

**Thomas Hildebrandt
Boudewijn F. van Dongen
Maximilian Röglinger
Jan Mendling (Eds.)**

LNCS 11675

Business Process Management

**17th International Conference, BPM 2019
Vienna, Austria, September 1–6, 2019
Proceedings**

 **Springer**

Founding Editors

Gerhard Goos

Karlsruhe Institute of Technology, Karlsruhe, Germany

Juris Hartmanis

Cornell University, Ithaca, NY, USA

Editorial Board Members

Elisa Bertino

Purdue University, West Lafayette, IN, USA

Wen Gao

Peking University, Beijing, China

Bernhard Steffen

TU Dortmund University, Dortmund, Germany

Gerhard Woeginger

RWTH Aachen, Aachen, Germany

Moti Yung

Columbia University, New York, NY, USA


More information about this series at <http://www.springer.com/series/7409>


Thomas Hildebrandt · Boudewijn F. van Dongen ·
Maximilian Röglinger · Jan Mendling (Eds.)


Business Process Management


17th International Conference, BPM 2019
Vienna, Austria, September 1–6, 2019
Proceedings

Editors

Thomas Hildebrandt 
University of Copenhagen
Copenhagen, Denmark

Maximilian Röglinger 
University of Bayreuth
Bayreuth, Germany

Boudewijn F. van Dongen 
Eindhoven University of Technology
Eindhoven, The Netherlands

Jan Mendling 
Vienna University of Economics
and Business
Vienna, Austria

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-26618-9 ISBN 978-3-030-26619-6 (eBook)
<https://doi.org/10.1007/978-3-030-26619-6>

LNCS Sublibrary: SL3 – Information Systems and Applications, incl. Internet/Web, and HCI

© Springer Nature Switzerland AG 2019

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

The 17th International Conference on Business Process Management provided a forum for researchers and practitioners in the broad and diverse field of business process management (BPM). To accommodate for the diversity of the field, the BPM conference hosted tracks for foundations, engineering, and management. These tracks cover not only different phenomena of interest and research methods, but also ask for different evaluation criteria. Each track had a dedicated track chair and Program Committee. The track chairs, together with a consolidation chair, were responsible for the scientific program.

BPM 2019 was organized by Jan Mendling, Vienna University of Economics and Business (WU Vienna), and Stefanie Rinderle-Ma, University of Vienna. The conference was held in Vienna, Austria, on the WU Campus during September 1–6, 2019.

The conference received 157 full paper submissions, well-distributed over the tracks, from which 115 went for review. Each paper was reviewed by at least three Program Committee members and a senior Program Committee member who initiated and moderated scientific discussions and reflected these in an additional meta-review. We accepted 23 excellent papers in the main conference (acceptance rate 20%). Moreover, 13 submissions appeared in the BPM Forum, published in a separate volume of the Springer LNBIP series.

In the foundations track chaired by Thomas Hildebrandt, core BPM topics including process verification and formal analysis were presented. The engineering track was chaired by Boudewijn van Dongen. This track featured topics such as process mining and machine learning. The management track was chaired by Maximilian Röglinger. Many papers in this track focused on matters related to process design and improvement as well as organizational challenges such as trusted processes, the alignment of employee rewards with processes, BPM skill configurations, and the adoption of process mining in practice.

Moreover, we had one keynote per track as well as for the Industry Track and the Blockchain Forum. Pat Geary from BluePrism shared his latest insights into robotic process automation, Monika Henzinger from the University of Vienna informed us about the state of the art in dynamic graph algorithms, and Kalle Lyytinen from Case Western University offered a digitalization and organizational routines perspective on BPM. Moreover, Petr Novotny from IBM Research shared a technical perspective on Hyperledger Fabric and its applications, while Max Pucher from Papyrus talked about adaptive case management with Converse. Finally, Stefan Schulte from TU Vienna shared insights on the interoperability of blockchains.

Organizing a scientific conference is a complex process, involving many roles and many more interactions. We thank all our colleagues involved for their excellent work. The workshop chairs attracted 15 innovative workshops, the industry chairs organized a top-level industry program, and the demo chairs attracted many excellent demos. The panel chairs compiled an exciting panel, which opened doors to future research

challenges. Without the publicity chairs, we could not have attracted such an excellent number of submissions. Younger researchers benefited from excellent tutorials; doctoral students received feedback about their work from experts in the field at the Doctoral Consortium. The mini-sabbatical program helped to bring additional colleagues to Austria. The proceedings chair, Claudio Di Ciccio, professionally interacted with Springer and with the authors to prepare excellent LNCS and LNBIP.

The members of the track Program Committees and senior Program Committees deserve particular acknowledgment for their dedication and commitment. We are grateful for the help and expertise of the additional reviewers, who provided valuable feedback during the reviewing process and engaged in deep discussions at times. BPM 2019 had a dedicated process to consolidate paper acceptance across tracks. During the very intensive weeks of this phase, many senior Program Committee members evaluated additional papers and were engaged in additional discussions. Special thanks goes to these colleagues, who were instrumental during this decisive phase of the reviewing process.

We cordially thank the colleagues involved in the organization of the conference, especially the members of the Program and Organizing Committees. We thank the Platinum Sponsor Signavio, the Gold Sponsors Austrian Center for Digital Production, Bizagi, Camunda, Celonis, FireStart, Process4.biz, the Silver Sponsors Heflo, JIT, Minit, Papyrus Software, Phactum, and the Bronze Sponsors ConSense, DCR, and TIM Solutions, as well as Springer and Gesellschaft für Prozessmanagement for their support. We also thank WU Vienna and the University of Vienna for its enormous and high-quality support. Finally, we thank the Organizing Committee and the local Organizing Committee, namely, Martin Beno, Katharina Distelbacher-Kollmann, Ilse Kondert, Roman Franz, Alexandra Hager, Prabh Jit, and Doris Wyk.

September 2019

Thomas Hildebrandt
Boudewijn van Dongen
Maximilian Röglinger
Jan Mendling

Organization

The 17th International Conference on Business Process Management (BPM 2019) was organized by the Vienna University of Economics and Business (WU Vienna) and the University of Vienna, and took place in Vienna, Austria.

Steering Committee

Mathias Weske (Chair)	HPI, University of Potsdam, Germany
Boualem Benatallah	University of New South Wales, Australia
Jörg Desel	Fernuniversität in Hagen, Germany
Schahram Dustdar	TU Wien, Austria (until 2018)
Marlon Dumas	University of Tartu, Estonia
Wil van der Aalst	RWTH Aachen University, Germany
Michael zur Muehlen	Stevens Institute of Technology, USA (until 2018)
Barbara Weber	University of St. Gallen, Switzerland
Stefanie Rinderle-Ma	University of Vienna, Austria
Manfred Reichert	University of Ulm, Germany
Jan Mendling	WU Vienna, Austria

Executive Committee

General Chairs

Jan Mendling	WU Vienna, Austria
Stefanie Rinderle-Ma	University of Vienna, Austria

Main Conference Program Chairs

Thomas Hildebrandt	University of Copenhagen, Denmark (Track I)
Boudewijn van Dongen	Eindhoven University of Technology, The Netherlands (Track II)
Maximilian Röglinger	University of Bayreuth, Germany (Track III)
Jan Mendling	WU Vienna, Austria (Consolidation)

Blockchain Forum Chairs

Claudio Di Ciccio	WU Vienna, Austria
Luciano García-Bañuelos	Tecnológico de Monterrey, Mexico
Richard Hull	IBM Research, USA
Mark Staples	Data61, CSIRO, Australia

Central Eastern European Forum Chairs

Renata Gabryelczyk	University of Warsaw, Poland
Andrea Kő	Corvinus University of Budapest, Hungary

Tomislav Hernaus University of Zagreb, Croatia
Mojca Indihar Štemberger University of Ljubljana, Slovenia

Industry Track Chairs

Jan vom Brocke University of Liechtenstein, Liechtenstein
Jan Mendling WU Vienna, Austria
Michael Rosemann Queensland University of Technology, Australia

Workshop Chairs

Remco Dijkman Eindhoven University of Technology, The Netherlands
Chiara Di Francescomarino Fondazione Bruno Kessler-IRST, Italy
Uwe Zdun University of Vienna, Austria

Demonstration Chairs

Benoît Depaire Hasselt University, Belgium
Stefan Schulte TU Wien, Austria
Johannes de Smedt University of Edinburgh, UK

Tutorial Chairs

Akhil Kumar Penn State University, USA
Manfred Reichert University of Ulm, Germany
Pnina Soffer University of Haifa, Israel

Panel Chairs

Jan Recker University of Cologne, Germany
Hajo A. Reijers Utrecht University, The Netherlands

BPM Dissertation Award Chair

Stefanie Rinderle-Ma University of Vienna, Austria

Event Organization Chair

Monika Hofer-Mozelt University of Vienna, Austria

Local Organization Chairs

Katharina WU Vienna, Austria
Disselbacher-Kollmann
Roman Franz WU Vienna, Austria
Ilse Dietlinde Kondert WU Vienna, Austria

Publicity Chair

Cristina Cabanillas WU Vienna, Austria

Web and Social Media Chair

Philipp Waibel WU Vienna, Austria

Proceedings Chair

Claudio Di Ciccio WU Vienna, Austria

Track I – Foundations**Senior Program Committee**

Florian Daniel Politecnico di Milano, Italy
 Jörg Desel Fernuniversität in Hagen, Germany
 Chiara Di Francescomarino Fondazione Bruno Kessler-IRST, Italy
 Dirk Fahland Eindhoven University of Technology, The Netherlands
 Marcello La Rosa The University of Melbourne, Australia
 Fabrizio Maria Maggi University of Tartu, Estonia
 Marco Montali Free University of Bozen-Bolzano, Italy
 John Mylopoulos University of Toronto, Canada
 Manfred Reichert University of Ulm, Germany
 Victor Vianu University of California San Diego, USA
 Hagen Völzer IBM Research – Zurich, Switzerland
 Mathias Weske HPI, University of Potsdam, Germany

Program Committee

Ahmed Awad University of Tartu, Estonia
 Jan Claes Ghent University, Belgium
 Søren Debois IT University of Copenhagen, Denmark
 Claudio Di Ciccio WU Vienna, Austria
 Rik Eshuis Eindhoven University of Technology, The Netherlands
 Hans-Georg Fill University of Fribourg, Switzerland
 Guido Governatori Data61, CSIRO, Australia
 Gianluigi Greco University of Calabria, Italy
 Richard Hull IBM Research, USA
 Irina Lomazova National Research University Higher School
 of Economics, Russia
 Andrea Marrella Sapienza University of Rome, Italy
 Oscar Pastor Lopez Universitat Politècnica de València, Spain
 Artem Polyvyanyy The University of Melbourne, Australia
 Wolfgang Reisig Humboldt-Universität zu Berlin, Germany
 Arik Senderovich University of Toronto, Canada
 Tijs Slaats University of Copenhagen, Denmark
 Ernest Teniente Universitat Politècnica de Catalunya, Spain
 Daniele Theseider Dupré Università del Piemonte Orientale, Italy

Track II – Engineering

Senior Program Committee

Boualem Benatallah	The University of New South Wales, Australia
Josep Carmona	Universitat Politècnica de Catalunya, Spain
Cesare Pautasso	University of Lugano, Switzerland
Hajo A. Reijers	Utrecht University, The Netherlands
Prina Soffer	University of Haifa, Israel
Wil van der Aalst	RWTH Aachen University, Germany
Ingo Weber	Data61, CSIRO, Australia
Barbara Weber	University of St. Gallen, Switzerland
Matthias Weidlich	Humboldt-Universität zu Berlin, Germany
Lijie Wen	Tsinghua University, China

Program Committee

Marco Aiello	University of Stuttgart, Germany
Amin Beheshti	Macquarie University, Australia
Andrea Burattin	Technical University of Denmark, Denmark
Cristina Cabanillas	WU Vienna, Austria
Fabio Casati	University of Trento, Italy
Massimiliano de Leoni	University of Padua, Italy
Jochen De Weerd	KU Leuven, Belgium
Remco Dijkman	Eindhoven University of Technology, The Netherlands
Marlon Dumas	University of Tartu, Estonia
Schahram Dustdar	TU Wien, Austria
Gregor Engels	Paderborn University, Germany
Joerg Evermann	Memorial University of Newfoundland, Canada
Walid Gaaloul	Télécom SudParis, France
Avigdor Gal	Technion, Israel
Luciano García-Bañuelos	Tecnológico de Monterrey, Mexico
Chiara Ghidini	Fondazione Bruno Kessler (FBK), Italy
Daniela Grigori	Laboratoire LAMSADE, University of Paris-Dauphine, France
Dimka Karastoyanova	University of Groningen, The Netherlands
Christopher Klinkmüller	Data61, CSIRO, Australia
Agnes Koschmider	Karlsruhe Institute of Technology, Germany
Jochen Kuester	FH Bielefeld, Germany
Henrik Leopold	Kühne Logistics University, Germany
Raimundas Matulevičius	University of Tartu, Estonia
Massimo Mecella	Sapienza University of Rome, Italy
Hamid Motahari	Ernst & Young (EY), USA
Jorge Munoz-Gama	Pontificia Universidad Católica de Chile, Chile
Hye-Young Paik	The University of New South Wales, Australia
Luise Pufahl	Hasso Plattner Institute, University of Potsdam, Germany

Manuel Resinas	University of Seville, Spain
Minseok Song	POSTECH (Pohang University of Science and Technology), South Korea
Arthur ter Hofstede	Queensland University of Technology, Australia
Farouk Toumani	Limos, Blaise Pascal University, Clermont-Ferrand, France
Eric Verbeek	Eindhoven University of Technology, The Netherlands
Moe Wynn	Queensland University of Technology, Australia
Bas van Zelst	Fraunhofer Gesellschaft Aachen, Germany

Track III – Management

Senior Program Committee

Jörg Becker	University of Münster, ERCIS, Germany
Adela Del Río Ortega	University of Seville, Spain
Marta Indulska	The University of Queensland, Australia
Susanne Leist	University of Regensburg, Germany
Mikael Lind	Research Institutes of Sweden (RISE)/Chalmers University of Technology, Sweden
Peter Loos	IWi at DFKI, Saarland University, Germany
Jan Recker	University of Cologne, Germany
Michael Rosemann	Queensland University of Technology, Australia
Flavia Santoro	University of the State of Rio de Janeiro, Brazil
Peter Trkman	University of Ljubljana, Slovenia
Amy Van Looy	Ghent University, Belgium
Robert Winter	University of St. Gallen, Switzerland

Program Committee

Wasana Bandara	Queensland University of Technology, Australia
Daniel Beimborn	University of Bamberg, Germany
Daniel Beverungen	Paderborn University, Germany
Alessio Maria Braccini	University of Tuscia, Italy
Michael Fellmann	University of Rostock, Institute for Computer Science, Germany
Peter Fettke	German Research Center for Artificial Intelligence (DFKI) and Saarland University, Germany
Kathrin Figl	University of Innsbruck, Austria
Andreas Gadatsch	Hochschule Bonn-Rhein-Sieg, Germany
Paul Grefen	Eindhoven University of Technology, The Netherlands
Thomas Grisold	University of Liechtenstein, Liechtenstein
Bernd Heinrich	Universität Regensburg, Germany
Mojca Indihar Štemberger	University of Ljubljana, Slovenia
Christian Janiesch	TU Dresden, Germany
Florian Johannsen	University of Applied Sciences Schmalkalden, Germany

John Krogstie	Norwegian University of Science and Technology, Norway
Michael Leyer	University of Rostock, Germany
Alexander Mädche	Karlsruhe Institute of Technology, Germany
Monika Malinova	WU Vienna, Austria
Juergen Moormann	Frankfurt School of Finance & Management, Germany
Michael zur Muehlen	Stevens Institute of Technology, USA
Markus Nüttgens	University of Hamburg, Germany
Sven Overhage	University of Bamberg, Germany
Geert Poels	Ghent University, Belgium
Jens Poeppelbuss	Ruhr-Universität Bochum, Germany
Michael Räckers	University of Münster, ERCIS, Germany
Shazia Sadiq	The University of Queensland, Australia
Bernd Schenk	University of Liechtenstein, Liechtenstein
Werner Schmidt	Technische Hochschule Ingolstadt (THI Business School), Germany
Theresa Schmiedel	University of Applied Sciences and Arts Northwestern Switzerland, Switzerland
Anna Sidorova	University of North Texas, USA
Oktay Türetken	Eindhoven University of Technology, The Netherlands
Irene Vanderfeesten	Eindhoven University of Technology, The Netherlands
Jan Vanthienen	KU Leuven, Belgium
Axel Winkelmann	University of Würzburg, Germany

Additional Reviewers

Abasi-Amefon Affia	Christian Fleig	Michael Poppe
Simone Agostinelli	Lukas-Valentin Herm	Raphael Schilling
Ivo Benke	Mubashar Iqbal	Roe Shraga
Sabrina Blaukopf	Samuel Kießling	Ludwig Stage
Djordje Djurica	Fabienne Lambusch	Peyman Toreini
Montserrat Estañol	Xavier Oriol	Jonas Wanner
Florian Fahrenbach	Baris Ozkan	Bastian Wurm

Abstracts of Keynotes

Digitalization and Routines - Another Look at Business Process Management

Kalle Lyytinen

Case Western Reserve University, Cleveland, OH 44106, USA
kalle@case.edu

Abstract. This key note examines the changing role of business process management from a broader perspective of organizational change routines. It seeks to integrate and contextualize business process management with recent advances in theorizing about routines and their change in the context of deep digitalization of firm operations. While business process management emerges from the internal, computational view of business process management advocated by computer science community since the 80's where the driving concern has been for semantically correct computational processes and their efficient enactment, the research on organizational routines provides an external view of this focal phenomenon. Studies of organizational routines have emerged from evolutionary economics and organizational theory as scholars have sought to examine and explain the structuring, evolution and effects of business routines and their enactment in evolving organizational and industrial contexts. The main focus of these inquiries been to account for economic or organizational effectiveness of coordination and collaboration tasks enabled by varying routine constellations. In the process management stream the key concerns have been efficient process execution, its semantic correctness, consistent state maintenance, exception handling and possibility for process mining and discovery. In the latter the dominant issues have been environmental fit and change, adaptability, maintenance of organizational memory, performativity and learning from and for process execution. The talk will review both research streams and look for their strengths and weaknesses. We call also for better integration and receptivity of organizational theory concepts to process management and execution and the need for routine theory to recognize the relevance of computational perspectives in accounting for process change and execution. The need for integration has become increasingly relevant and intra and inter-organizational business processes are increasingly automated and engulfed with intelligence. Several examples of using routines as a framing device to study organizational change are offered to demonstrate the value and relevance of such view on process management scholarship.

