vector size	Next activity		Next timestamp		
			Accuracy	F1-score (w)	MAE	MSE	
LSTM MODEL WITHOUT TEXT							
			0.4733 \pm 0.0009	0.4017 \pm 0.0053	0.1913 \pm 0.0001	0.3302 \pm 0.0002	
LSTM MODEL WITH NON-CONTEXTUALIZED WORD EMBEDDINGS (BASED ON [12])							
Baselines	BoW	50	0.4896 \pm 0.0010	0.4315 \pm 0.0016	0.1898 \pm 0.0001	0.3089 \pm 0.0019	
	BoW	100	0.4886 \pm 0.0008	0.4337 \pm 0.0003	0.1897 \pm 0.0002	0.3069 \pm 0.0021	
	BoW	500	0.4814 \pm 0.0015	0.4357 \pm 0.0027	0.1896 \pm 0.0003	0.3036 \pm 0.0020	
	BoNG	50	0.4871 \pm 0.0005	0.4290 \pm 0.0027	0.1901 \pm 0.0000	0.3111 \pm 0.0014	
	BoNG	100	0.4848 \pm 0.0005	0.4314 \pm 0.0005	0.1899 \pm 0.0004	0.3108 \pm 0.0007	
	BoNG	500	0.4811 \pm 0.0025	0.4309 \pm 0.0019	0.1901 \pm 0.0005	0.3069 \pm 0.0022	
	Doc2Vec	10	0.4815 \pm 0.0009	0.4121 \pm 0.0006	0.1809 \pm 0.0026	0.2909 \pm 0.0025	
	Doc2Vec	20	0.4811 \pm 0.0008	0.4203 \pm 0.0058	0.1907 \pm 0.0001	0.3260 \pm 0.0025	
	Doc2Vec	100	0.4822 \pm 0.0015	0.4207 \pm 0.0003	0.1902 \pm 0.0002	0.3178 \pm 0.0006	
	LDA	10	0.4830 \pm 0.0032	0.4169 \pm 0.0024	0.1906 \pm 0.0003	0.3214 \pm 0.0010	
LDA	20	0.4851 \pm 0.0028	0.4221 \pm 0.0041	0.1899 \pm 0.0002	0.3171 \pm 0.0017		
LDA	100	0.4882 \pm 0.0016	0.4326 \pm 0.0023	0.1901 \pm 0.0002	0.3096 \pm 0.0004		
LSTM MODEL WITH CONTEXTUALIZED WORD EMBEDDINGS (TRAINED FROM SCRATCH)							
TAPPBERT	(1) BERT	36	0.4666 \pm 0.0011	0.3899 \pm 0.0027	0.1914 \pm 0.0001	0.3291 \pm 0.0013	
	(2) BERT	768	0.4855 \pm 0.0007	0.4242 \pm 0.0021	0.1905 \pm 0.0003	0.3204 \pm 0.0009	
	LSTM MODEL WITH CONTEXTUALIZED WORD EMBEDDINGS (PRE-TRAINED)						
	(3) BERT	768	0.4843 \pm 0.0028	0.4279 \pm 0.0022	0.1776 \pm 0.0005	0.2801 \pm 0.0012	
	LSTM MODEL WITH CONTEXTUALIZED WORD EMBEDDINGS (PRE-TRAINED AND FINE-TUNED)						
	(4) BERT+Act	768	0.4933 \pm 0.0015	0.4387 \pm 0.0003	0.1773 \pm 0.0003	0.2782 \pm 0.0008	
(5) BERT+Time	768	0.4792 \pm 0.0036	0.4061 \pm 0.0099	0.1821 \pm 0.0038	0.2952 \pm 0.0014		
(6) BERT+Act+Time	1,536	0.4805 \pm 0.0012	0.4082 \pm 0.0043	0.1793 \pm 0.0001	0.2998 \pm 0.0026		

6 Discussion and Concluding Remarks

We derive three findings from our experimental evaluation. First, our results show that our technique can outperform all used baselines in terms of predictive performance. This indicates that BERT captures more semantic information from the given text than the used baselines. Second, we observe that our technique achieves the best results if we fine-tune BERT toward predicting the next process activity. Training BERT from scratch leads to considerably lower process prediction performance. This indicates that valuable information outside the event log can be utilized by applying transfer learning to improve the process prediction performance. Third, our results show that the selected target, toward which BERT is fine-tuned, has a strong impact on the performance of

the process prediction model. A considerably better performing process prediction model is obtained when fine-tuning BERT toward the next activity, instead of the next timestamp or both targets. We suppose that fine-tuning toward the next activity leads to better prediction results as activities determine the process control flow, that is, the ordering of process steps. In contrast, timestamps only provide additional time-related information about the process steps.

Besides our findings, this work also features two limitations. First, we used one event log in the set of experiments addressing text-aware PPM. This experimental design decision is attributed to the fact that there are only very few text-enriched event logs publicly available. In terms of data, we believe that the impact of contextualized word embeddings would become more apparent when using event logs containing a higher amount of text. Typically, larger text corpora promise more information content valuable to the prediction of future process properties. Second, we believe that the predictive performance of TAPPBERT can be further improved if the hyperparameters, in particular those regarding the fine-tuning of BERT, are determined by performing a more exhaustive hyperparameter search using a Bayesian search.

An avenue for future research is to evaluate TAPPBERT with other text-enriched event logs, containing larger text corpora, to better understand the impact of the amount of text used during fine-tuning of BERT on the process prediction result. Future research can also investigate the effect of word embeddings generated via BERT on the predictive performance of other DNN architectures designed for PPM. Finally, other approaches for creating contextualized word embeddings can be subject of future research in PPM.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
2. Camargo, M., Dumas, M., González-Rojas, O.: Learning accurate LSTM models of business processes. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 286–302. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_19
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 4171–4186. ACL (2019)
4. Di Francescomarino, C., Dumas, M., Federici, M., Ghidini, C., Maggi, F.M., Rizzi, W.: Predictive business process monitoring framework with hyperparameter optimization. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) CAiSE 2016. LNCS, vol. 9694, pp. 361–376. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39696-5_22
5. Evermann, J., Rehse, J.R., Fettke, P.: Predicting process behaviour using deep learning. *Decis. Support Syst.* **100**, 129–140 (2017)
6. Hofmann, V., Pierrehumbert, J., Schütze, H.: Dynamic contextualized word embeddings. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 6970–6984. ACL (2021)

7. Levy, O.: Word representation. In: *The Oxford Handbook of Computational Linguistics*. Oxford University Press (2018)
8. Liu, Q., Kusner, M.J., Blunsom, P.: A survey on contextual embeddings. arXiv preprint [arXiv:2003.07278](https://arxiv.org/abs/2003.07278) (2020)
9. Márquez-Chamorro, A.E., Resinas, M., Ruiz-Cortés, A.: Predictive monitoring of business processes: a survey. *IEEE Trans. Serv. Comput.* **11**(6), 962–977 (2017)
10. Mehdiyev, N., Evermann, J., Fettke, P.: A novel business process prediction model using a deep learning method. *Bus. Inf. Syst. Eng.* **62**(2), 143–157 (2020)
11. Navarin, N., Vincenzi, B., Polato, M., Sperduti, A.: LSTM networks for data-aware remaining time prediction of business process instances. In: *2017 IEEE Symposium Series on Computational Intelligence*, pp. 1–7. IEEE (2017)
12. Pegoraro, M., Uysal, M.S., Georgi, D.B., van der Aalst, W.M.: Text-aware predictive monitoring of business processes. In: *Proceedings of the 24th International Conference on Business Information Systems*, pp. 221–232. TIB Open Publishing (2021)
13. Peters, M.E., et al.: Deep contextualized word representations. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2227–2237. ACL (2018)
14. Poll, R., Polyvyanyy, A., Rosemann, M., Röglinger, M., Rupprecht, L.: Process forecasting: towards proactive business process management. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *BPM 2018*. LNCS, vol. 11080, pp. 496–512. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_29
15. Rama-Maneiro, E., Vidal, J., Lama, M.: Deep learning for predictive business process monitoring: review and benchmark. *IEEE Trans. Serv. Comput.* (2021)
16. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: Dubois, E., Pohl, K. (eds.) *CAiSE 2017*. LNCS, vol. 10253, pp. 477–492. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_30
17. 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
18. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: review and benchmark. *ACM Trans. Knowl. Discov. Data* **13**(2), 1–57 (2019)
19. Vaswani, A., et al.: Attention is all you need. In: *Proceedings of the 30th Conference on Neural Information Processing Systems*, pp. 5998–6008. MIT Press (2017)
20. Weinzierl, S., Revoredo, K.C., Matzner, M.: Predictive business process monitoring with context information from documents. In: *Proceedings of the 27th European Conference on Information Systems*, pp. 1–10. AIS (2019)
21. Weinzierl, S., et al.: An empirical comparison of deep-neural-network architectures for next activity prediction using context-enriched process event logs. arXiv preprint [arXiv:2005.01194](https://arxiv.org/abs/2005.01194) (2020)
22. 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



PET: An Annotated Dataset for Process Extraction from Natural Language Text Tasks

Patrizio Bellan^{1,2(✉)}, Han van der Aa³, Mauro Dragoni¹, Chiara Ghidini¹,
and Simone Paolo Ponzetto³

¹ Fondazione Bruno Kessler, Trento, Italy
pbellan@fbk.eu

² Free University of Bozen-Bolzano, Bolzano, Italy

³ University of Mannheim, Mannheim, Germany

Abstract. Process extraction from text is an important task of process discovery, for which various approaches have been developed in recent years. However, differently from other information extraction tasks, there is a lack of gold-standard corpora of business process descriptions carefully annotated with all the entities and relationships of interest. This paper presents the PET dataset, a first corpus of business process descriptions annotated with activities, gateways, actors, and flow information. We present our new resource, including a variety of baselines to benchmark the difficulty and challenges of business process extraction from text. The PET dataset, annotation guidelines, and inception schema are freely available via huggingface.co/datasets/patriziobellan/PET.

Keywords: Process extraction from text · Business process management · Information extraction · Natural language processing · Dataset · Gold standard

1 Introduction

Recent years have seen a growing interest of the Business Process Management (BPM) community on the task of extracting process models from text [1, 3, 7]. Nonetheless, when investigated in light of modern data driven approaches for information extraction, current work has major limitations [3]. Arguably this is in part due to the limited availability of domain-specific, human annotated, gold-standard data that could be used to train from scratch or fine-tune data-driven methods, and which are essential to enable task-specific comparisons across competing approaches. In fact, the availability of reference gold-standard datasets has the potential to further enable application of NLP techniques in the BPM field, and crucially makes clear what the applicability and limitations of state-of-the-art approaches for the domain of interest are.

With this work we aim to fill this gap and foster bridging of work in information extraction and data-driven BPM by providing a novel dataset of human-annotated processes in a corpus of process descriptions. The contributions of this work are:

1. We provide a new reference corpus, annotation schema, and guidelines for the task of annotating business process models in running text. Our corpus includes annotations for different kinds of extraction levels, such as actors, activities and relations between them. As a basis for the annotated data, we used a collection of 45 textual descriptions, initially used by Friedrich et al. [4].
2. We quantify the difficulty of fundamental information extraction tasks for process model extraction by deploying a variety of baselines on our annotated data, thus providing an initial assessment of the feasibility of process extraction from natural language text.

Our vision builds upon bringing together heterogeneous communities such as NLP and BPM practitioners by defining shared tasks and resources (cf. previous work from [8] at the intersection of NLP and political science).

2 Annotation Schema: Process Model Elements and Relations

In this section, we present the annotation schema we used to annotate textual process descriptions, loosely inspired by the analysis presented in [2]. For brevity, we here focus on the elements and their relations depicted in Fig. 1, whereas we refer the reader to the aforementioned URL for a complete description of the process elements, relations, annotation schema, and employed annotation rules.

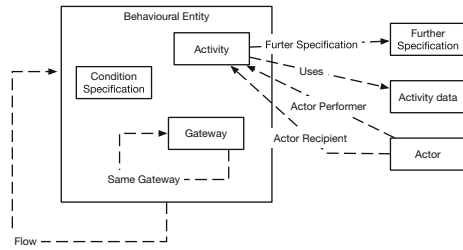


Fig. 1. Annotated elements and relations.

Figure 2 presents an excerpt of a procedural text with the respective process elements annotated. For brevity, we here omit the visualization of relations. An *Activity* represents a single task performed within a process (e.g., sends), while an associated *Activity Data* element captures the object that an activity acts on, (e.g., the questionnaire)¹. These two elements are linked by the *Uses* relation (e.g., sends→the questionnaire). An *Actor* defines the process participant involved in an activity execution (e.g., the customer office). We differentiate between actors that perform/are responsible for an activity using the *Actor Performer* relation (e.g., sends→the customer office); and actors that receive the results of an activity using the *Actor Recipient* relation (e.g., sends→the claimant). A *Further*

¹ Differently from customary BPM terminology, we break down activities to differentiate among an activity’s “action” expression and the object the activity acts on. This eases the annotation task when, e.g., different actions relate to the same object (or vice versa) and because NLP techniques may differ concerning the dealing of verb expressions and noun expressions.

The customer office sends the questionnaire to the claimant by email.
 If the questionnaire is received, the office records the questionnaire and the process end.
 Otherwise, a reminder is sent to the customer.

Fig. 2. The example shows an example of a procedural text fragment with process elements annotated as follows: *Activity*, *Activity Data*, *Actor*, *Further Specification*, *Gateway*, *Condition Specification*.

Specification element captures additional details of an activity that are relevant, but not covered by the other annotations, such as the means or the manner used to perform an activity (e.g., *by email*). An activity is connected to its further specification details by the relation of the same name (e.g., *sends*→*by email*). An activity can have zero or multiple *Uses*, *Actor Performer/Recipient*, or *Further Specification* relations, if described in text.