Keywords: Business process management · Routines · Routine theory · Coordination · Collaboration · Process analysis · Sequence analysis · Process mining · Evolutionary economics

Understanding the Potential of “Real RPA”

Pat Geary

Blue Prism, 338 Euston Road, London NW1 3BG, UK
pat.geary@blueprism.com

Abstract. Since the RPA software category was created over 17 years ago, it now successfully operates in large-scale, demanding, enterprise environments. RPA enables business teams to orchestrate easy-to-control, automated, digital workers to drive tactical change at large organizations - so they keep up with ever-changing demand. This potential to drive digital transformation across the world’s workplace operations has created a rapidly expanding market – with over 40 vendors offering some sort of RPA - or intelligent automation capabilities.

This category is now entering its next evolutionary phase – ‘connected-RPA’ – which is an even more intelligent form of business automation that enables collaborative, technology innovation – and it’s also widely acknowledged as providing the foundation for the AI revolution. Ultimately, the future of work is being driven by connected-RPA, and this is already occurring with the creation of entire communities, consisting of teams of humans – working in tandem with thinking, digital workers.

However, connected-RPA is complex and relatively misunderstood, so without a definitive reference point, organizations risk choosing the wrong automation options for enterprise environments. I’ll provide insight gained from more than a decade of working with the pioneers of connected-RPA technology, discuss why it’s more important than ever to re-define what this technology is – and what it isn’t – and where it sits within its industry.

I’ll also explain why connected-RPA is becoming the true standard for collaboration, securely, at scale for global organizations and how it can deliver sustainable, digital transformation. Finally, I’ll highlight how connected-RPA will play a key role in shaping the future of work and what outcomes are being achieved now.

Keywords: Robotic process automation · Digital transformation · Collaborative technology innovation

The State of the Art in Dynamic Graph Algorithms

Monika Henzinger

University of Vienna, Währinger Straße 29, 1090 Wien, Austria
monika.henzinger@univie.ac.at

Abstract. A dynamic graph algorithm is a data structure that maintains information about a graph that is being modified by edge and vertex insertions and deletions. Due to the recent increase of dynamically changing huge graphs (such as social network graphs) there is a rising demand for such data structures.

More formally, a dynamic graph algorithm is given an initial graph for which it builds a representation. With this representation it then supports two types of operations, namely (a) update operations and (b) queries. Each update operation modifies the representations to reflect the insertion or deletion of the edge or (isolated) vertex given as parameter to the update operation. Each query operation returns information about a graph property in the current graph. The challenge is to design a data structure for which each update operation is much faster than running the corresponding static algorithm, i.e., recomputation from scratch while keeping the query time small (usually constant or polylogarithmic in the size of the graph). Surprisingly, such data structures exist for certain basic graph properties such as connected components, minimum spanning forest, and maximal matching. Recent conditional lower bounds show, however, that for many other graph properties, such as single-source reachability and single-source shortest paths, no dynamic data structure can exist that is more efficient than the corresponding static algorithm.

We will survey some of the upper and lower bound techniques in this talk.

Keyword: Dynamic graph algorithms

Contents

Tutorials

Everything You Always Wanted to Know About Petri Nets, but Were Afraid to Ask	3
<i>Wil M. P. van der Aalst</i>	
Responsible Process Mining - A Data Quality Perspective	10
<i>Moe Thandar Wynn and Shazia Sadiq</i>	
IoT for BPMers. Challenges, Case Studies and Successful Applications	16
<i>Francesco Leotta, Andrea Marrella, and Massimo Mecella</i>	
Exploring Explorative BPM - Setting the Ground for Future Research	23
<i>Thomas Grisold, Steven Gross, Maximilian Röglinger, Katharina Stelzl, and Jan vom Brocke</i>	

Foundations

Dynamic Reconfiguration of Business Processes	35
<i>Leandro Nahabedian, Victor Braberman, Nicolás D'ippolito, Jeff Kramer, and Sebastián Uchitel</i>	
A First-Order Logic Semantics for Communication-Parametric BPMN Collaborations	52
<i>Sara Houhou, Souheib Baarir, Pascal Poizat, and Philippe Quéinnec</i>	
Modeling and Enforcing Blockchain-Based Choreographies	69
<i>Jan Ladleif, Mathias Weske, and Ingo Weber</i>	
Formal Reasoning on Natural Language Descriptions of Processes	86
<i>Josep Sànchez-Ferreres, Andrea Burattin, Josep Carmona, Marco Montali, and Lluís Padró</i>	
Goal-oriented Process Enhancement and Discovery	102
<i>Mahdi Ghasemi and Daniel Amyot</i>	
Checking Regulatory Compliance: Will We Live to See It?	119
<i>Silvano Colombo Tosatto, Guido Governatori, and Nick van Beest</i>	
Modeling and Reasoning over Declarative Data-Aware Processes with Object-Centric Behavioral Constraints.	139
<i>Alessandro Artale, Alisa Kovtunova, Marco Montali, and Wil M. P. van der Aalst</i>	

Formal Modeling and SMT-Based Parameterized Verification
of Data-Aware BPMN 157
*Diego Calvanese, Silvio Ghilardi, Alessandro Gianola, Marco Montali,
and Andrey Rivkin*

Engineering

Estimating Process Conformance by Trace Sampling
and Result Approximation 179
Martin Bauer, Han van der Aa, and Matthias Weidlich

Trace Clustering on Very Large Event Data in Healthcare
Using Frequent Sequence Patterns 198
*Xixi Lu, Seyed Amin Tabatabaei, Mark Hoogendoorn,
and Hajo A. Reijers*

ProcessExplorer: Intelligent Process Mining Guidance 216
*Alexander Seeliger, Alejandro Sánchez Guinea, Timo Nolle,
and Max Mühlhäuser*

Machine Learning-Based Framework for Log-Lifting in Business
Process Mining Applications 232
Ghalia Tello, Gabriele Gianini, Rabeb Mizouni, and Ernesto Damiani

Approximate Computation of Alignments of Business Processes
Through Relaxation Labelling 250
Lluís Padró and Josep Carmona

Metaheuristic Optimization for Automated Business Process Discovery 268
Adriano Augusto, Marlon Dumas, and Marcello La Rosa

Learning Accurate LSTM Models of Business Processes 286
Manuel Camargo, Marlon Dumas, and Oscar González-Rojas

Management

Trust-Aware Process Design 305
Michael Rosemann

Mining Process Mining Practices: An Exploratory Characterization
of Information Needs in Process Analytics 322
Christopher Klinkmüller, Richard Müller, and Ingo Weber

Towards a Process Reference Model for Research Management:
An Action Design Research Effort at an Australian University 338
Jeremy Gibson, Kanika Goel, Janne Barnes, and Wasana Bandara

What the Hack? – Towards a Taxonomy of Hackathons 354
Christoph Kollwitz and Barbara Dinter

Design Patterns for Business Process Individualization. 370
Bastian Wurm, Kanika Goel, Wasana Bandara, and Michael Rosemann

Understanding the Alignment of Employee Appraisals and Rewards
with Business Processes. 386
Aygun Shafagatova and Amy Van Looy

Business Process Improvement Activities: Differences in Organizational
Size, Culture, and Resources 402
Iris Beerepoot, Inge van de Weerd, and Hajo A. Reijers


Regulatory Instability, Business Process Management Technology,
and BPM Skill Configurations 419
Patrick Lohmann and Michael zur Muehlen

Author Index 437

Tutorials



Everything You Always Wanted to Know About Petri Nets, but Were Afraid to Ask

Wil M. P. van der Aalst^{1,2} 

¹ Process and Data Science (PADS), RWTH Aachen University, Aachen, Germany

² Fraunhofer Institute for Applied Information Technology,
Sankt Augustin, Germany

wvdaalst@pads.rwth-aachen.de

Abstract. Business Process Management (BPM), Process Mining (PM), Workflow Management (WFM), and other approaches aimed at improving processes depend on process models. Business Process Model and Notation (BPMN), Event-driven Process Chains (EPCs), and UML activity diagrams all build on Petri nets and have semantics involving ‘playing the token game’. In addition, process analysis approaches ranging from verification and simulation to process discovery and compliance checking often depend on Petri net theory. For the casual user, there is no need to understand the underlying foundations. However, BPM/PM/WFM researchers and ‘process experts’ working in industry need to understand these foundational results. Unfortunately, the results of 50 years of Petri net research are not easy to digest. This tutorial paper provides, therefore, an entry point into the wonderful world of Petri nets.

Keywords: Petri nets · Business Process Management · Process Mining

1 Petri Nets and Business Process Management

Since their inception in 1962, Petri nets have been used in a wide variety of application domains. A more recent development is the foundational role of Petri nets in Business Process Management (BPM) [8] and related fields such as Process Mining (PM) [2] and Workflow Management (WFM) [3]. Many WFM systems are based on Petri nets. In fact, the first prototypes developed in the late 1970-ties (e.g., Officetalk and SCOOP) already used Petri nets [1]. In today’s BPM/WFM systems, this is less visible. However, popular modeling languages such as Business Process Model and Notation (BPMN), Event-driven Process Chains (EPCs), and UML activity diagrams all borrow ideas from Petri nets (e.g., the ‘token game’ to describe semantics and to implement BPM/WFM engines and simulation tools) [5].

Petri nets also play a major role in the analysis of processes and event data. Many simulation tools are based on Petri nets [5]. Petri nets are also used for

the verification of processes in WFM/BPM systems, e.g., to check soundness [4]. However, this possibility is not used much in practice. Conversely, Process Mining (PM) is much more widely used than simulation and verification. Petri nets are the most widely used representation in PM [2]. There are dozens of techniques that can discover a Petri net from event data. Moreover, almost all conformance checking techniques use Petri nets internally.

This short paper is based on a tutorial with the same name presented at the 17th International Conference on Business Process Management (BPM 2019) in Vienna in September 2019. Here, we can only show a few of the ‘gems in Petri nets’ relevant for BPM, PM, and WFM.

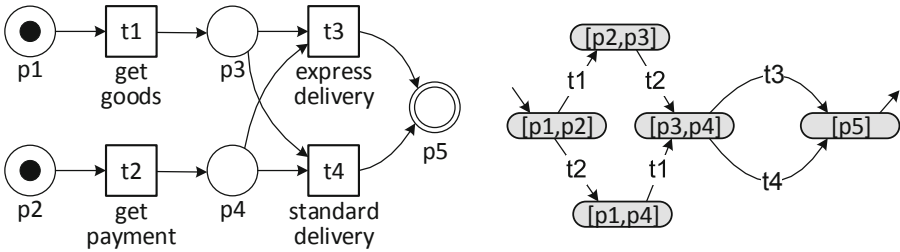


Fig. 1. An accepting Petri net N_1 (left) with transitions $T_1 = \{t_1, t_2, t_3, t_4\}$, places $P_1 = \{p_1, p_2, p_3, p_4, p_5\}$, initial marking $[p_1, p_2]$, and final marking $[p_5]$. N_1 allows for traces $traces(N_1) = \{\langle t_1, t_2, t_3 \rangle, \langle t_2, t_1, t_3 \rangle, \langle t_1, t_2, t_4 \rangle, \langle t_2, t_1, t_4 \rangle\}$. The reachability graph (right) shows the reachable markings $states(N_1) = \{[p_1, p_2], [p_1, p_4], [p_2, p_3], [p_3, p_4], [p_5]\}$.

2 Accepting Petri Nets

Figures 1 and 2 show two so-called *accepting Petri nets*. These Petri nets have an initial state and a final state. States in Petri nets are called *markings* that mark certain *places* (represented by circles) with *tokens* (represented by black dots). The accepting Petri net N_1 in Fig. 1 has five places. In the initial marking, $[p_1, p_2]$ two places are marked. Since a place may have multiple tokens, markings are represented by multisets. *Transitions* (represented by squares) are the active components able to move the Petri nets from one marking to another marking. N_1 has four transitions. A transition is called *enabled* if each of the input places has a token. An enabled transition may *fire* (i.e., *occur*) thereby consuming a token from each input place and producing a token for each output places. In the marking showing in Fig. 1, both t_1 and t_2 are enabled. Firing t_1 removes a token from p_1 and adds a token to p_3 . Firing t_2 removes a token from p_2 and adds a token to p_4 . In the resulting marking $[p_3, p_4]$ both t_3 and t_4 are enabled. Note that both transitions require both input places to be marked. However, only one of them can fire. Firing t_3 (or t_4) removes a token from both p_3 and p_4 and adds one token to p_5 . Transitions may be labeled, e.g., transitions $t_1, t_2, t_3,$

and $t4$ represent the activities “get goods”, “get payment”, “express delivery”, and “normal delivery” respectively. For simplicity, we ignore the transition labels and only use the short transition names.

The *behavior* of an accepting Petri net is described by all *traces* starting in the initial marking and ending in the final marking. $\langle t1, t2, t3 \rangle$ is one of the four traces of accepting Petri net N_1 . Figure 1 also shows the reachability graph of N_1 . The reachability graph shows all reachable markings and their connections. N_1 has five reachable states.

Accepting Petri net N_2 depicted in Fig. 2 also has five reachable states, but allows for infinitely many traces (due to the loop involving $t1$ and $t2$).

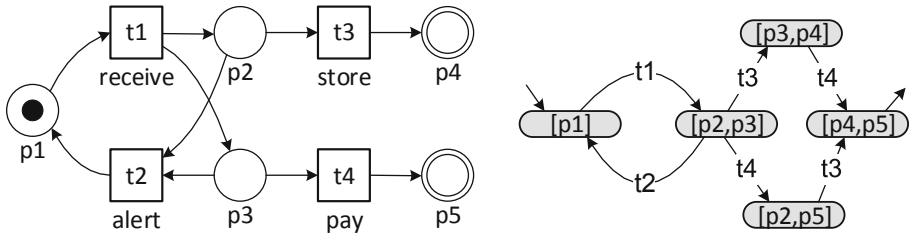


Fig. 2. An accepting petri net N_2 (left) with initial marking $[p1]$ and final marking $[p4, p5]$. N_2 allows for infinitely many traces $traces(N_2) = \{\langle t1, t3, t4 \rangle, \langle t1, t4, t3 \rangle, \langle t1, t2, t1, t3, t4 \rangle, \langle t1, t2, t1, t4, t3 \rangle, \langle t1, t2, t1, t2, t1, t3, t4 \rangle, \dots\}$. The reachability graph (right) shows the reachable markings $states(N_2) = \{[p1], [p2, p3], [p2, p5], [p3, p4], [p4, p5]\}$.

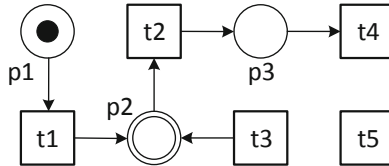


Fig. 3. An accepting petri net N_3 with initial marking $[p1]$ and final marking $[p2]$ showing the declarative nature of Petri nets.

3 Petri Nets Are More Declarative Than You Think

Petri nets are typically considered ‘procedural’ (like an imperative program) and not ‘declarative’. However, an accepting Petri net *without* places allows for any trace involving the transitions in the net. Each place corresponds to a *constraint*. Consider for example the accepting Petri net N_3 in Fig. 3. Place $p1$ models the constraint that $t1$ should occur precisely once. Place $p2$ models the constraint that $t2$ can only occur after $t1$ or $t3$. Each occurrence of $t2$ requires an earlier occurrence of $t1$ or $t3$ and, at the end, the number of occurrences of $t2$ is one

less than the sum of $t1$ and $t3$. Place $p3$ models the constraint that each $t4$ occurrence should be preceded by a $t2$ occurrence and, at the end, the number of occurrences of both $t2$ and $t4$ need to be the same. Note that transition $t5$ is not constrained by one of the places and can occur at any point in time and an arbitrary number of times. Removing a place can only enable more traces, thus illustrating the declarative nature of Petri nets (anything is possible unless specified otherwise).

4 Structure Theory and the Marking Equation

Structure theory focuses on behavioral properties that can be derived from the structural properties of a Petri net [6,7,9]. It is impossible to go into details. Therefore, we restrict ourselves to the *marking equation* which nicely shows how linear algebra can be used to exploit the structure of a Petri net. To start, we represent the first two Petri nets as a matrix with a row for each place and a column for each transition. The so-called incidence matrix shows the ‘net effect’ of firing a transition (column) on each place (row).

$$\mathbf{N}_1 = \begin{matrix} & t1 & t2 & t3 & t4 \\ p1 & \begin{pmatrix} -1 & 0 & 0 & 0 \end{pmatrix} \\ p2 & \begin{pmatrix} 0 & -1 & 0 & 0 \end{pmatrix} \\ p3 & \begin{pmatrix} 1 & 0 & -1 & -1 \end{pmatrix} \\ p4 & \begin{pmatrix} 0 & 1 & -1 & -1 \end{pmatrix} \\ p5 & \begin{pmatrix} 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix} \quad \mathbf{N}_2 = \begin{matrix} & t1 & t2 & t3 & t4 \\ p1 & \begin{pmatrix} -1 & 1 & 0 & 0 \end{pmatrix} \\ p2 & \begin{pmatrix} 1 & -1 & -1 & 0 \end{pmatrix} \\ p3 & \begin{pmatrix} 1 & -1 & 0 & -1 \end{pmatrix} \\ p4 & \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \\ p5 & \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

The incidence matrix imposes an order on places (rows) and transitions (columns). For N_1 and N_2 , the order is $p1, p2, p3, p4, p5$ and $t1, t2, t3, t4$. $\mathbf{t} = (1, 1, 1, 0)^T$ is an example of a transition column vector assigning value 1 to $t1$, $t2$, and $t3$ and value 0 to $t4$. $\mathbf{p} = (1, 1, 0, 0, 0)^T$ is an example of a place column vector assigning value 1 to $p1$ and $p2$, and value 0 to $p3$, $p4$, and $p5$. Assume that \mathbf{p}' and \mathbf{p}'' are two place column vectors representing the initial marking \mathbf{p}' and a target marking \mathbf{p}'' . If \mathbf{p}'' is reachable from \mathbf{p}' in some Petri net having incidence matrix \mathbf{N} , then the so-called marking equation

$$\mathbf{p}' + \mathbf{N} \cdot \mathbf{t} = \mathbf{p}''$$

has a solution for some transition column vector \mathbf{t} with non-negative values.

Consider N_1 in Fig. 1. We are interested in the different ways to get from the initial marking $[p1, p2]$ to the final marking $[p5]$. Hence, $\mathbf{p}' = (1, 1, 0, 0, 0)^T$ and $\mathbf{p}'' = (0, 0, 0, 0, 1)^T$, resulting in the following marking equation:

$$\mathbf{p}' + \mathbf{N} \cdot \mathbf{t} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & -1 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} t1 \\ t2 \\ t3 \\ t4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \mathbf{p}''$$

Hence, we can infer from the marking equation that $t_1 = 1$, $t_2 = 1$, and $t_3 + t_4 = 1$. Since N_1 allows for trace $\langle t_1, t_2, t_3 \rangle$, we know that $t_1 = t_2 = t_3 = 1$ and $t_4 = 0$ should be a solution. Suppose we would like to know whether N_1 allows for trace $\langle t_1, t_3, t_4 \rangle$. Since $t_1 = t_3 = t_4 = 1$ and $t_2 = 0$ is not solution of the marking equation, we know that $\langle t_1, t_3, t_4 \rangle$ is impossible *without* replaying the trace. For such a small example, this may seem insignificant. However, the marking equation provides a powerful ‘algebraic overapproximation’ of all possible traces. Note that the marking equation provides a necessary but not a sufficient condition. The algebraic overapproximation can be used to quickly prune search spaces in verification and conformance checking. For example, the marking equation can be used to guide the search for so-called optimal alignments in conformance checking [2].

The marking equation is related to place and transitions invariants. Any solution of the equation $\mathbf{p} \cdot \mathbf{N} = \mathbf{0}$ is a *place invariant*. For net N_1 :

$$\mathbf{p} \cdot \mathbf{N} = (p_1, p_2, p_3, p_4, p_5) \cdot \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & -1 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = (0, 0, 0, 0) = \mathbf{0}$$

For example, $\mathbf{p} = (p_1, p_2, p_3, p_4, p_5) = (1, 0, 1, 0, 1)$ is the place invariant showing that the number of tokens in the places p_1 , p_3 , and p_5 is constant. The so-called ‘weighted token sum’ is constant for any initial marking. Given the initial marking $[p_1, p_2]$, the weighted token sum is 1. If the initial marking is $[p_1^2, p_3, p_4^3, p_5]$, the weighted token sum is $(2 \times 1) + (0 \times 0) + (1 \times 1) + (3 \times 0) + (1 \times 1) = 4$ and will not change. $\mathbf{p} = (0, 1, 0, 1, 1)$, $\mathbf{p} = (1, 1, 1, 1, 2)$, and $\mathbf{p} = (1, -1, 1, -1, 0)$ are other place invariants since $\mathbf{p} \cdot \mathbf{N}_1 = \mathbf{0}$.

Any solution of the equation $\mathbf{N} \cdot \mathbf{t} = \mathbf{0}^T$ is a *transition invariant*. For net N_2 :

$$\mathbf{N} \cdot \mathbf{t} = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 \\ 1 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{0}^T$$

Any non-negative solution points to firing sequences returning to the same state. For example, $\mathbf{t} = (t_1, t_2, t_3, t_4)^T = (3, 3, 0, 0)^T$ is the transition invariant showing that if we are able to execute t_1 and t_2 three times, we return to the initial state. Again this property is independent of the initial marking. Trace invariants can again be seen as an ‘algebraic overapproximation’ of all possible traces returning to the same state.

5 A Beautiful Subclass: Free-Choice Petri Nets

The three models shown in Figs. 1, 2 and 3 are all *free-choice Petri nets*. These nets satisfy the constraint that any two transitions having the same place as

input place should have identical sets of input places. Formally, for any two transitions $t1, t2 \in T$ such that $\bullet t1 \cap \bullet t2 \neq \emptyset$: $\bullet t1 = \bullet t2$. In Fig. 1, $\bullet t3 \cap \bullet t4 \neq \emptyset$, but $\bullet t3 = \bullet t4 = \{p3, p4\}$. The free-choice requirement implies that choice and synchronization are ‘separable’, i.e., choices are ‘free’ and not controlled by places that are not shared by all transitions involved in the choice. Free-choice Petri nets are very relevant for BPM, PM, and WFM, because most modeling languages have constructs (e.g., gateways in BPMN, control nodes in UML activity diagrams, and connectors in EPCs) modeling AND/XOR-splits/joins. As a result, choice (XOR-split) and synchronization (AND-join) are separated.

To exploit the properties of free-choice Petri nets, we often ‘short-circuit’ the accepting Petri net, i.e., we add a transition consuming tokens from the places in the final marking and producing tokens for the places in the initial marking. This implies that when reaching the final marking, it is possible to do a ‘reset’ and start again from the initial state.

We refer to [7] for the many results known for free-choice Petri nets, e.g., Commoner’s Theorem, the two Coverability Theorems, the Rank Theorem, the Synthesis Theorem, the Home Marking Theorem, the two Confluence Theorems, the Shortest Sequence Theorem, and the Blocking Marking Theorem.

6 Conclusion

This short paper should be considered as a ‘teaser’ for researchers and experts working on BPM, PM, and WFM. Although often not directly visible, many techniques and tools depend on Petri nets. See [5–7,9] to learn more about the Petri net theory. For the use of Petri nets in BPM, PM, and WFM, see [1–4].

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

References

1. van der Aalst, W.M.P.: Business process management: a comprehensive survey. *ISRN Softw. Eng.* 1–37 (2013). <https://doi.org/10.1155/2013/507984>
2. van der Aalst, W.M.P.: *Process Mining: Data Science in Action*. Springer, Berlin (2016). <https://doi.org/10.1007/978-3-662-49851-4>
3. van der Aalst, W.M.P., van Hee, K.M.: *Workflow Management: Models, Methods, and Systems*. MIT Press, Cambridge (2004)
4. van der Aalst, W.M.P., et al.: Soundness of workflow nets: classification, decidability, and analysis. *Formal Aspects Comput.* **23**(3), 333–363 (2011)
5. van der Aalst, W.M.P., Stahl, C.: *Modeling Business Processes: A Petri Net Oriented Approach*. MIT Press, Cambridge (2011)
6. Best, E., Wimmel, H.: Structure theory of petri nets. In: Jensen, K., van der Aalst, W.M.P., Balbo, G., Koutny, M., Wolf, K. (eds.) *Transactions on Petri Nets and Other Models of Concurrency VII. LNCS*, vol. 7480, pp. 162–224. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38143-0_5

7. Desel, J., Esparza, J.: Free Choice Petri Nets. Cambridge Tracts in Theoretical Computer Science, vol. 40. Cambridge University Press, Cambridge (1995)
8. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.: Fundamentals of Business Process Management. Springer, Berlin (2013). <https://doi.org/10.1007/978-3-642-33143-5>
9. Reisig, W., Rozenberg, G. (eds.): ACPN 1996. LNCS, vol. 1491. Springer, Heidelberg (1998). <https://doi.org/10.1007/3-540-65306-6>



Responsible Process Mining - A Data Quality Perspective

Moe Thandar Wynn¹(✉) and Shazia Sadiq²

¹ Queensland University of Technology, Brisbane, Australia
m.wynn@qut.edu.au

² The University of Queensland, Brisbane, Australia
shazia@itee.uq.edu.au

Abstract. Modern organisations consider data to be their lifeblood. The potential benefits of data-driven analyses include a better understanding of business performance and more-informed decision making for business growth. A key road block to this vision is the lack of transparency surrounding the quality of data. A process mining study that utilises low-quality, unrepresentative data as input has little or no value for the organisation and becomes a catalyst for erroneous conclusions ('Garbage-in-Garbage-out'). Many process mining techniques do not take into account inherent inaccuracies in the data, or how the data might have been manipulated or pre-processed. It is thus impossible to ascertain the degree to which analysis outcomes can be relied upon. This tutorial paper outlines foundational concepts of data quality with a special focus on typical data quality issues found in event data used for process mining analyses. Key challenges and possible approaches to tackle these data quality problems are elaborated on.

Keywords: Process mining · Data quality

1 Introduction

Process Mining is a specialised form of data-driven process analytics where data about process executions, collated from the different IT systems typically available in organisations, is analysed to uncover *the real behaviour and performance* of business operations [2]. Without question, the extent to which the outcomes from process mining analyses can be relied upon for insights is directly related to the quality of the input data. The onus is usually on a process analyst to identify, assess and appropriately remedy data quality issues so as to avoid inadvertently introducing errors into the data while minimising information loss. It is widely acknowledged that eighty percent of the work of data scientists is taken up by data preparation and handling data quality issues¹. The case of the process analyst is no different [17].

¹ <https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#58f51e5d6f63>.

There has been an increased interest in research investigating the issues of responsible data science [3,15]. Key dimensions in the notion of responsible data science (such as fairness, accuracy, transparency, and confidentiality [3]) are being explored and also for different domains (e.g., healthcare). In order to take steps towards responsible process mining, there is the dual need to increase the importance of data quality awareness and mitigate the opportunity to make erroneous conclusions, while helping process analysts overcome the burden of managing data quality.

In this tutorial paper, we focus on event logs as the primary form of input into process mining. Accordingly, we first present a brief summary of existing work on understanding data quality requirements for event logs. In the words of Edward Demming, the father of quality management, you can't manage what you can't measure. Hence, our next section outlines key techniques for measuring data quality in event logs. Finally, we provide a synopsis of current contributions and future needs of data quality awareness in process mining.

2 Understanding Data Quality Requirements for Event Logs

An event log used for process mining contains a collection of cases whereby each case can be seen as a sequence of events [2]. Each event refers to a case, an activity being undertaken, a point in time and a transaction type. An event may also refer to a resource or an organisational role and other data attributes (e.g., customer details and case outcomes).

The process mining manifesto [1] highlighted the need for high-quality event logs for process mining. The manifesto describes five levels of maturity ranging from one star to five stars. At the lowest level of maturity (*) where events are recorded manually, one may find that events that are incorrectly entered (e.g., incorrect timestamps or activity labels) or events may be missing. At the highest level of maturity (*****), event logs are considered to be complete and accurate as events are recorded automatically by a system (e.g., a process-aware information system). Most real-life event logs are found to be in-between these two extremes of the scale with many quality issues [6,17].

As most process mining techniques make use of key event data, namely, case identifiers, activity labels, and timestamps, missing, inaccurate or erroneous values (e.g., only a date is recorded but no time, incorrect spellings or variations in how activities are labeled) for any of this data may mean that a case or an event has to be filtered out or an erroneous value may need to be replaced, or a missing value may need to be inferred.

Given the diversity of data quality problems, it is important to understand the key requirements. While Juran and Godfrey [10] provide the fundamental "fitness for use" principle, decades of data quality research has proliferated various understandings of data quality requirements through its underlying dimensions [8,14,16,20]. Over the course of time, many of the definitions for different data quality dimensions have overlapped, and the same definitions for the same

dimensions have developed conflicting interpretations, resulting in a level of disparity that does not support a shared understanding. Recent work offers an empirically validated consolidation of these dimensions covering both academic and practitioner perspectives [9], and provides 33 dimensions clustered into eight categories, namely Completeness, Accuracy, Validity, Consistency, Currency, Availability and Accessibility, Reliability and Credibility, and Usability and Interpretability. These studies indicate that data quality requirements cover both objective (e.g. uniqueness and format consistency) as well as subjective (e.g. relevance and freshness) dimensions.

There have been efforts by process mining researchers to classify data quality issues typically found in event logs [6, 12, 17, 18] with a view to take steps towards addressing these issues and thus to increase the reliability of analysis results.

Bose et al. [6] identify four broad categories of issues affecting event log quality: missing data (where data items are not recorded in an event log), incorrect data (where data items are incorrectly recorded in an event log), imprecise data (where recorded values are considered too coarse to be useful) and irrelevant data (where data items contains irrelevant information). The authors also identify 27 classes of event log quality issues (e.g., problems related to timestamps in event logs, imprecise activity names, and missing events) depending on where they occur such as cases, events, activity labels, timestamps, resources. Their intention is to “encourage systematic logging approaches (to prevent event log issues), repair techniques (to alleviate event log issues) and analysis techniques (to deal with the manifestation of process characteristics in event logs)” [6]. These issues were illustrated from the analysis of five real-life event logs from different application domains.

Suriadi et al. [17] identify eleven event log imperfection patterns based on their experience with over 20 Australian industry data sets which confirm the severity of data quality issues in process data and their potential impact on process mining analyses. The eleven patterns include form-based event capture, inadvertent time travel, unanchored event, scattered event, elusive case, scattered case, collateral event, polluted label, distorted label, synonymous labels and homonymous label. Each pattern is described using the following components: description of the pattern, real-life example of the pattern, affect which captures the consequence of the occurrence of the pattern on process mining outcomes, the type of data event and event log entities affected by the pattern, strategy to detect the presence of a pattern, potential remedies and side-effects of these remedies, and indicative rules for detection.

Lu and Fahland [12] propose a conceptual framework to better understand event data quality for process mining analysis. The framework categorises event data into three entities: quality of events, quality of ordering of events and quality of labels of event. These three entities are then evaluated based on two dimensions: individual trustworthiness and global conclusiveness whereby individual trustworthiness focuses on the intrinsic qualities of event data (e.g., accuracy or correctness dimensions) while the global conclusiveness indicates if a significant pattern is being observed.