A *Gateway* element represents a branching point in a process. Currently, we cover *AND gateways* (for parallelism) and *XOR gateways* (for alternative paths). The optional *condition specification* captures a prerequisite that a process execution instance must satisfy to enter a specific branch of a gateway (e.g., *the questionnaire is received*), whereas the *Same Gateway* relation allows us to connect all the parts describing the same gateway, since its description may span over multiple sentences (e.g., *if*↔*otherwise*). Finally, a *Flow* relation defines the process logic enriched with the control-flow information. It connects all the activities, gateways, and condition specification elements in their flow sequence (e.g., *sends*→*if*). *Flow* is thus also used to capture the relationship between a gateway and its condition(s) (e.g., *if*→*the questionnaire is received*).

3 The PET Dataset

We used the annotation schema described in Sect. 2 and the associated annotation guidelines to establish the PET dataset described in this section. This dataset is based on a set of textual descriptions initially used by Friedrich in work on the extraction of process models from text [4]. The primary reason to use this data set as a basis is that the included textual descriptions are well-known within the community, which allows us to provide continuity to the investigation in this research area, as well as to start from a base set of textual documents that are in-line with the type of process narratives considered relevant by the community and used as a basis for the development of existing approaches. However, no publicly available gold-standard annotation of the textual descriptions has been provided so far, a gap that we bridge with the PET dataset.

The PET dataset construction process has been split in five main phases:

1. **Text pre-processing.** As the first operation, we checked the content of each document, we corrected some translation errors, and we tokenized it. This initial check was necessary since the data were never validated. Indeed, several errors have been found and fixed.

Table 1. Inter-annotator agreement for entities and relations.

	Prec.	Recall	F1		Prec.	Recall	F1
Activity	0.96	0.87	0.91	Sequence Flow	1.00	0.67	0.80
Activity Data	0.93	0.73	0.82	Uses	1.00	0.72	0.83
Actor	0.96	0.84	0.89	Actor Performer	1.00	0.74	0.85
Further Specification	0.43	0.33	0.37	Actor Recipient	1.00	0.78	0.87
XOR Gateway	0.88	0.86	0.87	Further Specification	0.64	0.32	0.43
AND Gateway	0.89	0.73	0.80	Same Gateway	0.88	0.72	0.80
Condition Specification	0.86	0.76	0.81				
Overall	0.92	0.79	0.85	Overall	0.98	0.69	0.81

- Text annotation.** Each document has been assigned to three experts that were in charge of identifying all the elements and flows with each document. Annotators were instructed to assign only one label to each annotation. In this phase, we used the Inception tool [6] to support annotators.
- Automatic annotation fixing.** After the second phase, we ran an automatic procedure relying on a rule-based script to automatically fix annotations that were not compliant with the guidelines.
- Agreement computation.** Here, we computed, on the annotation provided by the experts, the agreement scores for each process element and for each relation between process elements pair adopting the methodology proposed in [5]. By following such a methodology, an annotation was considered in agreement among the experts if and only if they capture the same span of words and they assign the same process element tag to the annotation. In the same way, a relation was considered in agreement if and only if the experts strictly annotated the same span of words representing (i) the process element related to the source element; (ii) the process element related to the target element; and, (iii) the relation tag between source and target. The final agreement scores were obtained by averaging the individual scores obtained by the comparison of annotators pairs. Table 1 shows the annotation agreement computed for each process element and each process relation, respectively.
- Reconciliation.** The last phase consisted of the mitigation of disagreements within the annotations provided by the experts. The aim of this phase is to obtain a shared and agreed set of gold standard annotations on each text for both entities and relations. Such entities also enable the generation of the related full-connected process model flow that can be rendered by using, but not limited to, a BPMN diagram. During this last phase, among the 47 documents originally included into the dataset, 2 of them were discarded. These texts were not fully annotated by the annotators since they were not able to completely understand which process elements were actually included in some specific parts of the text, i.e., more than one interpretation would be provided.

The final size of the dataset is 45 textual descriptions of the corresponding process models together with their annotations. The total number of sentences

Table 2. Entity statistics.

	Activity	Activity data	Actor	Further specification	XOR gateway	AND gateway	Condition specification
Absolute count	501	451	439	64	117	8	80
Relative count	30.16%	27.21%	26.43%	3.86%	7.04%	0.48%	4.82%
Per document	11.13	10.04	9.76	1.42	2.60	0.18	1.78
Per sentence	1.20	1.08	1.05	0.15	0.28	0.02	0.19
Average length	1.10	3.49	2.32	5.19	1.26	2.12	6.04
Standard dev	0.48	2.47	1.11	3.40	0.77	1.54	3.04

Table 3. Relation statistics.

	Flows	Uses	Actor performer	Actor recipient	Further specification	Same gateway
Absolute count	674	468	312	164	64	42
Relative count	39.10%	27.15%	18.10%	9.51%	3.71%	2.44%
Count per document	15.31	10.60	6.96	3.64	1.42	0.96
Count per sentence	1.65	1.14	0.75	0.39	0.15	0.10

of the dataset is 413, with an average sentences per document of 9.27 and each sentence has 18.15 words on average. Tables 2 and 3 contains the detailed statistic about process elements and relations respectively.

4 Baseline Results

In this section, we present three baselines we developed to provide preliminary results obtained on the dataset and also to show how the dataset can be used to test different extraction approaches. As described in Sect. 3, there are different type of elements that can be extracted (e.g., activities, actors, relations) and different assumptions that can be made (e.g., the exploitation of gold information or the process of the raw text).

From this perspective, we tested our baselines under three different settings and by using two different families of approaches: Conditional Random Fields (CRF) and Rule-Based (RB):

- **Baseline 1 (B1):** by starting from the raw text (i.e., no information related to process elements or relations has been used), a CRF-based approach has been used for building a model to support the extraction of single entities (e.g., activities, actors).
- **Baseline 2 (B2):** by starting from the existing gold information concerning the annotation of process elements, a RB strategy has been used for detecting relations between entities.
- **Baseline 3 (B3):** this baseline relies on the output of B1 concerning the annotations of process elements. Then, a RB strategy has been used for detecting relations between entities. This baseline simulates a real extraction scenario.

Concerning the CRF approach, we adopted a CRF model encoding data following the IOB2 schema. Results have been obtained by performing a 5-folds

2. Adamo, G., Di Francescomarino, C., Ghidini, C.: Digging into business process meta-models: a first ontological analysis. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) CAiSE 2020. LNCS, vol. 12127, pp. 384–400. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49435-3_24
3. Bellan, P., Dragoni, M., Ghidini, C.: A qualitative analysis of the state of the art in process extraction from text. In: Proceedings of the AIXIA 2020 Discussion Papers Workshop Co-located with AIXIA2020. CEUR Workshop Proceedings, vol. 2776, pp. 19–30. CEUR-WS.org (2020)
4. Friedrich, F.: Automated generation of business process models from natural language input. M. Sc., School of Business and Economics. Humboldt-Universität zu Berlin (2010)
5. Hripcsak, G., Rothschild, A.S.: Technical brief: agreement, the f-measure, and reliability in information retrieval. *J. Am. Med. Inform. Assoc.* **12**(3), 296–298 (2005)
6. Klie, J.C., et al.: The inception platform: machine-assisted and knowledge-oriented interactive annotation. In: Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, pp. 5–9. ACL (2018)
7. Maqbool, B., et al.: A comprehensive investigation of BPMN models generation from textual requirements—techniques, tools and trends. In: Kim, K.J., Baek, N. (eds.) ICISA 2018. LNEE, vol. 514, pp. 543–557. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-1056-0_54
8. Nanni, F., Glavaš, G., Ponzetto, S.P., et al.: Findings from the hackathon on understanding euroscepticism through the lens of textual data. In: LREC. European Language Resources Association (ELRA) (2018)



Supporting Event Log Extraction Based on Matching

Vinicius Stein Dani¹(✉), Henrik Leopold², Jan Martijn E. M. van der Werf¹,
and Hajo A. Reijers¹

¹ Utrecht University, Princetonplein 5, 3584 CC Utrecht, The Netherlands
{v.steindani, j.m.e.m.vanderwerf, h.a.reijers}@uu.nl

² Kühne Logistics University, Großer Grasbrook 17, 20457 Hamburg, Germany
henrik.leopold@the-klu.org

Abstract. Process mining allows organizations to obtain relevant insights into the execution of their processes. However, the starting point of any process mining analysis is an event log, which is typically not readily available in practice. The extraction of event logs from the relevant databases is a manual and highly time-consuming task, and often a hurdle for the application of process mining altogether. Available support for event log extraction comes with different assumptions and requirements and only provides limited automated support. In this paper, we therefore take a novel angle at supporting event log extraction. The core idea of our paper is to use an existing process model as a starting point and automatically identify to which database tables the activities of the considered process model relate to. Based on the resulting mapping, an event log can then be extracted in an automated fashion. We use this paper to define a first approach that is able to identify such a mapping between a process model and a database. We evaluate our approach using three real-world databases and five process models from the purchase-to-pay domain. The results of our evaluation show that our approach has the potential to successfully support event log extraction based on matching.

Keywords: Event log extraction · Natural language processing · Automated matching

1 Introduction

Process mining is used in many different organizations for tasks such as analyzing, improving, and auditing business processes [5, 9, 18]. However, the application of process mining requires an event log [1], which is often not readily available in practice [4]. One of the main reasons is that the information systems supporting the execution of many business processes do not produce event logs that can be used for process mining. As a result, event logs need to be extracted manually by exploring the underlying databases of these information systems. In essence, every activity executed in the context of the business process must be manually related to specific tables in the database. This mapping is then used to extract the event log. This effort for event log extraction is very time-consuming and requires considerable manual work [20]. It, thus, creates a substantial hurdle for the application of process mining in practice [22].

Recognizing this, many researchers have developed techniques to support the extraction of event logs. However, they usually require creating an intermediate data model [16] or using instance data [13]. Furthermore, they do not automatically identify the mapping between the tables of a database and the activities of a considered process because they do not focus on extracting event logs that relate to an already known process flow.

In this paper, we propose a novel approach for supporting event log extraction that takes an existing process model as a starting point. The core idea is to automatically identify to which database tables the activities of a given process model relate to and, based on the resulting mapping, provide an effective alternative for event log extraction. In prior work, the problem of mapping entities from two different representations has been addressed in various contexts. Among others, researchers have proposed techniques for finding mappings between database schemas [14, 17], between ontologies [10, 11], or between process models [21, 23]. Such techniques for automatically deriving mappings between two different representations are commonly referred to as *matchers* [23]. However, to the best of our knowledge, there is no approach available that focuses on identifying a mapping between a database and a process model [20]. To accomplish this, we build on a two layer matching architecture and different notions of similarity.

The remainder of the paper is structured as follows. In Sect. 2, we illustrate the problem of and the challenges related to creating a mapping between database tables and process model activities. In Sect. 3, we describe our proposed approach to support event log extraction based on matching. Section 4 evaluates an implemented proof-of-concept. Finally, in Sect. 5, we discuss related work and in Sect. 6, we conclude this paper.

2 Problem Illustration and Challenges

In this paper, we approach the problem of event log extraction from a matching perspective. More specifically, we aim to develop an approach that automatically identifies a mapping between the tables of a database and the activities of a given process model. To illustrate the problem and the associated challenges, consider the example shown in Fig. 1. It shows a simplified purchase-to-pay process model (extracted from [5]) and a corresponding exemplary database. The goal of our approach is to identify for each activity from a given model to which database table it relates (if any). Formally, such a mapping is a relation over the activities and tables, such that (a, t) maps activity a to table t . In other words, table t contains data of an event for activity a . A *potential mapping* is a candidate mapping that needs to be verified for correctness. Figure 1 depicts several potential mappings. The relations with a checkmark are correct mappings, whereas the mappings marked with a cross are incorrect. Automatically identifying the correct mappings comes with four main challenges:

1. *Large search space*: Given that databases often contain hundreds of tables, the search space for the mapping is typically very big. To illustrate this, consider the example from Fig. 1. The combination of 6 activities and 26 tables already results in over 300

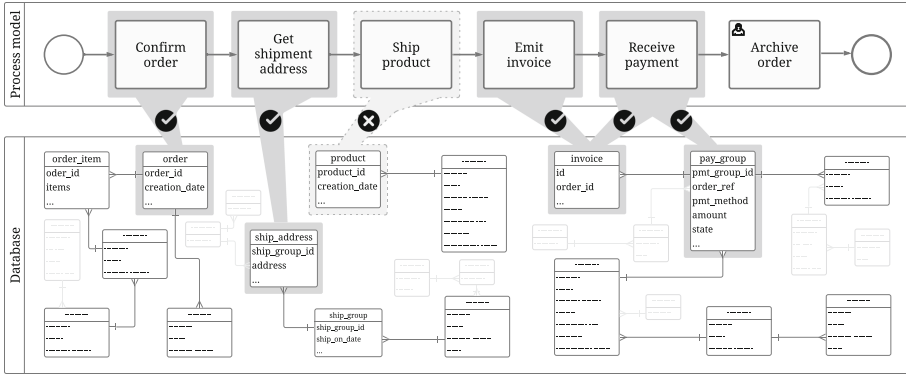


Fig. 1. A process model, a database, and the mappings between them.

million possible mappings. A useful matching technique, therefore, must be able to effectively reduce the search space and precisely recognize which activity-table pairs represent correct mappings.

2. *Granularity differences:* Processes and databases dramatically differ in their level of granularity. While a process model typically only has a handful of activities [6], a database often has hundreds of tables. This causes two related problems. First, this means that a single activity may have multiple corresponding tables. For example, in Fig. 1, the activity “Receive payment” produces a payment entry for the database table “pay_group” while also producing an update of an entry in the table “invoice”. Second, this means that a single table may have multiple corresponding activities. For example, the table “invoice” stores data about a newly created invoice produced by the activity “Emit invoice”. The same table also reflects a payment status updated via the execution of the activity “Receive payment”.
3. *Scope differences:* The scope of the process model and the database rarely overlap to a full extent. As a result, the mapping between process model and database is partial. This means that some activities do not have a correspondence to any table and, the other way around, many tables do not have a correspondence to any activity. For example, in Fig. 1, the activity “Archive order” may be related to a manual status update executed on an external system managed by another department of the organization and, therefore, does not relate to any of the tables of the considered database.
4. *Ambiguous semantics:* Both process models and database tables typically have very short labels. As a result, it is often hard to identify which words from the considered labels carry the important semantics. To illustrate this, consider the activity “Ship product” from Fig. 1. We can see that this activity contains the action “ship” and the object “product”. In Fig. 1 it is, however, incorrectly mapped to the table “product” instead of “ship_group”. The problem is that it is hard to evaluate which term should be used in this context to decide about the mapping since both “ship” and “product” are used in the database tables.