3 Measuring Data Quality of Event Logs

Data quality requirements continue to be dictated by the fitness for use principle [10], thus making them highly dependent on the use context. Further a plethora of diversified requirements (i.e. dimensions) exist, which are in turn deeply bound to use context making them complicated to model, analyse, and re-use, resulting in a prohibitive capacity to have a common set of measures for detecting and quantifying data quality.

Batini et al. [4] provide a comprehensive analysis of existing approaches for data quality assessment. We note that most, if not all, of these approaches follow a user centric approach where requirements are solicited from users before the data is explored (see e.g. [5, 11, 19]).

However, in the process mining context, access to the creators of data that constitutes event-logs cannot be relied on. This is mostly the case for publicly available event logs. Furthermore a process analyst cannot typically influence data capture practices and hence expectation of cleaning of the source data may be misplaced. Thus it is imperative to measure the quality of an event log respective to the particular type of analysis intended such as process discovery, performance analysis or conformance checking. For instance, the missing values metric assesses the fraction of the log for which a particular log attribute is populated which contributes to quantifying the Completeness dimension. In a log where the majority of events only have “complete” (rather than “start” and “complete”) timestamps, i.e. have a high degree of missing values, the suitability of that log for performance analysis is low while the suitability for process discovery may not be negatively affected. On the other hand, if recorded timestamps do not accurately reflect when an activity occurred, process discovery will be compromised.

In [18], the authors propose an extensible framework to measure event data quality based on twelve dimensions collated from prior literature and to quantify the prevalence of data quality issues in event data. They include completeness, uniqueness, timeliness, validity, accuracy/correctness, consistency, believability, credibility, relevancy, security/confidentiality, complexity, coherence, representation/format.

Another early advocate of detecting data quality issues in event logs is Anna Rozinat, the co-founder of Disco Process Mining Tool. Through a number of blog posts which have now been collated into a book on process mining in practice², various data quality issues in event logs and ways to detect and (potentially) repair them were discussed. The quality issues mentioned in the book include formatting errors, missing data (event, attribute values, case IDs, activities, timestamps, attribute history, timestamps for activity repetition) as well as zero timestamps, wrong timestamps, same timestamps for multiple activities and different timestamp granularity.

² <http://processminingbook.com>.

4 Data Quality Awareness in Process Mining

Keeping a detailed record of the origins of data and how data is transformed along the way will increase its traceability and trustworthiness. Where such information is unavailable, the extent and effect of changes on the data will be opaque to the analyst who, may view the data as ‘ground truth’, i.e. direct observations as opposed to already modified data. Such a view can result in inaccurate or misleading analysis results or inappropriate further transformations. For instance, where the analyst is unaware that a data set extracted from a hospital’s emergency department has been modified through time-shifting in order to de-identify patients (as in the case of MIMIC critical care data set³), using this data for performance analysis will lead to incorrect results.

There has been some work to detect and repair quality issues associated with event logs. For example, Dixit et al. [7] presents a user-guided technique to detect event ordering imperfection patterns in a log associated with timestamps and then repairing identified issues using user input. The timestamp related quality issues such as different granularities, order anomaly and statistical anomaly are detected and repaired. Similarly, Lu et al. [13] presents an interactive way to assist users explore data quality patterns of interest using the context information contained in an event log. Five measures to quantify the pervasiveness of a pattern in an event log are also proposed. They include the pattern support, pattern confidence, case support, case confidence and case coverage.

To date there has been little research aimed at developing a comprehensive framework to address the issue of incorrect analysis results from inadequate data quality of event logs. Lessons from prior work in quality awareness for database (e.g., [21]) indicate that there are at least three essential components of such frameworks, each of which presents a number of research challenges, namely (1) data quality profiling that builds on shared understanding of data quality dimensions and associated metrics, (2) user preference modelling that allows users analytic needs to be captured, and (3) visibility of quality profiles together with analysis (process mining) results to improve understanding of the impact of inadequate data quality. We invite process mining researchers to tackle these challenges to move towards responsible process mining with the aim to improve the credibility and trust of stakeholders in process mining results.

Acknowledgements. The authors would like to acknowledge the input from QUT researchers (Professor ter Hofstede, Dr Andrews, Dr Suriadi and Dr Poppe) who work on this topic. This work is partly supported by ARC Discovery Project DP190102141 on Building Crowd Sourced Data Curation Processes.

References

1. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_19

³ <https://mimic.physionet.org/>.

2. Van der Aalst, W.M.P.: *Process Mining: Data Science in Action*, 2nd edn. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4d>
3. van der Aalst, W.M.P., Bichler, M., Heinzl, A.: Responsible data science. *Bus. Inf. Syst. Eng.* **59**(5), 311–313 (2017)
4. Batini, C., Cappiello, C., Francalanci, C., Maurino, A.: Methodologies for data quality assessment and improvement. *ACM Comput. Surv.* **41**(3), 16 (2009)
5. Batini, C., Scannapieco, M.: *Data Quality: Concepts, Methodologies and Techniques*. Springer, Heidelberg (2006). <https://doi.org/10.1007/3-540-33173-5>
6. Bose, J.C., Mans, R., van der Aalst, W.M.P.: Wanna improve process mining results - it's high time we consider data quality issues seriously. In: *IEEE Symposium on Computational Intelligence and Data Mining*, pp. 127–134 (2013)
7. Dixit, P.M., et al.: Detection and interactive repair of event ordering imperfection in process logs. In: Krogstie, J., Reijers, H.A. (eds.) *CAiSE 2018*. LNCS, vol. 10816, pp. 274–290. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91563-0_17
8. Eppler, M.J.: *Managing Information Quality: Increasing the Value of Information in Knowledge-intensive Products and Processes*. Springer, Heidelberg (2006). <https://doi.org/10.1007/3-540-32225-6>
9. Jayawardene, V., Sadiq, S., Indulska, M.: The curse of dimensionality in data quality. In: *24th Australasian Conference on Information Systems (ACIS)*, pp. 1–12. RMIT University (2013)
10. Juran, J., Godfrey, A.: *Quality Handbook*. Republished McGraw-Hill, New York (1999)
11. Lee, Y.W., Strong, D.M., Kahn, B.K., Wang, R.Y.: AIMQ: a methodology for information quality assessment. *Inf. Manag.* **40**(2), 133–146 (2002)
12. Lu, X., Fahland, D.: A conceptual framework for understanding event data quality for behavior analysis. In: Kopp, O., Lenhard, J., Pautasso, C. (eds.) *Central European Workshop on Services and their Composition ZEUS*. *CEUR Workshop Proceedings*, vol. 1826, pp. 11–14 (2017)
13. Lu, X., et al.: Semi-supervised log pattern detection and exploration using event concurrence and contextual information. In: Panetto, H., et al. (eds.) *OTM 2017*. LNCS, vol. 10573, pp. 154–174. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69462-7_11
14. Scannapieco, M., Catarci, T.: Data quality under a computer science perspective. *Arch. Comput.* **2**, 1–15 (2002)
15. Srivastava, D., Scannapieco, M., Redman, T.C.: Ensuring high-quality private data for responsible data science: vision and challenges. *J. Data Inf. Qual.* **11**(1), 1:1–1:9 (2019)
16. Stvilia, B., Gasser, L., Twidale, M.B., Smith, L.C.: A framework for information quality assessment. *J. Am. Soc. Inform. Sci. Technol.* **58**(12), 1720–1733 (2007)
17. Suriadi, S., Andrews, R., ter Hofstede, A.H.M., Wynn, M.T.: Event log imperfection patterns for process mining: towards a systematic approach to cleaning event logs. *Inf. Syst.* **64**, 132–150 (2017)
18. Verhulst, R.: *Evaluating quality of event data within event logs: an extensible framework*. Master's thesis, Technische Universiteit Eindhoven, August 2016
19. Wang, R.Y.: A product perspective on total data quality management. *Commun. ACM* **41**(2), 58–65 (1998)
20. Wang, R.Y., Strong, D.M.: Beyond accuracy: what data quality means to data consumers. *J. Manag. Inf. Syst.* **12**(4), 5–33 (1996)
21. Yeganeh, N.K., Sadiq, S., Sharaf, M.A.: A framework for data quality aware query systems. *Inf. Syst.* **46**, 24–44 (2014)



IoT for BPMers. Challenges, Case Studies and Successful Applications

Francesco Leotta, Andrea Marrella, and Massimo Mecella^(✉)

Sapienza Università di Roma, Rome, Italy
{leotta,marrella,mecella}@diag.uniroma1.it

Abstract. The Internet-of-Things (IoT) refers to a network of connected and interacting devices (e.g., sensors, actuators) collecting and exchanging data over the Internet. In the last years, we have witnessed an increasing presence of IoT devices in scenarios of the Business Process Management (BPM) domain, which can strongly influence the coordination of the real-world entities (e.g., humans, robots) that execute specific tasks or entire business processes in such environments. While, on the one hand, the IoT can provide many opportunities for improving BPM initiatives, on the other hand, it poses challenges that require enhancements and extensions of the current state-of-the-art in BPM. This paper discusses how BPM can benefit from IoT, *(i)* showing which emerging challenges have to be tackled to integrate the IoT technology in a BPM project, and *(ii)* presenting concrete case studies on process adaptation and habit mining exploiting IoT and addressing the specific challenges posed by IoT itself.

Keywords: IoT · Business process · Habit mining · Process adaptation

1 Introduction

Business Process Management (BPM) is an active area of research based on the observation that each product and/or service that an organization offers, is the outcome of a number of performed activities. Business processes (BPs) are the key instrument for organizing such activities and improving the understanding of their interrelationships. Nowadays, BPs are enacted in many complex industrial (e.g., manufacturing, logistics, retail) and non-industrial (e.g., emergency management, healthcare, smart environments) domains [10]. In all these domains, we have witnessed an increasing presence of Internet-of-Things (IoT) devices (e.g., sensors, RFIDs, video cameras, actuators) that operate over the existing network infrastructure, including the Internet, to collect data from the physical environment, monitor in detail the evolution of several real-world objects of interest, and actuate concrete feedbacks (e.g., in the form of suggestions or alerts) in response to the observed information. From a BPM perspective, the knowledge extracted from the physical environment by IoT devices allows to

depict the contingencies and the context in which BPs are carried out, providing a fine-grained monitoring, mining, and decision support for them.

The interplay of IoT devices with BPM can provide many opportunities for improving the enactment of BPs. For example, among the main benefits, the execution of BPs can be driven by event data detected at real-time, enabling BPs to become more *adaptive* and *reactive* to what is happening in the real world. However, on the other hand, there is a conflict between the stability and meaningfulness of the services at work in a BP as opposed to the dynamic and changing environment that IoT is able to offer. This poses several challenges to concretely interconnect the two worlds and make them interact, which require enhancements and extensions to the current state-of-the-art in BPM.

According to the BPM-Meet-IoT manifesto [4], sixteen challenges have been identified to make this vision a reality. In this contribution, we focus on two specific areas from the BPM literature where it is strongly required to tackle these challenges, namely *habit mining* and *process adaptation*.

In Sect. 2, we show how data collected by IoT devices (a.k.a. IoT data) should be properly abstracted and managed when willing to employ BP discovery techniques [1] to model human habits as “personal processes”. Secondly, in Sect. 3, we present a reference conceptual architecture to build a BPM engine that is able to reason over the discrete counterpart of the “continuous” IoT data for achieving automated adaptation of running BPs in case of unanticipated exceptions. Finally, in Sect. 4, we conclude the paper.

2 Visual Process Maps for Habit Mining

A smart space (e.g., a smart house) represents a typical example of IoT environment. The aim of a smart space is providing people with automatic or semi-automatic services realizing the concept of ambient intelligence (AmI). To this aim, a set of both software and hardware networked artefacts, acting as sensors (e.g. presence, temperature sensors) or actuators (e.g., ovens, rolling shutters), are coordinated according to a previously acquired knowledge expressed in the form of models representing human preferences and environmental dynamics.

Models in smart spaces are usually classified as *specification-based*, which are hand-made by experts, or *learning-based*, which are instead obtained by applying machine learning and data mining. In the first case, models are usually based on logic formalisms, relatively easy to read and validate (once the formalism is known to the reader), but their creation requires a major cost in terms of expert time and effort. In the latter case, the model is automatically learned from a training set (whose labeling cost may vary according to the proposed solution) but employed formalisms are usually not “explainable” due to the statistic techniques they are based on, making them less immediate to understand.

Authors in [5] suggested that applying methods originally taken from BPM to human habits may represent a compromise between specification-based and learning-based methods, provided that the gap between raw sensor measurements and human actions can be filled in by performing a log-preprocessing

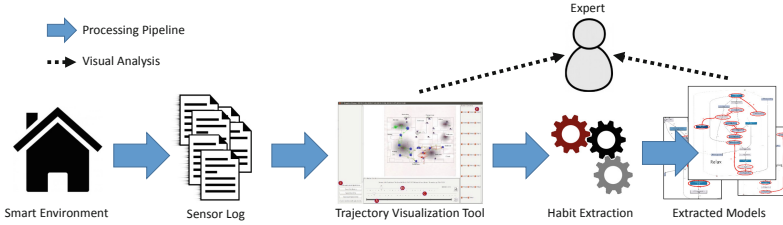


Fig. 1. The conceptual architecture of the Visual Process Maps (VPM) system.

step. Such a step may consist of simple inferences on data, or complex machine learning algorithms. On the line of this argument, [2,6] propose the Visual Process Maps (VPM) system, consisting of a complete pipeline formed by (i) a tool for the visual analysis of sensor logs, (ii) a method to transform raw movement measurements into actions, and (iii) a method to identify and visually analyze precedence relationships between human actions through the employment of fuzzy mining [3].

Figure 1 shows the conceptual architecture of VPM. A smart space produces, during runtime, a *sensor log* containing raw measurements from available sensors. Measurements can be produced by a sensor on a periodic base (e.g., temperature) or whenever a particular event is detected (e.g., a door opening). The current version of VPM focuses on sensor logs produced by a grid of Passive InfraRed (PIR) sensors triggering upon the detection of an object entering their field of view and automatically reset after a fixed amount of time since the last detected movement. The detection area of a PIR can be usually tuned to cover different area sizes ranging from a tile on the floor to an entire room.

The first step of VPM consists in a visual analysis tool, named Trajectory Visualization Tool, able to “play” specific portions of a log, perform automatic analysis tasks and visualize the result. Here, playing means to animate the sensor log showing the trajectory followed by a person in the house. The tool also allows to produce an *event log* obtained from the sensor log by aggregating simple PIR sensor measurements into sub-trajectories representing movement actions belonging to the following categories: (a) moving between areas of the house, (b) staying still under a PIR, or (c) moving in a specific area of the house.

Such an event log can be used as input for fuzzy mining. The rationale here is that, if we know the location of devices that humans can interact with inside the space, we can associate to each of these movement actions, the physical actions performed by humans (e.g., using the oven). As an example, if the model contains a precedence relation between the action “moving inside the bathroom” and “stay under the PIR sensor corresponding to the bed”, we have a clear idea of the human actions determined by the movement actions.

The process extracted by fuzzy mining depends on how the sensor log fed into VPM is labeled. If no label is available, the mined process model will represent the daily habit of a person. If instead, labels corresponding to the beginning and

the end of daily routines are available, it is possible to obtain specialized process models for them.

3 A Conceptual Architecture for Process Adaptation

During the enactment of BPs in IoT-based environments, variations or divergence from structured reference models are common due to exceptional circumstances arising in form of *exogenous events*, thus requiring the ability to properly *adapt* the process behavior. *Process adaptation* can be seen as the ability of a BP to react to exceptional circumstances (that may or may not be anticipated) and to adapt/modify its structure accordingly.

Since in IoT-based environments the number of possible anticipated exceptions is often too large, manual implementation of exception handlers at design-time is not feasible, since it is required to anticipate all potential problems and ways to overcome them in advance directly in the BP [10]. Furthermore, in such environments, many unanticipated exogenous events may arise during the BP execution, and the needed knowledge to tackle such events at the outset is often missing. Finally, a BPM engine can only reason over a discrete knowledge of the world, thus requiring to convert the continuous raw data collected by the IoT technology into discrete information.

To tackle this issue, we summarize the main ideas discussed in [8,9] and we introduce our architectural solution to build a BPM engine that is able to automatically adapt BPs at run-time when *unanticipated exceptions* occur in IoT-based environments, thus requiring no specification of recovery policies at design-time. The general idea builds on the dualism between an *expected reality* and a *physical reality*: process execution steps and exogenous events have an impact on the physical reality and any deviation from the expected reality results in a mismatch to be removed to allow process progression. As shown in Fig. 2, we identified 5 main architectural layers that we present in a bottom-up fashion.