In this work, we make a first attempt to address these challenges. We propose an approach that identifies a set of potential mappings between process model activities and database tables. While this does not provide the user with a final set of correct mappings, the user is provided with a small set of potential mappings. From those, the user can simply select the correct mappings and, hence, no longer needs to look at all possible mappings and identify each mapping manually. We realize that this only represents a first step. We are, nonetheless, convinced that this already dramatically reduces the burden of the process analyst and saves a considerable amount of manual work. In the next section, we introduce our approach on a conceptual level.

3 Mapping Database Tables to Process Activities

In this section, we describe our matching approach to automatically map database tables to process model activities. We first present an overview of the architecture of our matching approach in Sect. 3.1. Then, in Sect. 3.2 and Sect. 3.3, we discuss the main components of our matcher in detail.

3.1 Overview

Figure 2 shows the architecture of our proposed approach. The first module is responsible for *preprocessing* and feeding input data into the matcher. Among others, the preprocessing component parses the input, removes irrelevant tokens (such as punctuation), and turns all strings into lower case. The input data includes a database and a process model. At this point, we expect that both have already been transformed into a textual format and are provided as CSV files. These files contain the table attributes from the database (e.g., tables names, descriptions, and columns with their names and descriptions), and the activity labels from the process model.

Inspired by [7], the *matcher* module consists of two main components: a first- and a second-line matcher (1LM and 2LM), where the 2LM builds on the output of the 1LM. The matcher automatically generates a set of potential mappings. To generate these potential mappings, we leverage natural language processing (NLP) techniques and the available input information. The main intuition behind relying on NLP techniques is that tables and activities with similar names are more likely to be conceptually similar and, therefore, related. In the following sections, we explain the details of the components from the matcher module.

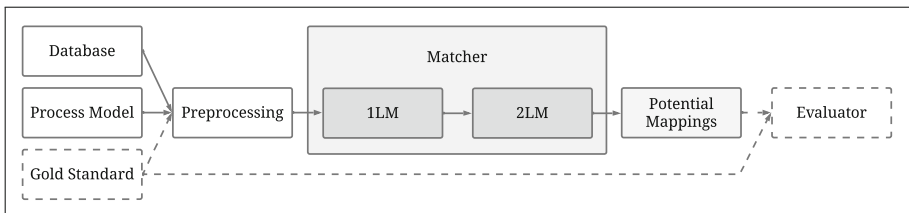


Fig. 2. Architectural overview of our approach.

3.2 First-Line Matcher (1LM)

Our approach starts with analyzing the set of activity labels A of the process model and tables T of the database using different similarity metrics. For each table, we consider all database table attributes, denoted by R . Then, for each activity a and database table attribute t_r , several similarity measures are calculated. This results in a set of similarity matrices M_s , for each similarity measure s .

Table 1 shows a cohort of the similarity matrix $M_{s(A \times R)}$ for the normalized Levenshtein-based similarity measure on a process model with two activities, “*Create order*” and “*Create invoice*”, and a database consisting of two tables, “*Order*” and “*Invoice*”. In this example, the table “*Order*” has two columns: “*id*”, and “*creation_date*” and, the table “*Invoice*” has three columns: “*id*”, “*id_order*”, and “*date*”.

3.3 Second-Line Matcher (2LM)

The 2LM derives the set of potential mappings between tables and activities by using as input the similarity matrix M_s generated by the 1LM. Our approach maps exactly one database table t to one activity label a , and the inner workings of the 2LM adheres to the following rationale: First, considering all available similarity scores in M_s (cf., Table 1), the 2LM determines a similarity score to represent a table with respect to each activity. This is performed for each tuple (a, t) . Second, for each activity, it selects one table as a potential mapping considering the similarity score assigned to the table. Many different mechanisms can be implemented to derive the table’s similarity score from its attributes’ similarity scores.

We developed a baseline 2LM inspired by [21], which selects the *Highest raw 1LM-based Scoring Table* as a potential mapping for an activity label. Based on the output of this 2LM for each 1LM similarity matrix, we performed an inductive content analysis with open coding [19]. Recurrent observations from the coding served as a basis for the definition of two new 2LM implementations: one based on *Word Frequency (2LM₂)*, and another based on the *Surface Measure of Overall Table Scores (2LM₃)*. Next, we further explain each implemented 2LM.

Table 1. Similarity matrix generated by the 1LM for the normalized Levenshtein-based similarity algorithm. The closer the similarity score is to 1, the higher the similarity between the two compared objects.

Database tables attributes t_k	Process model activities a	
	Create order $f_s(a, t_k)$	Create invoice $f_s(a, t_k)$
Order	0.590	0.210
id	0.140	0.120
creation_date	0.480	0.440
Invoice	0.210	0.670
id	0.140	0.120
id_order	0.500	0.180
date	0.380	0.330

Highest Raw ILM-Based Scoring Table (2LM₁). Each row in a similarity matrix M_s produced by the ILM represents the similarity scores of an activity and all attributes t_r of all tables $t \in T$. 2LM₁ selects for each activity and table combination the attribute with the highest similarity as *table score*. Then, for each activity, the table with the highest table score is selected as potential mapping.

Word Frequency (2LM₂). This technique multiplies the table attributes similarity score by the number of activity label word repetition within the table attribute. This is done before the *table score* definition and, if there is no word repetition, the similarity score is kept as is. Hence, this matcher derives each of its potential mappings similarly to 2LM₁.

Surface Measure of Overall Table Scores (2LM₃). This technique is inspired by [8], and leverages all similarity scores of a table to build a radar chart, where each similarity score is an axis of the chart. The *table score* $S(a, t)$ is then determined by calculating its surface area, as shown in Eq. 1, where R denotes the set of table attributes.

$$S(a, t) = \sin\left(\frac{\pi}{|R|}\right) \sum_{x \in R} \sum_{y \in R} (M_s(a, t_x) \cdot M_s(a, t_y)) \quad (1)$$

4 Evaluation

In this section, we present a quantitative evaluation of our approach. In Sect. 4.1 and Sect. 4.2, we describe the data and our setup. In Sect. 4.3, we report on the results and provide a discussion in Sect. 4.4.

4.1 Data

The evaluation builds on three inputs: 1) a set of databases, 2) a set of process models, and 3) a gold standard.

Databases. For the evaluation, we used three databases: 1) Odoo (former Open ERP), 2) Magento Commerce, and 3) Oracle ATG Webcommerce. The selected databases cover two scenarios we want to evaluate: databases with and without textual descriptions of the tables and columns. Oracle is accompanied by a textual description, whereas Odoo and Magento are not. Additionally, these databases were selected considering two other factors: 1) they store purchase-to-pay data; and, 2) they are widely used. Table 2 summarizes the overall characteristics of the selected databases.

Process Models. We used five process models of a purchase-to-pay process of different sizes. The set of process models contains one small process model extracted from [5], and four medium-sized process models extracted from the BPM Academic Initiative (BPMAI) repository [24]. The BPMAI models were selected based on the following criteria: 1) it is modelled in English, 2) it contains at least 10 activities, and 3) it relates to a purchase-to-pay process. To make sure the latter is the case, we selected process models containing the business objects “*order*”, “*invoice*”, and “*shipment*”. Table 3 summarizes the overall characteristics of the selected process models.

Table 2. Characteristics of the databases used in the evaluation of our approach.

Characteristic	Odoo	Magento	Oracle
Database			
N ^o of tables	571	358	239
N ^o of columns	6294	3561	1199
N ^o of words	57297	42189	37051
Table			
Avg N ^o of words per table name	2.794	3.502	2.838
Avg N ^o of words per table description	2.356	3.815	14.197
Avg N ^o of words per column name	1.978	2.216	1.952
Avg N ^o of words per column description	1.974	2.311	12.009

Table 3. Characteristics of the process models used in the evaluation of our approach.

Characteristic	PM ₁	PM ₂	PM ₃	PM ₄	PM ₅
Process model					
N ^o of activities	6	10	11	12	14
N ^o of words	13	34	33	34	50
Activity label					
Min N ^o of words	2	1	2	1	2
Max N ^o of words	3	6	5	5	6
Avg N ^o of words	2.166	3.400	3.000	2.833	3.571

Gold Standard. The gold standard G contains the true mappings between the database tables t and the process model activities a . It is a set of relations (a, t) . To evaluate the quality of the output of our approach (i.e., the potential mappings), we compare it to G as we further explain in the next section. We manually compiled G based on prior experience and insights into which tables hold the information related to the considered activities. For activities we did not know the corresponding table, we consulted the documentation of the database. We fine-tuned G based on discussions until consensus.

4.2 Setup

For each combination of database and process model, we generated ten similarity matrices $M_{s(A \times R)}$ via 1LM, one for each similarity algorithm $s \in S$, comprising different string-similarity scoring techniques, such as: edit-based (via Levenshtein, and a normalized Levenshtein-based algorithm), Jaccard, n-gram, and Cosine similarity. Then, we implemented the 2LMs as discussed in Sect. 3.3, and to assess the performance of our approach we use precision, recall, and F1-score. This is in line with evaluations from other matching papers from the BPM domain (see e.g. [21]). To calculate these metrics, we compare the output from our approach with the mappings from the gold standard G .

Given a combination of a process model containing the activities A and a database containing the tables T , we compare the set of mappings between A and T from the

gold standard G with the set of potential mappings P automatically produced by our approach. Based on this comparison, we can identify: 1) the correct mappings (i.e., the true positives TP) via $G \cap P$, 2) the incorrect mappings (i.e., the false positives FP) via $P \setminus TP$, and, 3) the missing mappings (i.e., the false negatives FN) via $G \setminus TP$. Thus, we can calculate precision via $\frac{TP}{TP+FP}$ and recall via $\frac{TP}{TP+FN}$. The F1-score is the harmonic mean between precision and recall.

4.3 Results

Table 4 summarizes the performance results of our approach in terms of precision, recall, and F1-score. For each database, the fourth column of this table presents the number of mappings in the gold standard G . This allows us to compare the number of mappings from G to the amount of correct mappings (TP) generated by each of the 2LMs.

On average the implemented 2LM₃ finds 39% of the correct mappings for the databases with table and column descriptions. The implementations 2LM₂ and 2LM₃, perform similarly for a scenario where the database does not have useful textual descriptions, as shown in Figs. 3d and 3g, for example. The 2LM₃ implementation performs better than 2LM₂ for a scenario where the database has textual descriptions, as shown in Figs. 3f and 3i. In both scenarios, the 2LM₃ performs well when the process model does not have too many similar activity labels, which is the case for the results related to PM₁. All the results presented in this work are based on a 1LM using cosine similarity, which is the similarity algorithm that performed best. Figures 3a to 3i depict,

Table 4. Evaluation summary with Precision, Recall, F1-scores, total true positives (TP), and false positives (FP) for the three different 2LM implementations. The baseline 2LM₁ results are zero for Odoo and Magento because more than one table had the same highest similarity score for each activity and the baseline selects the first table with the highest similarity score as a potential mapping.

DB	PM	A	G	2LM ₁					2LM ₂					2LM ₃				
				P	R	F1	TP	FP	P	R	F1	TP	FP	P	R	F1	TP	FP
Odoo	1	6	7	0.000	0.000	0.000	0	6	0.000	0.000	0.000	0	6	0.167	0.143	0.154	1	5
	2	10	9	0.000	0.000	0.000	0	10	0.100	0.111	0.105	1	9	0.100	0.111	0.105	1	9
	3	11	12	0.000	0.000	0.000	0	11	0.091	0.083	0.087	1	10	0.091	0.083	0.087	1	10
	4	12	6	0.000	0.000	0.000	0	12	0.000	0.000	0.000	0	12	0.000	0.000	0.000	0	12
	5	14	8	0.000	0.000	0.000	0	14	0.000	0.000	0.000	0	14	0.000	0.000	0.000	0	14
Magento	1	6	6	0.000	0.000	0.000	0	6	0.167	0.167	0.167	1	5	0.167	0.167	0.167	1	5
	2	10	5	0.000	0.000	0.000	0	10	0.100	0.167	0.125	1	9	0.000	0.000	0.000	0	10
	3	11	6	0.000	0.000	0.000	0	11	0.182	0.333	0.235	2	9	0.000	0.000	0.000	0	11
	4	12	4	0.000	0.000	0.000	0	12	0.000	0.000	0.000	0	12	0.000	0.000	0.000	0	12
	5	14	7	0.000	0.000	0.000	0	14	0.071	0.143	0.095	1	13	0.071	0.143	0.095	1	13
Oracle	1	6	8	0.000	0.000	0.000	0	6	0.167	0.125	0.143	1	5	0.834	0.625	0.714	5	1
	2	10	8	0.100	0.125	0.112	1	9	0.100	0.125	0.112	1	9	0.300	0.375	0.334	3	7
	3	11	11	0.091	0.091	0.091	1	10	0.273	0.273	0.273	3	8	0.454	0.454	0.454	5	6
	4	12	6	0.167	0.334	0.223	2	10	0.167	0.334	0.223	2	10	0.250	0.500	0.334	3	9
	5	14	8	0.071	0.125	0.091	1	13	0.143	0.250	0.182	2	12	0.357	0.625	0.454	5	9