The *cyber-physical layer* consists mainly of two classes of physical components: (i) sensors (such as GPS receivers, RFIDs, 3D scanners, cameras, etc.) that collect data from the physical environment by monitoring real-world objects and (ii) actuators (robotic arms, 3D printers, electric pistons, etc.), whose effects affect the state of the physical environment. The cyber-physical layer is also in charge of providing a physical-to-digital interface, which is used to transform *raw* data collected by the sensors into machine-readable events, and to convert *high-level* commands sent by the upper layers into *raw* instructions readable by the actuators. The cyber-physical layer does not provide any intelligent mechanism neither to clean, analyse or correlate data, nor to compose high-level commands into more complex ones; such tasks are in charge of the upper layer.

On top of the cyber-physical layer lies the *service layer*, which contains the set of services offered by the real-world entities (software, robots, agents, humans, etc.) to perform specific BP activities. In the service layer, available data can be aggregated and correlated, and high-level commands can be orchestrated to provide higher abstractions to the upper layers. For example, a smartphone

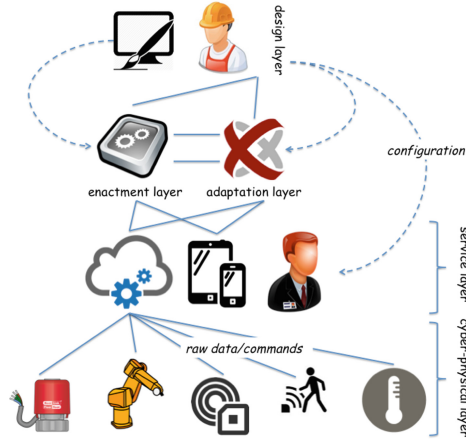


Fig. 2. A conceptual architecture for BPs enacted in IoT-based environments.

equipped with an application allowing to sense the position and the posture of a user is at this layer, as it collects the raw GPS, accelerometer and motion sensor data and correlates them to provide discrete and meaningful information.

On top of the service layer, there are two further layers interacting with each other. The *enactment layer* is in charge of (i) enacting complex BPs by deciding which activities are enabled for execution, (ii) orchestrating the different available services to perform those activities and (iii) providing an execution monitor to detect the anomalous situations that can possibly prevent the correct execution of BP instances. The execution monitor is responsible for deciding if process adaptation is required. If this is the case, the *adaptation layer* will provide the required algorithms to (i) reason over the available BP activities and contextual data and to (ii) find a recovery procedure for adapting the BP instance under consideration, i.e., to re-align the BP to its expected behaviour. Once a recovery procedure has been synthesized, it is passed back to the enactment layer for being executed.

Finally, the *design layer* provides a GUI-based tool to define new BP specifications. A BP designer must be allowed not only to build the BP control flow, but also to explicitly formalize the data reflecting the contextual knowledge of the IoT-based environment under study. It is important to underline that data formalization must be performed without any knowledge of the internal working of the physical components that collect/affect data in the cyber-physical layer. To link activities to contextual data, which are the main driver for triggering process adaptation, the GUI-based tool must go beyond the classical “activity model” as known in the literature, by allowing the BP designer to explicitly state what data may constrain an activity execution or may be affected after an activity completion or an exogenous event. Finally, besides specifying the BP, configuration files should also be produced to properly configure the enactment, the services and the sensors/actuators in the bottom layers.

The SMARTPM system presented in [7] is a concrete instantiation of the above reference architecture.

4 Concluding Remarks

This paper provides an introduction to the IoT with the eyes of a BPM researcher. The focus is on identifying and presenting those IoT features that directly impact BPM, i.e., *data*, *quality* and *granularity of such data*, *events*, *identification of process instances*, etc. In particular, we have focused on the issue of dealing with continuous and frequent data readings, and on the low level of abstraction provided by IoT measurements wrt. the traditional concept of “events” and “traces” in the BPM literature. Through two specific outcomes of our research activities, we have exemplified the above concepts in order to provide insights for further research.

Acknowledgments. The multi-year research work has been partly supported by the “Dipartimento di Eccellenza” grant, the H2020 RISE project FIRST (grant #734599), the H2020 ERC project NOTAE (grant #786572), the Sapienza grants IT-SHIRT, ROCKET and METRICS, the Lazio regional initiative “Centro di eccellenza DTC Lazio” and the project ARCA.

References

1. Augusto, A., et al.: Automated discovery of process models from event logs: review and benchmark. *TKDE* **31**(4), 686–705 (2019)
2. Dimaggio, M., Leotta, F., Mecella, M., Sora, D.: Process-based habit mining: experiments and techniques. In: 2016 International IEEE Conferences on Ubiquitous Intelligence and Computing (UIC), pp. 145–152. IEEE (2016)
3. Günther, C.W., van der Aalst, W.M.P.: Fuzzy mining – adaptive process simplification based on multi-perspective metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 328–343. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_24
4. Janiesch, C., et al.: The internet-of-things meets business process management: mutual benefits and challenges. arXiv preprint [arXiv:1709.03628](https://arxiv.org/abs/1709.03628) (2017)
5. Leotta, F., Mecella, M., Mendling, J.: Applying process mining to smart spaces: perspectives and research challenges. In: Persson, A., Stirna, J. (eds.) *CAiSE 2015*. LNBIP, vol. 215, pp. 298–304. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19243-7_28
6. Leotta, F., Mecella, M., Sora, D.: Visual process maps: a visualization tool for discovering habits in smart homes. *J. Ambient Intell. Humaniz. Comput.* 1–29 (2019). <https://link.springer.com/article/10.1007/s12652-019-01211-7>
7. Marrella, A., Halapuu, P., Mecella, M., Sardiña, S.: SmartPM: an adaptive process management system for executing processes in cyber-physical domains. In: *BPM Demo Track 2015*. CEUR-WS.org (2015). <http://ceur-ws.org/Vol-1418/paper24.pdf>
8. Marrella, A., Mecella, M., Sardina, S.: Intelligent process adaptation in the SmartPM system. *ACM Trans. Intell. Syst. Technol.* **8**(2), 1–43 (2016)

9. Marrella, A., Mecella, M., Sardiña, S.: Supporting adaptiveness of cyber-physical processes through action-based formalisms. *AI Commun.* **31**(1), 47–74 (2018)
10. Reichert, M., Weber, B.: *Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies*. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-3-642-30409-5>



Exploring Explorative BPM - Setting the Ground for Future Research

Thomas Grisold^{2(✉)}, Steven Gross^{3(✉)}, Maximilian Röglinger^{1(✉)},
Katharina Stelzl^{1(✉)}, and Jan vom Brocke^{2(✉)}

¹ University of Bayreuth, 95444 Bayreuth, Germany
{maximilian.roeglinger,

katharina.stelzl}@uni-bayreuth.de

² University of Liechtenstein, 9490 Vaduz, Liechtenstein

{thomas.grisold, jan.vom.brocke}@uni.li

³ Vienna University of Economics and Business, 1020 Vienna, Austria
steven.gross@wu.ac.at

Abstract. Recent claims in the literature highlight that BPM should become more explorative and opportunity-driven. The underlying argument is that BPM has been mainly concerned with exploitation activities – i.e., analysis and improvement of existing business processes – but it has neglected the role of innovation. In this conceptual article, we aim to establish a systematic understanding of what explorative BPM is and how it can be brought about. We pursue three goals. First, we derive an overarching definition of explorative BPM. Second, we propose the “triple diamond model” as a means to integrate explorative BPM activities in business process work. Third, we point to future research opportunities in the context of explorative BPM.

Keywords: Explorative BPM · Process innovation · Opportunity identification

1 Introduction

Recent claims highlight that BPM should become more innovation-driven [15, 29, 30]. Underlying these claims is the observation that BPM has made considerable progress to increase efficiency and effectiveness of business process work but has neglected the question of how organizations can facilitate innovation. That said, it is remarkable that Michael Hammer, founder of the Business Process Reengineering discipline and perhaps the most influential promoter of the process paradigm, prominently stated in one of his last articles that BPM has two primary intellectual antecedents: the quality movement and business process engineering. The first focusing on improving existing processes and the second focusing on rethinking processes using new technology aiming at innovation [11]. To some extent, this element of innovation, which has been at the core of BPM, seems to have faded into the background over the past decade. Moreover, digital technology today drives innovation at such speed and scale [3] that it provides new opportunities and challenges for BPM. What is subsumed under the term *explorative BPM* suggests that process-oriented organizations should be able to develop new capabilities and competences to detect emerging opportunities in terms of

new technologies and business models [22]. In light of today's highly dynamic business environment, this is considered essential to ensure customer satisfaction and foster inter- and intra-organizational collaboration [8]. Research on explorative BPM is still a new frontier and innovation and opportunity-driven process redesign are, to some extent, contradictory to the logics of what has been researched and practiced referred to as BPM in the recent past. Following Benner and Tushman [1], BPM helps organizations to achieve error reduction and variance control but, due to increasing standardization, it decreases an organization's capability to sense, seize, and transform ground-breaking innovations. How can exploration be integrated into business process work? This is a question that should deserve more attention in the literature [8, 15, 22].

Within the tutorial presented at BPM 2019, we aim to approach explorative BPM in a systematic way and develop the grounds for future research agendas. In particular, we will (1) point to related (management) disciplines, (2) review the most important features of explorative BPM, (3) introduce an integrated framework to realize explorative BPM in organizations, and (4) derive a research agenda.

2 Related Fields and Agendas

Following Rosemann's call to "mix up relevant communities" when studying explorative BPM [17, p. 637], we draw on a number of related fields and research agendas in the domains of management and organizational science. We draw on research on organizational ambidexterity (OA) and innovation management (IM) as both fields seem promising to extend our understanding of explorative BPM.

OA has been a buzzword in the management literature for more than twenty years, promising organizations long-term survival in turbulent environments [24, 27]. In essence, OA is an organization's dual capability to develop management capabilities for both exploitation, i.e., incremental innovation leading to short-term efficiency gains, and exploration, i.e., radical long-term innovation activities [12, 20, 24]. In detail, *exploration* is characterized as a consistent opportunity-seeking approach with a focus on long-term growth through the development and introduction of radical innovation in line with external demands [12, 21]. It requires organizations to develop adaptive processes and structures that can be (re-) configured on demand [20] and a culture marked as risk-taking, flexible, and fast, supported by a visionary leadership style. Moreover, OA research investigates how to put OA into practice, e.g. by systematically developing OA capabilities based on an OA maturity model or by using a decision model that assists organizations in selecting and scheduling exploration and exploitation projects to become ambidextrous in an economically reasonable manner [18, 23].

IM encourages the development and introduction of new products, services, business models, and processes by frequently following an outside-in approach that is guided by customers' needs [28]. Specifically, process innovation describes the "implementation of a new or significantly improved production or delivery method [i.e., process]. This includes significant changes in techniques, equipment and/or software" (OECD 2005, p. 49). Hence, process innovation comprises the effective redesign (and efficiency or effectiveness improvement) of existing processes but also the development of new ones by following a structured procedure [6, 14]. Process innovation can further

be split into ‘technological process innovation’ and ‘organizational process innovation’ [9, 11], ranging from single process elements (e.g. activities) to whole process chains [14], and it can be described as incremental or radical [25, 28]. Triggers for (process) innovation can be push and pull factors, where the former is based on the demand of internal or external customers while the latter is brought about by new technologies [28]. Besides the definition of IM, researchers and practitioners have developed a large set of methods and tools to generate new ideas for innovation, and have brought them into practice for an organizational competitive advantage [25]. All in all, descriptive and prescriptive knowledge on OA and IM appears to be a promising theoretical lens for closing the theoretical and methodological gap in explorative BPM.

3 Conceptualizing Explorative BPM

While there is no explicit definition of explorative BPM, authors highlight a number of features that are commonly associated with it.

First and foremost, explorative BPM emerged as a complementary view to established understandings of BPM. Over the past decades, the main focus of BPM has shifted towards an exploitative management discipline that pursues “reaction-based” improvements when negative deviance and changes occur. From this perspective, BPM has been driven by an “inside-out” logic, ensuring the efficiency and effectiveness of processes. Innovation, on this view, occurs when organizations face a pressure to do so. This implies that innovation is incremental as it relies on existing resources and capabilities to detect and correct for undesired deviance [2].

Explorative BPM, in contrast, calls for an “outside-in” logic that applies “environmental scanning capabilities” to identify new opportunities in dynamic business environments [22, p. 3]. It is transformational in the sense that it develops new capabilities, identifies new business opportunities, and capitalizes on emerging technologies [19]. While new opportunities and emerging technologies might be novel and unfamiliar, organizations should be able to recognize and evaluate them in terms of their affordances and underlying potentials [3, 17]. Even if organizations do not face any threat or crisis, they are able to rethink or unlearn their established practices and approaches [16] to integrate innovations into their business process work.

When organizations capitalize on new business opportunities such as emerging technologies or new customer expectations, explorative BPM can transform business processes in three different ways leading to different value propositions for customers. For a better overview, Fig. 1 differentiates between three dimensions of process innovation and its typical combinations for explorative and exploitative BPM. Focusing on explorative BPM, in the first and second case, new opportunities can lead to a reengineering of existing processes that, in turn, entail the same or an enhanced value proposition for customers [4]. For example, leveraging artificial intelligence for detecting fraudulent insurance claims innovates the process from the company’s internal perspective, while the value proposition for the consumer remains the same. An enhanced value proposition may result from the integration of customers’ connected devices into the insurance premium calculation, especially for cost-conscious customers. In the third case, new opportunities may enable the creation of entirely new

Three dimensions of BPM			
Trigger	Problem-driven		Opportunity-driven
Action	Improve existing process	Reengineer existing process	Create new process
Value proposition	Same value proposition	Enhanced value proposition	New value proposition

Typical combinations for explorative and exploitative BPM	
Exploration	Opportunity + Reengineer existing process + Same value proposition
	Opportunity + Reengineer existing process + Enhanced value proposition
	Opportunity + Create new process + New value proposition
Exploitation	Problem + Improve existing process + Same value proposition
	Problem + Reengineer existing process + Same value proposition
	Problem + Improve existing process + Enhanced value proposition
	Problem + Reengineer existing process + Enhanced value proposition

Fig. 1. The three dimensions of process innovation and its typical combinations for explorative and exploitative BPM

processes resulting in a new value proposition for customers [4]. For example, building on smart contracts enables insurances to provide a flight delay coverage without the required customer trust, as the involved parties may even remain anonymous and the contract executes itself. This clearly differs from traditional insurance contracting where the insurance payout depends on and is initiated by the insurance company.

A third important feature is that explorative BPM activities must not be understood as selective and single interventions but require continuity [32]. Explorative BPM needs ongoing visioning [22] and experimenting [8] to develop new resources and capabilities [19]. Establishing continuity in terms of exploration is a challenge for process-oriented organizations. According to [1, 2], BPM, while reducing errors and variation, leads to inertia in the long run. Organizations become path-dependent in the sense that they draw on existing capabilities, resources, and best practices; those practices that helped organizations to achieve efficiency in the first place can now “impede an organization’s adaptation to major technological transitions” [2, p. 323]. In order to ensure continuity in terms of explorative BPM, organizations can, for example, implement “opportunity points” in business processes to integrate new technologies, or establish teams and units that are concerned with identifying new trends and opportunities and mapping them to organizational work [2, 22]. Central to these interventions is that explorative BPM is contingent on capabilities to continuously sense, seize and transform opportunities into innovations.

To summarize, based on the features commonly associated with explorative BPM, we understand it as *the continuous process of questioning underlying business logics – i.e. the established understanding how value is generated – and integrating innovation opportunities (both in terms of business and technology) into business process work, even if there is no perceived pressure to do so. Explorative BPM refers to the offering of the same, enhanced, or new value propositions through the reengineering of existing processes or the creation of new processes.*

4 Realizing Explorative BPM: Towards an Integrated Model

Having outlined the features and characteristics of explorative BPM, we now introduce an integrated framework to realize explorative BPM in organizations. Figure 2 depicts the so-called *Triple Diamond Model*. This model builds on the concept of divergent and convergent thinking [5] to integrate innovation opportunities. Within divergent thinking, novelty is created through a creative process. Taking the information at hand, multiple alternative solutions are created through unexpected combinations, a transformation of information into unexpected forms and other techniques [5]. Convergent thinking evaluates a solution according to criteria such as speed, accuracy, or logic [5].