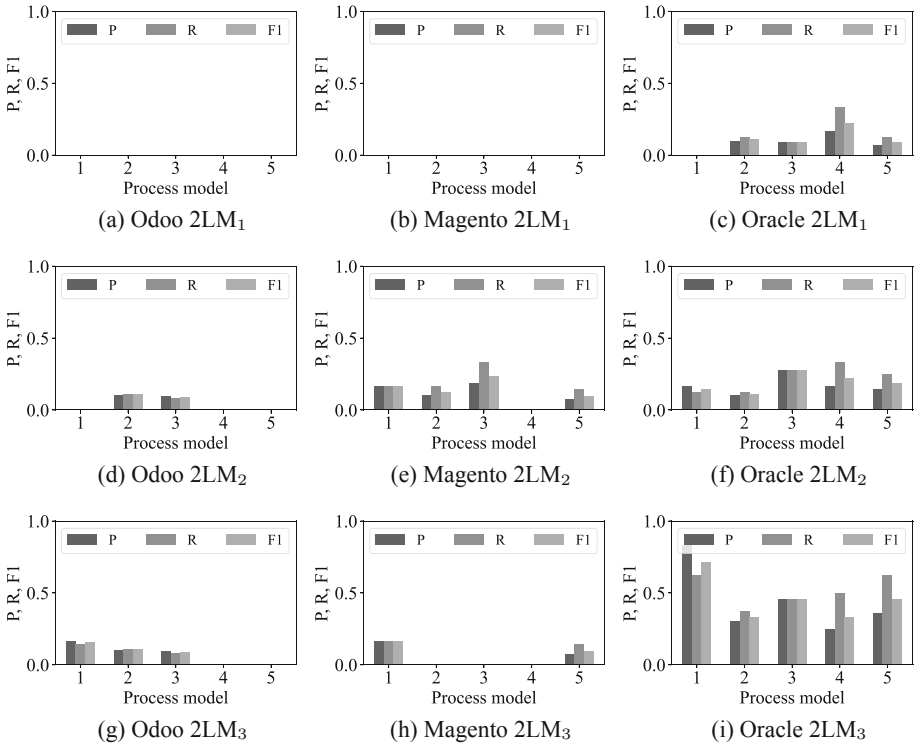


Fig. 3. Evaluation output for three databases and five process models used in this evaluation. The first row of figures shows the output for the baseline 2LM₁, while the second and the third rows show the output for the other two implemented 2LMs. Figures 3a, 3d, and 3g refer to Odoo; Figs. 3b, 3e, and 3h refer to Magento; and, Figs. 3c, 3f, and 3i refer Oracle. On each figure, the vertical axis represents the value of precision, recall, and F1-score, for each of the five process models, shown in the horizontal axis.

respectively, the output of our approach for the three different implemented 2LMs presented in Sect. 4.2. The first column of Fig. 3 presents the output for Odoo, the second for Magento, and the third for Oracle.

In summary, we can state that all 2LM implementations performed better on the scenario where textual descriptions were available for the tables and columns. Moreover, the 2LM₂ and the 2LM₃ improve consistently when compared to the baseline implementation on the Oracle database, which is the database with textual descriptions for both tables and columns. For the databases without textual descriptions, the results deteriorate in general, showing the importance of additional textual information about the objects being mapped. The reason for this results deterioration is that multiple tables end up receiving the same similarity score, driven by similarly named columns throughout different tables.

4.4 Discussion

To generate the $2LM_2$ and the $2LM_3$, we derived improvement opportunities based on recurrent observations acquired via an inductive content analysis with open coding [19] performed over all outputs from the $2LM_1$. By doing so, we avoided optimizing a new $2LM$ to any particular scenario.

The performed content analysis supported the identification of commonalities and differences among all potential mappings (correctly and incorrectly identified mappings) versus the ones that should have been identified, but were not. We made the following key observations: First, the missing mappings (i.e., *FN*) often had repetition of words that were similar to the ones within the activity label, while it was not the case for the incorrect mappings. Second, the incorrect mappings often had multiple table elements with mild similarity scores, while the wrongly selected potential mapping had usually only one slightly higher score, which then misled the baseline mapping derivation. Therefore, the matcher should consider the table attributes scores altogether.

With the current work, we provide a first step towards supporting event log extraction based on a given process model. Our approach is able to identify a set of potential mappings, which then can be processed by a process analyst. While our technique can be further improved, we also provide some insights into how this can be accomplished (cf. $2LM_3$).

5 Related Work

While this paper is the first work on database to process model matching, there are three major research areas that are concerned with matching: schema matching, ontology matching, and process model matching.

Approaches for *schema matching* aim to identify matches between the elements of two different database schemata. The purpose of schema matching techniques include data integration, schema evolution, and maintenance [14, 17]. The matching strategies pursued by these techniques are similar to the ones presented in this paper. For example, in [14], the authors determine the similarity between two database schema elements using attributes, such as names and data types, and combine it with structural similarity. In [17], the authors leverage the results of a variety of basic matchers to determine whether two schema elements match.

Approaches for *ontology matching* are concerned with matching the elements of two ontologies [10, 11]. One of the key use cases for ontology matching is ontology merging, i.e., the combination of two ontologies. The matching strategies are again similar to one presented here. For example, in [10], the authors leverage lexical and structural characteristics of the considered ontologies to determine matching elements.

Approaches for *process model matching* aim to identify correspondences between the activities of two process models [12, 15, 23]. The main use case of process model matching is to detect differences and commonalities between two processes. Available approaches for process model matching exploit textual, structural, and behavioral features of the models. Early work mainly built on simple textual similarity features, such as the Levenshtein distance, and mainly focused on structural features [23]. Later, also semantic similarity measures and behavior were used to identify corresponding activities [12].

This brief review illustrates that existing matching approaches are closely related to our work. There is, however, a key difference: The works above focus on matching entities of the same type. While this does not guarantee that the to-be-matched entities are similar, they are at least comparable. In the setting addressed in this paper, we need to deal with the fact that the entities are very different in nature. A process model, for example, does not come with instance data and a database does not have a clear notion of control-flow or activities. Hence, while we partially build on matching strategies explored in previous work, the conceptual setting of our work differs considerably.

6 Conclusion

In this paper, we presented a new approach to support event log extraction based on matching. The main idea of our approach is to automatically identify the mappings between a database and a process model. Against the background of the challenges associated with this task, we focused on the automated identification of potential mappings in this paper. While this requires process analysts to select the correct mappings, it still saves them from a considerable amount of manual work. To evaluate our approach, we tested it using three different databases and five different process models related to a purchase-to-pay process. We found that textual information is highly important to improve the performance of our approach. At the same time, we also found that more sophisticated mechanisms are required to further improve our approach.

As for future work, we see several directions. First, we plan extend the idea from the syntactic level to a level that incorporates semantic relations as well [2,3] between the activities and tables by, for example, leveraging bidirectional encoder representations from transformers. Second, we aim to take order relations between the database instance data and the process model activities into account. In this way, we can, for instance, exclude candidate matches if the order relations from the process model contradict the timestamps from the associated database tables. Third, we intend to incorporate feedback from humans. By letting the user select which potential mappings are correct, we can leverage a feedback loop to further improve the potential mappings generated by our approach.

Acknowledgements. Part of this research was funded by NWO (Netherlands Organisation for Scientific Research) project number 16672.