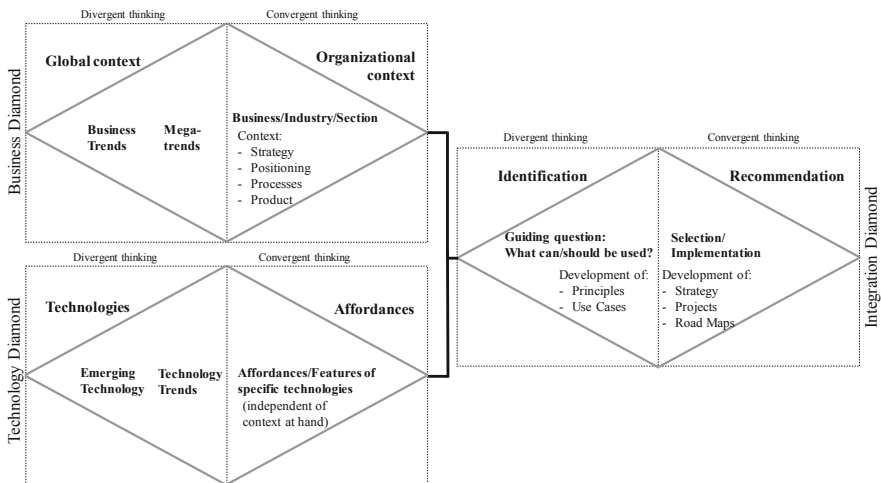


Fig. 2. The “Triple Diamond Model” to foster explorative BPM

This process of divergent exploration and convergent evaluation is the core of the three diamonds. These diamonds represent the business perspective, the technology perspective, and the integration perspective. It is not meant to be understood as a rigid method, but rather as a blueprint which forms the explorative BPM process. This also means that there is no strict order within its elements. The model is context-aware, i.e. it takes into consideration the specific situation of the organization at hand [7, 31].

In the business diamond, current local and global trends are identified. This may be a shift of consumer expectations towards higher data control, greater personalization of products and services, or an orientation towards subscription-based models, to name a few examples. The identification and exploration of these trends is part of the divergent thinking phase, in which new ideas are generated regardless of their feasibility or limitations given. Subsequently, in the convergent thinking phase, the context of the organization is considered. This includes the industry, its strategy, positioning, current products and processes. The previously identified and generated alternatives are thereby evaluated regarding criteria like applicability, feasibility, and fit to the current strategy.

The technology diamond identifies current technological trends and innovations. Emerging technical solutions have been found to be a promising source of business process innovation [8, 13]. In the divergent thinking phase, it is first explored which technological trends are available and how these technologies can potentially influence future process interactions and executions [3]. This boundless exploration of emerging technologies is followed by the convergent thinking phase. In this stadium, the identified opportunities are evaluated regarding their associated risks, technological affordances, and features of the specific technologies. In this phase, the evaluation of the technology is yet context-independent of a specific application scenario.

Once both business and technological trends have been identified and evaluated, the integration diamond combines both sides with the organizational process perspective. The development of use cases takes part in the divergent thinking phase. These use cases should be identified and developed independent from potential organizational limitations. The guiding question in this phase is: Which technological and business trends can be leveraged in the organization at hand? This supports the organization to generate new alternative ways how value is generated. The evaluation of these use cases takes part in the convergent thinking phase of the integration diamond. There, the generated alternatives are assessed and selected. At the end of this phase, concrete results are generated. This includes but is not limited to a clear strategy for the implementation and concrete projects and roadmaps for operationalizing the new business process idea. By doing so, either existing processes are majorly modified, new processes are established, or new outcomes are generated within the organization.

5 Conclusion and Research Agenda

The management of business processes in research and practice has much focused on process-driven operational excellence. Current trends in the area like process mining, which has received great attention beyond the BPM community, emphasize this course by targeting and analyzing existing processes for identifying operational weaknesses. This paper aims at expanding this exploitative view on BPM by an integration of innovation opportunities into organizational business processes – called explorative BPM.

In the course of this paper, we defined explorative BPM and its characteristics. While radically re-thinking existing business processes has been aimed for over three decades, it is only a part of what we describe as explorative BPM. Compared to the traditional exploitative perspective, it is opportunity-driven and includes the creation of new processes offering new value propositions. Thus, rather than focusing solely on

existing processes, explorative BPM also takes into consideration novel ways to extend the organizations' existing process landscape. The exploration of new opportunities is also a continuous endeavor rather than a one-time process. We also proposed an integrated model, which incorporates the different dimensions of explorative BPM into one consistent framework.

We see explorative BPM as a promising research stream which has the potential to reshape existing paradigms and self-perceptions in BPM research and practice. We acknowledge that, despite existing research which aims to define and characterize explorative BPM, more research is needed to understand its nature and integration into organizational practices. Hence, we want to point to the following future research streams.

1. *Defining and evaluating methods which put explorative BPM into practice.* A plethora of methods and management approaches exist which aim to incrementally improve existing processes [7, 10]. However, more research is needed to design procedures that help implement explorative BPM in a step-wise manner [29]. To evaluate these methods, appropriate criteria have to be defined to ensure the applicability of these methods and the validity of their results.
2. *Investigating which organizational capabilities foster explorative BPM.* Despite the methodological view on the topic, other organizational capabilities such as culture and governance are likely to affect the successful integration of explorative BPM into practice. We see explorative BPM as a holistic approach [30], within the organizational context, thus investigating required capabilities that affect the successful integration of the approach is of high interest [15].
3. *Expanding the theoretical foundation by other research streams.* Although explorative BPM specifically targets business processes within organizations, relevant insights from the IM, OA, and organization science discipline and related fields should be used to complement the theoretical foundation by integrating relevant concepts and theories. For practical and historical reasons, research has often been conducted within separate communities. However, explorative BPM should build on and integrate innovation research from different disciplines like innovation-, design-, and organizational research, to name but a few.

We believe the BPM community has the chance to complement its existing mature understanding of methods, techniques, and tools for operational excellence by an explorative perspective. We want to emphasize that these are two sides of the same coin. It is not a shift but an expansion of the current understanding of BPM and thus beneficial for the community as a whole.

References

1. Benner, M.J., Tushman, M.L.: Exploitation, exploration, and process management: the productivity dilemma revisited. *Acad. Manag. Rev.* **28**(2), 238–256 (2003)
2. Benner, M.J., Tushman, M.L.: Process management, technological innovation, and organizational adaptation. In: Grover, V., Markus, M.L. (eds.) *Business Process Transformation*, pp. 317–326. M.E. Sharpe, Irvine (2007)

3. Berger, S., Denner, M.S., Röglinger, M.: The nature of digital technologies: development of a multi-layer taxonomy. In: 26th European Conference on Information Systems (ECIS), Portsmouth, UK, 2018, p. 96 (2018)
4. Christensen, C.M., Hall, T., Dillon, K., Duncan, D.S.: Know your customers' "jobs to be done". *Harvard Bus. Rev.* **94**, 54–62 (2016)
5. Cropley, A.: In praise of convergent thinking. *Creat. Res. J.* **18**(3), 391–404 (2006)
6. Davenport, T.H.: *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press, Boston (1992)
7. Denner, M.-S., Röglinger, M., Schmiedel, T., Stelzl, K., Wehking, C.: How context-aware are extant BPM methods? - development of an assessment scheme. In: Weske, M., Montali, M., Weber, L., vom Brocke, J. (eds.) *BPM 2018. LNCS*, vol. 11080, pp. 480–495. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_28
8. Dezi, L., Santoro, G., Gabteni, H., Pellicelli, A.C.: The role of big data in shaping ambidextrous business process management: case studies from the service industry. *Bus. Process. Manag. J.* **24**(5), 1163–1175 (2018)
9. Edquist, C., Hommen, L., McKelvey, M.: *Innovation and Employment: Process Versus Product Innovation*. Edward Elgar Publishing, Cheltenham (2001)
10. Gross, S., Malinova, M., Mendling, J.: Navigating through the maze of business process change methods. In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*, pp. 6270–6279 (2019)
11. Hammer, M., Champy, J.: *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Business Books, New York (1993)
12. He, Z.-L., Wong, P.-K.: Exploration vs. exploitation: an empirical test of the ambidexterity hypothesis. *Organ. Sci.* **15**(4), 481–494 (2004)
13. Kemsley, S.: Emerging technologies in BPM. In: vom Brocke, J., Schmiedel, T. (eds.) *BPM - Driving Innovation in a Digital World*. MP, pp. 51–58. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-14430-6_4
14. Kern, E.-M., Röser, T., Ulrich, S.: Prozessmanagement: Trigger Und Befähiger Für Prozessinnovation? In: Mieke, C. (ed.) *Prozessinnovation Und Prozessmanagement. Zwei Managementfelder Zur Stärkung Der Prozessleistung in Unternehmen*, pp. 1–26. Logos, Berlin (2013)
15. Kerpedzhiev, G., König, U., Röglinger, M., Rosemann, M.: Business process management in the digital age. *BPT Trends* (2017)
16. Klammer, A., Grisold, T., Guldenberg, S.: Introducing a 'stop-doing' culture: how to free your organization from rigidity. *Business Horizons* (in press)
17. Kohlborn, T., Mueller, O., Poepplbuss, J., Roeglinger, M.: Interview with Michael Rosemann on ambidextrous business process management. *Bus. Process. Manag. J.* **20**(4), 634–663 (2014)
18. Linhart, A., Röglinger, M., Stelzl, K.: A project portfolio management approach to tackling the exploration/exploitation trade-off. *Bus. Inf. Syst. Eng.* 1–17 (2018)
19. Ohlsson, J., Han, S., Bouwman, H.: The prioritization and categorization method (PCM) process evaluation at Ericsson: a case study. *Bus. Process. Manag. J.* **23**(2), 377–398 (2017)
20. O'Reilly, C.A., Tushman, M.L.: The ambidextrous organization. *Harvard Bus. Rev.* **82**(4), 74–81 (2004)
21. O'Reilly, C.A., Tushman, M.L.: Organizational ambidexterity: past, present, and future. *Acad. Manag. Perspect.* **27**(4), 324–338 (2013)
22. Rosemann, M.: Proposals for future BPM research directions. In: Ouyang, C., Jung, J.-Y. (eds.) *AP-BPM 2014. LNBP*, vol. 181, pp. 1–15. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08222-6_1

23. Röglinger, M., Schwindenhammer, L., Stelzl, K.: How to put organizational ambidexterity into practice – towards a maturity model. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNBIP, vol. 329, pp. 194–210. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98651-7_12
24. Teece, D.J.: Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. *Strateg. Manag. J.* **28**(13), 1319–1350 (2007)
25. Tidd, J., Bessant, J.: *Managing Innovation: Integrating Technological, Market and Organizational Change*, 5th edn. Wiley, Chichester (2013)
26. Tumbas, S., Berente, N., vom Brocke, J.: Three types of chief digital officers and the reasons organizations adopt the role. *MISQ Exec.* **16**(2), 121–134 (2017)
27. Tushman, M.L., O’Reilly, C.A.: Ambidextrous organizations: managing evolutionary and revolutionary change. *Calif. Manag. Rev.* **38**(4), 8–30 (1996)
28. Vahs, D., Brem, A.: *Innovationsmanagement: Von Der Idee Zur Erfolgreichen Vermarktung*, 4th edn. Schäffer-Poeschel, Stuttgart (2013)
29. vom Brocke, J., Mendling, J.: Frameworks for business process management: a taxonomy for business process management cases. In: vom Brocke, J., Mendling, J. (eds.) *Business Process Management Cases*. MP, pp. 1–17. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-58307-5_1
30. vom Brocke, J., Rosemann, M. (eds.): *Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture*. Springer, Heidelberg (2014). <https://doi.org/10.1007/978-3-642-01982-1>
31. vom Brocke, J., Zelt, S., Schmiedel, T.: On the role of context in business process management. *Int. J. Inf. Manag.* **36**, 486–495 (2016)
32. vom Brocke, J., Schmiedel, T., Recker, J., Trkman, P., Mertens, W., Viaene, S.: Ten principles of good business process management. *Bus. Process. Manag. J.* **20**(4), 530–548 (2014)

Foundations



Dynamic Reconfiguration of Business Processes

Leandro Nahabedian¹(✉), Victor Braberman¹, Nicolás D'ippolito¹,
Jeff Kramer², and Sebastián Uchitel^{1,2}

¹ Universidad de Buenos Aires/CONICET, Buenos Aires, Argentina
{lnahabedian,vbraber,ndippolito,suchitel}@dc.uba.ar

² Department of Computing, Imperial College, London, UK
j.kramer@imperial.ac.uk

Abstract. Organisations require that their business processes reflect their evolving practices by maintaining compliance with their policies, strategies and regulations. Designing workflows which satisfy these requirements is complex and error-prone. Business process reconfiguration is even more challenging as not only a new workflow must be devised but also an understanding of how the transition between the old and new workflow must be managed. Transition requirements can include both domain independent, such as delayed and immediate change, or user-defined domain specific requirements. In this paper we present a fully automated technique which uses control synthesis to not only produce correct-by-construction workflows from business process requirements but also to compute a reconfiguration process that guarantees the evolution from an old workflow to a new one while satisfying any user-defined transition requirements. The approach is validated using three examples from the BPM Academic Initiative described as Dynamic Condition Response Graphs which we reconfigured for a variety of transitions requirements.

Keywords: Dynamic reconfiguration · Controller synthesis · DCR graph

1 Introduction

Business processes are invaluable for ensuring that task and activity execution achieves business objectives. Workflows, operational representations of business processes, are typically derived from requirements in a manual process that is complex and error-prone. Organisations require that their business processes reflect their evolving practices maintaining compliance with their policies, strategies and regulations (e.g., [27]). Workflows must be evolved accordingly too. *Business process reconfiguration* involves not only devising the new workflow but also dynamically changing the old workflow with the new one.

Key to reconfiguration is understanding how the transition between the old and new workflow should be managed. Domain independent transition requirements have been studied extensively. For instance, [9] discusses “immediate”

reconfiguration requirements that assert that reconfiguration must occur as soon as possible but only at a state in which the new workflow prescribes behaviour consistent with the old one. “Delayed” reconfiguration asserts that living instances must finish using the old workflow, while fresh instances are created using the new one. In some cases, *domain specific transition requirements* are required. For instance, reconfiguration may be required as soon as possible yet for some live instances in particular states an exceptional treatment may be required, including repeating or roll-backing an activity.

Indeed, business process reconfiguration can be extremely challenging and can greatly benefit from automated techniques that support *(i)* analysing business process requirements and transition requirements, and *(ii)* constructing workflows and reconfiguration strategies satisfying these requirements.

One approach to automation is *build and verify*, in which formal verification techniques provide a sound basis for workflow analysis and can be used to ensure workflow requirement satisfaction. However, post-hoc verification requires prior construction of the workflow, and modification entails re-verification. An alternative approach is to automatically produce *correct-by-construction* workflows and reconfiguration strategies directly from requirements.

Although automatic construction of workflows from requirements has been studied (e.g., [11, 20]), the *synthesis of reconfiguration strategies* for domain specific user defined transition requirements has not received attention so far.

In this paper we present a fully automated technique for business process reconfiguration based on discrete event controller synthesis. We use synthesis to not only produce correct-by-construction workflows from business process requirements but also to compute a reconfiguration strategy that guarantees progress from an old workflow towards the a new one while satisfying any user-defined transition requirements. We discuss a translation of Dynamic Condition Response (DCR) graphs [11], a declarative language for business process requirements, into a formalism based on Labelled Transition Systems and Linear Temporal Logic [21] which is suitable for controller synthesis [7] and build upon recent work on dynamic controller update [19]. We validate the approach using three examples from the BPM Academic Initiative [1] described as DCR graphs which we reconfigured for a variety of transitions requirements.

The rest of the paper is structured as follows. Section 2 presents an illustrative example. Formal definitions are presented in Sect. 3. In Sect. 4 we present problems setting out how to frame it as a synthesis problem. An analysis of our technique is presented in Sect. 5. Finally, we present a discussion on related work.

2 Motivating Example

Consider a hospital process taken from a real-life study on a oncology workflow at Danish hospitals [11]. This workflow has *prescribe medicine* and *sign* activities, representing a doctor adding and signing a prescription to the patient record. In addition, a nurse, is capable of doing *give medicine* in response to the

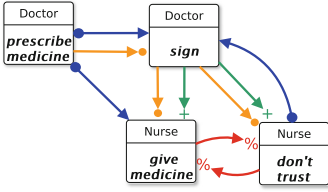


Fig. 1. DCR graph for a hospital process

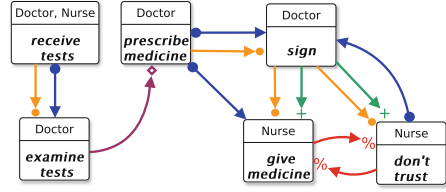


Fig. 2. DCR graph model for new hospital process.

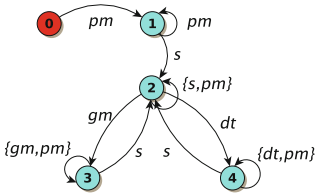


Fig. 3. Workflow for a hospital process

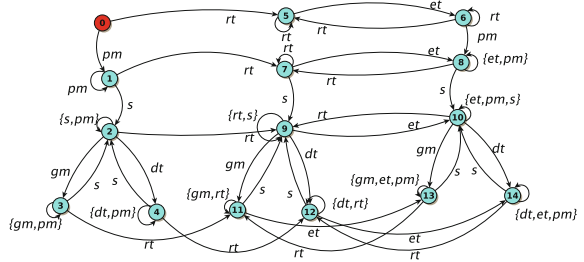


Fig. 4. Workflow for new hospital process.

doctor prescription, or, in contrast, the nurse may indicate that they do not trust the prescription or signature by performing the *don't trust* activity. Workflow requirements include that (i) the doctor must perform *prescribe medicine* to a patient before *sign*, (ii) the nurse can not do *give medicine* nor *don't trust* if the doctor has not done *sign*, and (iii) the nurse can not perform both *give medicine* and *don't trust*, only one is allowed.

Figure 1 shows these requirements modelled using Dynamic Condition Response (DCR) graphs as originally presented in [11]. A workflow that satisfies these requirements is depicted in Fig. 3 where *pm*, *s*, *gm* and *dt* labels refer to activities *prescribe medicine*, *sign*, *give medicine* and *don't trust*, respectively. The workflow is the underlying semantics of the model in Fig. 1 and can be constructed automatically using controller synthesis as described in Sect. 4.

Consider a scenario in which while patients are being treated the workflow must be changed (taken from [18]). For instance, suppose that a new internal regulation is to be put in place stating that doctors must not do *prescribe medicine* if new tests have arrived (*receive tests*) but have not been examined (*examine tests*). Also, as expected, *receive tests* must happen before *examine tests*. This change involves two new activities and extra rules as depicted in Fig. 2 which describes a significantly more complicated workflow (Fig. 4 where *rt* and *et* labels refer to *receive tests* and *examine tests*) that can be automatically synthesised.

A crucial decision to make is how to reconfigure a live instance running the old workflow to the new one. A naive approach would be to require an immediate [9] reconfiguration regardless of the living instance's state. Thus, if the living

instance is in state 2 of Fig. 3 it should evolve to being in state 2 of Fig. 4 (i.e., $2 \rightsquigarrow 2$). However, this puts the patient at risk: The old workflow does not track the occurrence of *receive tests*, yet new tests may actually have been received and new prescriptions should not be done without examining them. It is safer to assume, at reconfiguration time, that tests may exist and to require be examining them (should they be available) rather than ignoring them. Consequently, it is more appropriate to update the old workflow to the new workflow according to the following mapping: $(0 \rightsquigarrow 5), (1 \rightsquigarrow 7), (2 \rightsquigarrow 9), (3 \rightsquigarrow 11), (4 \rightsquigarrow 12)$.

The provision of a mapping between workflow states ensuring that a transition requirement holds can be very difficult for complex workflows. An alternative is to allow a declarative description of transition requirements and to compute a mapping automatically. For our example, what is needed is to force *examine tests* when reconfiguring. Note that is inconsistent with both the old and new workflow requirements. In the old workflow, there is no *examine tests* activity and in the new workflow requirements *examine tests* is required after *receive tests*. Thus, what we need to express is that there is a period during the reconfiguration where neither workflow requirements hold and in which *examine tests* (and nothing else) must occur. In this paper we show how domain specific transition requirements such as these can be modelled and how to automatically build a strategy for taking a live instance running a workflow to a new workflow guaranteeing all transition requirements.

3 Preliminaries

In this work we use Dynamic Condition Response Graphs [12] to specify business processes. To simplify presentation we use a reduced version that does not include nesting, roles, principals and roles assignments.

Definition 1 (Dynamic Condition Response Graph). *A Dynamic Condition Response Graph (DCR Graph) is a tuple $DG = (A, R, M)$ where A is a finite set of activities, the nodes of the graph. R is a set of graph edges. Edges are partitioned into five kinds, named and drawn as follows: conditions ($\rightarrow\bullet$), responses ($\bullet\rightarrow$), inclusions ($\rightarrow+$), exclusions ($\rightarrow\%$) and milestones ($\rightarrow\diamond$). M is the marking of the graph. This is a triple of sets of activities (Ex, Re, In), where Ex are the previously executed, Re the currently pending and In the currently included. For all $(e, e') \in E \times E$, $e \rightarrow+e'$ or $e \rightarrow\%e'$ or neither of them. We denote $(\bullet\rightarrow e) = \{e' \in A \mid e' \bullet\rightarrow e\}$, $(e\bullet\rightarrow) = \{e' \in A \mid e \bullet\rightarrow e'\}$, and similarly for $\rightarrow\bullet$, $\rightarrow+$, $\rightarrow\%$ and $\rightarrow\diamond$.*

Definition 2 (Enable activity of a DCR Graph). *Let $DG = (A, R, M)$ be a DCR graph, with $M = (Ex, Re, In)$. An activity $e \in A$ is enabled if and only if (a) $e \in In$, (b) $(In \cap (\rightarrow\bullet e)) \subseteq Ex$, and (c) $Re \cap In \cap (\rightarrow\diamond e) = \emptyset$.*

Definition 3 (Executing DCR Graph). *Let $DG = (A, R, M)$ be a DCR graph, with marking $M = (Ex, Re, In)$ and e is enabled. The result of executing e is a DCR Graph $DG' = (A, R, M')$ with $M' = (Ex', Re', In')$ such that (a) $Ex' =$*

$Ex \cup \{e\}$, (b) $Re' = (Re \setminus \{e\}) \cup (e \bullet \rightarrow)$, and (c) $In' = (In \cup (e \rightarrow_+)) \setminus (e \rightarrow_\%)$. We assume that initially $In = A$ and $Re = Ex = \emptyset$.

To capture the underlying semantics of DCR graphs we use Labelled Transition Systems [14]. They are a canonical, compositional, representation of events structures ideally suited to model checking of business processes and synthesis of discrete event controllers.

Definition 4 (Labelled Transition System). *A Labelled Transition System (LTS) E is a tuple $(S_E, L_E, \Delta_E, e_0)$, where S_E is a finite set of states, $L_E \subseteq \mathcal{L}$ is its communicating alphabet, \mathcal{L} is the universe of all observable events, $\Delta_E \subseteq (S_E \times L_E \times S_E)$ is a transition relation, and $s_0 \in S_E$ is the initial state. A path of E is a sequence $\pi = s_0, \ell_0, s_1, \ell_1, s_2, \dots$ where for every $i \geq 0$ we have $(s_i, \ell_i, s_{i+1}) \in \Delta_E$. A trace w is a sequence obtained by removing states from π .*

Definition 5 (Parallel Composition). *The parallel composition $E \parallel C$ of LTS $E = (S_E, L_E, \Delta_E, e_0)$ and $C = (S_C, L_C, \Delta_C, c_0)$ is an LTS $(S_E \times S_C, L_E \cup L_C, \Delta_{\parallel}, (e_0, c_0))$ such that Δ_{\parallel} is the smallest relation that satisfies the rules:*

$$\frac{(e, \ell, e') \in \Delta_E \wedge \ell \notin L_C \quad (c, \ell, c') \in \Delta_C \wedge \ell \notin L_E \quad \ell \in L_E \cap L_C}{((e, c), \ell, (e', c')) \in \Delta_{\parallel}} \quad \frac{(e, \ell, e') \in \Delta_E \wedge (c, \ell, c') \in \Delta_C}{((e, c), \ell, (e', c')) \in \Delta_{\parallel}} \quad \frac{\ell \in L_E \cap L_C}{((e, c), \ell, (e', c')) \in \Delta_{\parallel}}$$

We use a linear temporal logic of fluents to provide a uniform framework for specifying state-based temporal properties in event-based models [10]. FLTL [10] is a linear-time temporal logic for reasoning about fluents. A *fluent* is defined by a pair of sets and a Boolean value: $f = \langle I, T, Init \rangle$, where $f.I$ is the set of initiating events, $f.T$ is a set of terminating events and $f.I \cap f.T = \emptyset$. A fluent may be initially *true* or *false* as indicated by $f.Init$.

Let \mathcal{F} be the set of all possible fluents. An FLTL formula is defined inductively using the standard Boolean connectives and temporal operators **X** (next), **U** (strong until) as follows: $\varphi ::= f \mid \neg\varphi \mid \varphi \vee \psi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U}\psi$, where $f \in \mathcal{F}$. We define $\varphi \wedge \psi$ as $\neg\varphi \vee \neg\psi$, $\Diamond\varphi$ (eventually) as $\top \mathbf{U}\varphi$, $\Box\varphi$ (always) as $\neg\Diamond\neg\varphi$, and $\varphi \mathbf{W}\psi$ (weak until) as $\varphi \mathbf{U}\psi \vee \Box\varphi$.

The trace $\pi = \ell_0, \ell_1, \dots$ satisfies a fluent f at position i , denoted $\pi, i \models f$, if and only if, one of the following conditions holds: (a) $f.Init \wedge (\forall j \in \mathbb{N} \cdot 0 \leq j \leq i \Rightarrow \ell_j \notin f.T)$, and (b) $\exists j \in \mathbb{N} \cdot (j \leq i \wedge \ell_j \in f.I) \wedge (\forall k \in \mathbb{N} \cdot j < k \leq i \Rightarrow \ell_k \notin f.T)$. In other words, a fluent holds at position i if and only if it holds initially or some initiating event has occurred, but no terminating event has yet occurred.

We say φ is a safety formula if there is a finite trace π such that:

$$\begin{aligned} \pi, i \models \neg\varphi &\triangleq \neg(\pi, i \models \varphi) \\ \pi, i \models \varphi \vee \psi &\triangleq (\pi, i \models \varphi) \vee (\pi, i \models \psi) \\ \pi, i \models \mathbf{X}\varphi &\triangleq \pi, i+1 \models \varphi \\ \pi, i \models \varphi \mathbf{U}\psi &\triangleq \exists j \geq i \cdot \pi, j \models \psi \wedge \forall i \leq k < j \cdot \pi, k \models \varphi \end{aligned}$$

We use $\pi \models \varphi$, instead of $\pi, 0 \models \varphi$.

Control problems aim to build an LTS that satisfies a given set of declarative requirements under certain environment conditions by having control of only a subset of the events of the environment.

Definition 6 (LTS Control [7]). *Let $E = (S_E, L_E, \Delta_E, e_0)$ be an environment model in the form of an LTS, $L_c \subseteq L_E$ be a set of controllable events, and G be a controller goal in the form of an FLTL property. A solution for the LTS control problem with specification $\mathcal{E} = (E, G, L_c)$ is an LTS C such that C only blocks events in L_c , $E \parallel C$ is deadlock free, and $E \parallel C \models G$.*

Definition 7 (DCU Problem [19]). *Let $\mathcal{E} = (E, G, L_c)$ be an old specification, $\mathcal{E}' = (E', G', L'_c)$ be a new specification, T be a safety FLTL formula, $R \subseteq (S_E \times S_{E'})$ be a mapping relation of states and, $stopOldReq$ and $startNewReq$ are special events denoting the ending of old and start of new requirements, respectively. A solution for the DCU Synthesis Problem is a controller C_u such that: (a) $C_u \models G \mathbf{W} stopOldReq$, (b) $C_u \models T$, (c) $C_u \models \Box(startNewReq \rightarrow G')$, and (d) $C_u \models \Box(beginReconf \rightarrow (\Diamond stopOldReq \wedge \Diamond startNewReq))$.*

The output of a DCU problem is an LTS C_u where every trace satisfies that (a) the old requirements hold G until $stopOldReq$ is triggered, (b) the transition requirements hold, (c) the new specification G' must be valid from $startNewReq$ is onwards, and (d) the update eventually happens.

4 Dynamic Reconfiguration of Business Processes

In this section we first show how to synthesize a workflow from a DCR graph using controller synthesis (Definition 6) and then show how to use dynamic controller update (Definition 7) for workflow reconfiguration.

4.1 Workflow Synthesis as a Control Problem

We now show how to extract from a DCR graph a set of controllable events L_C , an LTS E , and a FLTL formula G such that controller synthesis (Definition 6) results in a controller that enables and disables activities in such a way that its environment, as long as it only executes enabled activities, satisfies the business process requirements as described in the DCR graph. Thus L_C will contain activity enabling and disabling events, while events modelling the execution of activities will be monitorable but not controllable. The LTS E will model the assumptions the controller can rely upon to guarantee workflow requirements. Finally, the formula G encodes the domain specific aspects of the DCR graph, namely the arrows that establish dependencies between activities.

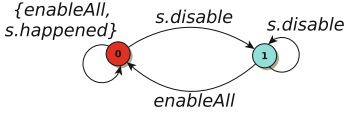


Fig. 5. *Happens(s)* LTS constraining the occurrence of *s.happened*.

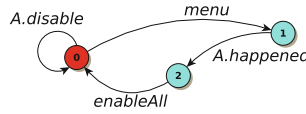


Fig. 6. *Turns* LTS constraining controller and environment turns.

Controllable and Monitorable Events. The set of events that describe the control problem are defined by the activities that appear in the DCR graph (i.e., the set A). We introduce two events for each activity $a \in A$: $a.disable$ and $a.happened$. The first is an event controlled by the controller. The second, is an event that will be selected by the environment (e.g., the nurse and the doctor) to indicate that the activity was executed. We say that $a.happened$ is monitorable or uncontrolled. Note that we do not introduce $a.enabled$, rather we assume an event $enableAll$ to reduce the number of events and states of the control problem. The controller will $enableAll$ activities then select which ones to disable in such a way that if the environment executes an enabled activity, it will be consistent with the business process requirements.

We introduce one extra event, $menu$, to model the turn based interaction where the controller offers to its environment a menu of activities to perform. First, the controller will select what activities to disable then it indicates using $menu$ that it is the environment’s turn to decide what activity to execute.

In conclusion, the set of controllable and uncontrollable events are $L_C = \{a.disable \mid a \in A\} \cup \{menu, enableAll\}$ and $\overline{L_C} = \{a.happened \mid a \in A\}$

Environment Model. The LTS E models the two assumptions that the controller can rely upon to guarantee workflow requirements.

The first assumption is that activities can only happen when they are enabled. This can be modelled using one LTS model for each activity and composing them all in parallel. In Fig. 5 we show an LTS, *Happens(s)*, modelling the assumption for activity *sign*. State 0 models that *sign* is enabled (thus, the outgoing transition $s.happened$) while state 1 models that the activity is disabled (i.e., there is no outgoing $s.happened$ transition). Events $enableAll$ and $s.disable$ toggle between state 0 and 1. We assume the activity is initially enabled.

The second assumption is that the environment will play in turns with the controller. The controller chooses what activities may be executed without violating workflow requirements, and then, the environment picks which of the enabled activities is to be executed. We use only one LTS, *Turns* depicted in Fig. 6, to model this assumption. The initial state (0) models the turn of the controller where any activity in A can be disabled. Event $menu$ models when the controller relinquishes its turn offering a menu of activities to perform. State 1 is the environment’s turn in which it can select only one activity in A to be executed, going to state 2. Here, all activities are enabled with $enableAll$ event to start again with controller’s turn at state 0.

The assumptions reflect the operation of the workflow engine that will be controlled. In the hospital example, the controller first decides which activities should be enabled (*enableAll* and *a.disabled*) and then presents them to hospital staff (*menu*). It is assumed that nurses and doctor will only perform an activity if the activity is displayed by the engine, and that once performed they will report back through the engine (*a.happened*). At this point, the controller will decide again what activities to enable and update the engine display. Obviously, the *menu* event must only occur when the controller has enabled exactly all activities that if executed would not violate workflow requirements. This controller behaviour is synthesised automatically based on the formalisation of goals described next. In conclusion the LTS environment E is defined as follows $E = Turns \parallel Happens(a_1) \parallel \dots \parallel Happens(a_n)$ with $A = \{a_1, \dots, a_n\}$.

Controller Goals. Goal G must model the constraints between activities that are expressed in DCR Graphs with arrows between activities. Our encoding resembles that of [20] where LTL formulas are used to formalise activity constraints of similar nature to those of DCR graphs.

We introduce three fluents for each activity $a \in A$ modelling if a belongs to sets Ex , Re , and In according to Definition 3. For simplicity, we assume that the initial marking of the DCR graph is such that $In = A$ and $Re = Ex = \emptyset$.

- $a.Executed$ models if $a \in Ex$ and is defined as $\langle \{a.happened\}, \emptyset, \perp \rangle$. In other words, initially no activity is in Ex and once in Ex it is never removed (see Definition 3a).
- $a.Required$ models if $a \in Re$ and is defined as $\langle \{a'.happened \mid a' \in (\bullet \rightarrow a)\}, a.happened, \perp \rangle$. That is, all activities are initially not required and the execution of a activity makes it no longer required, and any activity in a response relation with a makes it a required (see Definition 3b). In the hospital example, fluent $s.Required$ is defined as $\langle \{pm.happened, dt.happened\}, s.happened, \perp \rangle$ because activity *sign* is a response to *don't trust* and *prescribe medicine* according to Fig. 1. Note that for cases where $a \bullet \rightarrow a$, we define $a.Required$ as $\langle \{a'.happened \mid a' \in (\bullet \rightarrow a)\}, \emptyset, \perp \rangle$ because the execution of a does not turn false the fluent.
- $a.In$ models if $a \in In$ and is defined as $a.In = \langle \{a'.happened \mid a' \in (\rightarrow_+ a)\}, \{a'.happened \mid a' \in (\rightarrow\% a)\}, \top \rangle$, which mimics Definition 3c. Based on the relations modelled in Fig. 1, the fluent $gm.In$ is defined as $\langle \{s.happened\}, \{dt.happened\}, \top \rangle$.