References

1. van der Aalst, W.M.P.: *Process Mining: Data Science in Action*. Springer, Heidelberg (2016)
2. Calvanese, D., Kalayci, T.E., Montali, M., Santoso, A.: OBDA for log extraction in process mining. In: Ianni, G., et al. (eds.) *Reasoning Web 2017*. LNCS, vol. 10370, pp. 292–345. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61033-7_9
3. Calvanese, D., Kalayci, T.E., Montali, M., Tinella, S.: Ontology-based data access for extracting event logs from legacy data: the *onprom* tool and methodology. In: Abramowicz, W. (ed.) *BIS 2017*. LNBIP, vol. 288, pp. 220–236. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59336-4_16
4. 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) (2020)

5. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*. Springer, Heidelberg (2018)
6. Figl, K., Mendling, J., Strembeck, M.: The influence of notational deficiencies on process model comprehension. *J. Assoc. Inf. Syst.* **14**, 312–338 (2013)
7. Gal, A.: *Uncertain Schema Matching*, vol. 3. Morgan & Claypool (2011)
8. Jagroep, E., Van der Werf, J.M., Broekman, J., Blom, L., van Vliet, R., Brinkkemper, S.: A resource utilization score for software energy consumption. In: *Proceedings of ICT for Sustainability 2016* (2016)
9. Jans, M., Alles, M., Vasarhelyi, M.: The case for process mining in auditing: sources of value added and areas of application. *Int. J. Account. Inf. Syst.* **14**, 1–20 (2013)
10. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. *Web Semant.* **7**(3), 235–251 (2009)
11. Lambrix, P., Tan, H.: Sambo - a system for aligning and merging biomedical ontologies. *J. Web Semant.* **4**(3), 196–206 (2006)
12. Leopold, H., Niepert, M., Weidlich, M., Mendling, J., Dijkman, R., Stuckenschmidt, H.: Probabilistic optimization of semantic process model matching. In: Barros, A., Gal, A., Kindler, E. (eds.) *BPM 2012. LNCS*, vol. 7481, pp. 319–334. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32885-5_25
13. Li, G., de Murillas, E.G.L., de Carvalho, R.M., van der Aalst, W.M.P.: Extracting object-centric event logs to support process mining on databases. In: Mendling, J., Mouratidis, H. (eds.) *CAiSE 2018. LNBIP*, vol. 317, pp. 182–199. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92901-9_16
14. Madhavan, J., Bernstein, P., Rahm, E.: Generic schema matching with cupid. In: *Proceedings of the 27th VLDB Conference* (2001)
15. Meilicke, C., Leopold, H., Kuss, E., Stuckenschmidt, H., Reijers, H.A.: Overcoming individual process model matcher weaknesses using ensemble matching. *Decis. Support Syst.* **100**, 15–26 (2017)
16. Murillas, E., Reijers, H., Aalst, W.: Connecting databases with process mining: a meta model and toolset. *Softw. Syst. Model.* 231–249 (2016)
17. Nikovski, D., Esenther, A., Ye, X., Shiba, M., Takayama, S.: Matcher composition methods for automatic schema matching. In: Cordeiro, J., Maciaszek, L.A., Filipe, J. (eds.) *ICEIS 2012. LNBIP*, vol. 141, pp. 108–123. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40654-6_7
18. Post, R., et al.: Active anomaly detection for key item selection in process auditing. In: Munoz-Gama, J., Lu, X. (eds.) *ICPM 2021. LNBIP*, vol. 433, pp. 167–179. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98581-3_13
19. Saldaña, J.: *The Coding Manual for Qualitative Researchers*. Sage (2009)
20. Stein Dani, V., et al.: Towards understanding the role of the human in event log extraction. In: Marrella, A., Weber, B. (eds.) *BPM 2021. LNBIP*, vol. 436, pp. 86–98. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94343-1_7
21. van der Aa, H., Leopold, H., Reijers, H.A.: Comparing textual descriptions to process models - the automatic detection of inconsistencies. *Inf. Syst.* **64**, 447–460 (2017)
22. Aalst, W.M.P.: Extracting event data from databases to unleash process mining. In: vom Brocke, J., Schmiedel, T. (eds.) *BPM - Driving Innovation in a Digital World. MP*, pp. 105–128. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-14430-6_8
23. Weidlich, M., Dijkman, R., Mendling, J.: The ICop framework: identification of correspondences between process models. In: Pernici, B. (ed.) *CAiSE 2010. LNCS*, vol. 6051, pp. 483–498. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13094-6_37
24. Weske, M., Decker, G., Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: *Model collection of the bpm academic initiative* (2020)

Author Index

A

Aamer, Heba 132
Amantea, Ilaria Angela 197
Amit, Guy 45
Andree, Kerstin 231
Aringhieri, Roberto 197

B

Bano, Dorina 231
Bein, Leon 231
Bellan, Patrizio 315
Bergmann, Ralph 274
Bertrand, Yannic 63
Braun, Stephan A. 274
Burattin, Andrea 101, 243

C

Cabrera, Lena 303
Carbajales, Sebastian 13
Casaluce, Roberto 243
Chan, Allen 13
Chiaromonte, Francesca 243
Christfort, Axel Kjeld Fjelrad 286

D

de Araujo, Renata Mendes 167
de Classe, Tadeu Moreira 167
De Weerd, Jochen 63
Debois, Søren 286
del-Río-Ortega, Adela 210
der Aalst, Wil van 117
Di Cunzolo, Matteo 197
Di Federico, Gemma 101
Di Francescomarino, Chiara 117, 197
Dragoni, Mauro 315

E

Estrada-Torres, Bedilia 210

F

Fettke, Peter 37
Fonio, Paolo 197
Fournier, Fabiana 45

G

Gallik, Florian 89
Geyer, Tobias 274
Ghidini, Chiara 117, 197, 315
Gomes, Thayná 167
Goossens, Alexandre 25
Grosso, Marco 197
Grüger, Joscha 274
Guastalla, Alberto 197

H

Haarmann, Stephan 231
Holz, Julia 257

K

Khandaker, Faria 13
Kim, Inkyu 149
Kirikkayis, Yusuf 89
König, Maximilian 231
Kourani, Humam 117
Kuhn, Martin 274

L

Lahann, Johannes 37
Leopold, Henrik 322
Limonad, Lior 45
Lopes, Tatiane Neves 167

M

Maes, Ulysse 25
Maleki Shamasbi, Simin 179
Mandel, Caterina 231
Matzner, Martin 303
Montali, Marco 5, 132

P

Pentland, Brian T. 149
 Pfeiffer, Peter 37
 Ponzetto, Simone Paolo 315
 Pufahl, Luise 257

R

Reichert, Manfred 89
 Reijers, Hajo A. 322
 Resinas, Manuel 210
 Röglinger, Maximilian 179
 Ronzani, Massimiliano 197
 Rosenau, Marc 231

S

Seiger, Ronny 76
 Senderovich, Arik 13
 Serral, Estefanía 63
 Skarbovsky, Inna 45
 Slaats, Tijds 286
 Stein Dani, Vinicius 322
 Sulis, Emilio 197

T

Terboven, Carla 231
 Timmermans, Yves 25

V

Van den Bussche, Jan 132
 van der Aa, Han 315
 van der Werf, Jan Martijn E. M. 322
 Van Looy, Amy 179
 van Zelst, Sebastiaan 117
 Vandin, Andrea 243
 Vanthienen, Jan 25

W

Weber, Barbara 76, 179
 Weber, Ingo 257
 Weinzierl, Sven 303
 Weske, Mathias 231
 Weyers, Flemming 76
 Wolf, Julie Ryan 149

Y

Yu, Eric 13

Z

Zhang, Quan 149
 Zilker, Sandra 303