We introduce FLTL formulas to preserve the rules that govern when an activity can be executed (i.e., is enabled) according to Definition 2. In other words, the formulas will relate the occurrence of *a.happened* with fluents $a'.Executed$, $a'.Required$, and $a'.In$ for all $a' \in A$.

- For rule (a) of Definition 2 we introduce for every activity $a \in A$ a formula $\alpha_a = \Box(a.happened \rightarrow a.In)$.

- For rule (b) of Definition 2 we introduce for all $a \in A$: $\beta_a = \Box(a.happened \rightarrow \bigwedge_{a' \in (\rightarrow \bullet a)} (a'.In \rightarrow a'.Executed))$. For instance, for *sign*, according to Fig. 1 we have $\beta_s = \Box(s.happened \rightarrow (pm.In \rightarrow pm.Executed))$.
- For rule (c) of Definition 2 we introduce for each $a \in A$: $\kappa_a = \Box(a.happened \rightarrow \bigwedge_{a' \in (\rightarrow \diamond a)} (\neg a'.Required \vee \neg a'.In))$. For instance, $\kappa_{pm} = \Box(pm.happened \rightarrow (\neg et.Required \vee \neg et.In))$ for Fig. 2.

In summary, G is defined as $\bigwedge_{a \in A} \alpha_a \wedge \beta_a \wedge \kappa_a$.

Workflow Synthesis. Above we have described how to build from a DCR graph model D , the set of controllable events L_c , the LTS environment E , and the FLTL formula G that can be used to define a control problem $\mathcal{E} = (E, G, L_c)$. A solution to this problem is a controller LTS C that decides when to enable and disable activities (which correspond to events in L_c) such that when running with an environment that plays in turns and only executes enabled activities (as described in E) satisfies all business process requirements (as captured in G). In other words: $E \parallel C \models G$ (Definition 6).

Note that $E \parallel C \models G$ is not enough. We need the controller to be maximal in the sense of that at any *menu*, the maximal set of activities should be enabled that do not violate G . Consider a workflow for the hospital in which after *sign* only *give medicine* is enabled. The sequence *sign*, *give medicine* does not violate G , but *sign* followed by *don't trust* should also be possible. To ensure maximality we exploit a characteristic of the synthesis algorithm implemented in the MTSA tool [6] that we use for synthesis: MTSA builds eager components in the sense that they take the shortest route to satisfying their requirements. As the controller is forced to do *enableAll*, the synthesis algorithm will try to do as few disable actions as possible while still ensuring G , thus a maximal number of activities will always be enabled.

The controllers for the DCR graphs depicted in Figs. 1 and 2 have 188 and 2291 states respectively and are too large to depict in this paper. Instead we show abstract versions of these controllers (Figs. 3 and 4) in which *enableAll*, disable and *menu* events are hidden. This provides a view similar to what the Nurse and Doctor would see, only the activities that are enabled and not the controllers incremental decisions of enabling and disabling activities. Note that the abstract controllers are built automatically by MTSA tool using a hiding operator and weak bisimilarity minimisation [17].

4.2 Workflow Reconfiguration as a Dynamic Controller Update

This section is organised as follows. We first discuss how domain specific transition requirements for a workflow reconfiguration can be described using FLTL. This involves introducing two new events. We then discuss what a solution to a reconfiguration problem may look like and finally how such solutions can be built automatically solving a Dynamic Controller Update problem.

Specification of Transition Requirements. Recall the Hospital workflow example discussed in Sect. 2 where a transition requirement stating that activity *examine tests* should be forced when reconfiguring. More precisely, just before the moment the new business process requirements should be enforced, *examine tests* is required. The reason for requiring “just before” is that executing *examine tests* without a previous *receive tests* is not allowed in the new business process.

To formalise this transition requirement we need to refer to the moment in which the old business requirements are to be dropped (*stopOldReq*) and the moment in which the new business requirements come into force (*startNewReq*). With these two new events the transition requirement can be formulated as follows $T_h = \Box(\text{stopOldReq} \rightarrow ((\bigwedge_{a \in A \setminus \{et\}} \neg a.happened) \mathbf{W} (et.Executed \wedge \text{startNewReq})))$. Note that guaranteeing this formula requires enabling and disabling activities such that the uncontrolled events *a.happened* occur or not as required by T_h . A standard domain independent transition requirement that states that at any point one of the two business process must be adhered to (i.e., there is no transition period) can be stated as follows: $T_\emptyset = \Box((\text{StopOldReq} \wedge \neg \text{StartNewReq}) \rightarrow \bigwedge_{a \in A} \neg a.happened)$.

Reconfiguration Workflows. Returning to T_h , what would a solution to this reconfiguration problem be? Assume the workflow in Fig. 3 is in state 2, a solution to the reconfiguration is to deploy a workflow that does forces *examine tests* and then reaches a state 10 in Fig. 4. In other words, we need to build a workflow that manages the transition from the old to the new workflow, we call this workflow the *reconfiguration workflow*.

This reconfiguration workflow that assumes that the old workflow is in state 2 is inadequate as, before it takes control, a new activity (e.g., *give medicine*) may be executed taking the old workflow (Fig. 3) in state 2 to state 3. Should this happen then the reconfiguration should force *examine tests* and then move to state 13 in Fig. 4 instead of 10. Thus, the goal is to build a reconfiguration workflow that can manage the transition from any state in the old workflow.

Conceptually, our solution builds one reconfiguration workflow that consists of three phases. The first is structurally equivalent to the old workflow (modulo a new event *beginReconf*). This allows hot-swapping the old workflow with the reconfiguration workflow, and setting the initial state of the latter according to the current state of the former. The second phase is triggered by an event *beginReconf*. At this point, the reconfiguration workflow may start to deviate from the behaviour of the old workflow to ensure transition requirements. At the point it does so, it must first signal *stopOldReq*. The third phase is one in which the new workflow requirements are satisfied. Entering this third phase is signalled with *startNewReq*.

In Fig. 7 we depict an abstract reconfiguration workflow (enabling, disabling and *menu* events are hidden) that implements the reconfiguration from business process requirements of Fig. 1 to those of Fig. 2 under transition requirement T_h . The blue rectangle on the left represents the first phase of the reconfiguration workflow. Note that the structure of states and transitions is that of the workflow

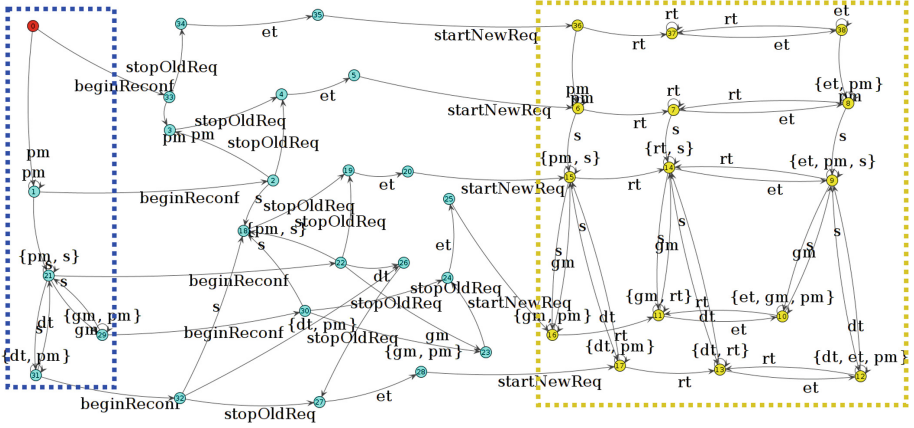


Fig. 7. Reconfiguration workflow with transition requirement T_h .

to be replaced (Fig. 3), thus hotswapping this workflow in is trivial. Note that all states in the blue region have an outgoing transition labelled *beginReconf*. When *beginReconf* is triggered, no matter what the current state is, there is a path to the yellow region on the right. The yellow region represents the new workflow as in Fig. 4. The transition from the old requirements to new ones, while satisfying the transition requirements is represented by between both rectangles. It is noteworthy that there are no loops during the transition phase which guarantees that eventually the new business process requirements will be enforced.

Automatic Construction of Reconfiguration Workflows. Summarising, Fig. 7 represents a solution to the problem of reconfiguring business process requirements in Fig. 1 to those of Fig. 2 under transition requirement T_h . We now discuss how such solution can be built by solving a DCU problem Definition 7. The DCU problem requires two control problems $\mathcal{E} = (E, G, L_c)$ and $\mathcal{E}' = (E', G', L'_c)$ which represent in this case the old and new business process synthesis problems as described in Sect. 4.1. DCU also requires a transition requirement T and a state mapping R from the states of E to those of E' . We have discussed T , we now discuss R .

The purpose of relation R is to explain the relationship between the assumptions modelled in each control problem. The issue is that E tracks assumptions for a controller synthesised from C , when a reconfiguration is deployed it is not possible to know what the state of the assumption E' is. R must be provided by a user to address this problem. In this setting, the mapping can be trivially defined as the only differences between E and E' are the LTSs (like the one in Fig. 5) representing activities that are present in one business process and not the other. Furthermore, we know that for any new activity, this one can never have been enabled by the controller of the old workflow. In consequence, R can be defined as the state

Table 1. Case study summary

Case study	# Activities	# Arrows	Transition requirement	Reconfiguration workflow (# States)	Minimised reconf. workflow (# States)
Oncology hospital	6	13	T_{\top}	18667	54
			T_{\emptyset}	9817	34
			T_h	11155	39
			T'_h	15094	54
Doctor assessment process	10	25	T_{\emptyset}	22448	39
			T_D	27512	42
Insurance process	11	25	T_{\emptyset}	15484	51
			T_I	14233	48
Computer repair process	18	26	T_{\emptyset}	43307	59
			T_C	52652	63

identity relation for all LTS that are in E and E' and as the constant relation 0 (i.e., the initial state) for LTSs representing new activities.

Thus, given two DCR graphs D and D' describing the old and new business process requirements and a transition requirement T we can automatically build control problems $\mathcal{E} = (E, G, L_c)$ and $\mathcal{E}' = (E', G', L'_c)$ as described in Sect. 4.1 and R to describe and solve a DCU problem. An abstraction of the solution to the DCU problem for the Hospital reconfiguration problem with T_h described above is depicted in Fig. 7 and was built automatically using MTSA.

An important methodological note is that not every DCU problem has a solution. It is possible to provide two control problems \mathcal{E} and \mathcal{E}' that are individually realisable yet for certain transition requirements, the update is impossible. In terms of business process reconfiguration this means that it is possible to start with two sets of business process requirements that are consistent yet to propose a transition requirement that is too stringent to allow for a correct reconfiguration. An example of this, for the Hospital example, is to require $T = \square(\text{startNewReq} \rightarrow \neg pm.Executed)$. There is no reconfiguration strategy that can guarantee that the new business process requirements will be put in force *independently of the current state* of the live instances of the old workflow: There is no reconfiguration strategy for a live instance in which activity *prescribe medicine* has been executed.

5 Validation

The purpose of this section is to show applicability of the approach by using, in addition to motivational example, three business processes taken from BPM Academic Initiative [1] that were also modelled in the DCR Graph Tool [15]. We chose these to avoid bias in producing our own DCR graphs from workflows.

Each case study requires two DCR graphs, a source and a target for reconfiguration. We manually produced variants for each case study and used domain independent transition requirement such as T_\emptyset (see Sect. 4.2) in addition to domain specific ones. All examples were run using an extension of the MTSA tool [6] and can be found at [2]. Overall, 10 reconfigurations were defined and solved, corresponding to different choices of transition requirements for each case study. In Table 1 we report on examples, the number of distinct activities and constraints they involve, the size of the resulting reconfiguration workflow and of its minimised version (this involves hiding all enable, disable, and *menu* events).

5.1 Oncology Hospital

This case study already discussed above is the only one for which both reconfiguration source and target DCR graphs existed. Both were taken from [18]. We modelled various alternative transition requirements and built business process reconfiguration for each of them.

We first used a trivial transition requirement ($T_\top = \top$) to confirm that a reconfiguration strategy exists but it allows undesired behaviour. Indeed the reconfiguration process allowed: *beginReconf*, *stopOldReq*, *give medicine*, *startNewReq* ... The trace is one in which a live instance for which no activities have occurred start to be reconfigured, the old business process requirements are dropped and before the new ones are enforced the patient is given medicine (without a signed prescription by a doctor!). This problem arises because T_\top allows any activity during reconfiguration. Using a stronger domain independent transition requirement, T_\emptyset , the reconfiguration behaviour obtained is exactly that of an immediate reconfiguration es defined by [9].

We considered two domain specific transition requirements, T_h as discussed in Sect. 4.2 and one that delays reconfiguration when a nurse has indicated distrust regarding a patient's record: $T'_h = T_\emptyset \wedge \square((dt.Executed \wedge \neg gm.Executed) \rightarrow \neg stopOldReq)$. As expected the resulting reconfiguration behaviour is like that of T_\emptyset except that *stopOldReq* is delayed when between *don't trust* and *give medicine*.

5.2 Doctor Assessment Process

An assessment process for doctors in a hospital involves a manager asking an expert to evaluate each doctor. We used the original DCR graph as the target for reconfiguration and removed one activity to produce the source DCR graph. We considered a process that initially does not pay experts for their evaluation and that is to be reconfigured to support paying expert revision fees.

Using the transition requirement T_\emptyset we obtain a reconfiguration that can be performed immediately at any point of the execution of the first process. This is because the activity of paying experts simply adds to the end of the current process an additional activity. However, immediate reconfiguration may result in paying experts that had agreed to do a review for free in the old process. to avoid this scenario we specified the transition requirement T_D stating that if

reconfiguration is requested after receiving expert review, the expert must not be paid: $T_D = T_\emptyset \wedge (\Box(\text{startNewReq} \wedge \text{recExp.Executed}) \rightarrow \Box \neg \text{pay.Executed})$ where recExp is the activity representing the reception of expert review.

5.3 Insurance Process

The business process for an insurance company includes two roles: agents and clerks. Originally, the clerk must, upon receiving a *new customer claim*, call the agent to check the claim and create a new customer case. The new requirement to be put in place states that *create a new case* must happen before *call the agent* (this corresponds to the classic parallel to sequential reconfiguration [27]). We solved the reconfiguration for two different transition requirements.

We used T_\emptyset to compute a reconfiguration workflow which delays reconfiguration when *call the agent* has been executed but *create a new case* has not. For all other scenarios, the reconfiguration workflow do an immediate change. An alternative is to modify the target DCR graph with a *kill* activity that excludes all other activities, modelling the killing of an instance. Then a transition requirement that forces *kill* when *call the agent* has been executed before *create a new case* can be specified as $T_I = T_\emptyset \wedge \Box((\text{call.Executed} \wedge \neg \text{create.Executed} \wedge \text{startNewReq}) \rightarrow (\neg ED \mathbf{W} \text{kill.Executed}))$, where ED is the disjunction of disable events for all activities except *kill* plus *enableAll*.

5.4 Computer Repair Process

A computer repair service starts when a customer brings a defective computer. If service provider and customer agree on a budget, then hardware and software repair activities are performed. We added a new role, that of a supervisor, that must approve a budget before it is sent to the customer. We used three activities for this: *send to supervisor*, *approve*, and *reject*.

Initially, we solved this reconfiguration problem with the transition requirement T_\emptyset . As expected, executions in which the reconfiguration is requested after the budget is sent to the customer, the reconfiguration is delayed so as to not contradict the requirement of supervisor approval.

An alternative we modelled is one in which we force asking for approval for any instance in which the customer has received the budget but repair has not started. If the supervisor rejects the budget, then the customer must be contacted and apologies must be offered. The following formula (where *sup* is the activity *send to supervisor*) captures this reconfiguration requirement: $T_C = \Box((\text{stopOldReq} \wedge \neg \text{RepairStart}) \rightarrow (\neg \text{Happens} \mathbf{W} (\text{sup.Executed} \wedge \text{startNewReq})))$ where Happens is the disjunction of disable events for all activities except *yes*, *no*, and *sup* plus *enableAll*.

6 Discussion and Related Work

The problem of business process reconfiguration has been studied extensively for some time [9]. [27] provides a classification of potential errors resulting from

process changes. A survey of correctness criteria guaranteed by dynamic change techniques is presented in [24]. A taxonomy of reasons for reconfiguration is presented in [25]. Methodological and automated support for reconfiguration has also been studied previously. Work such as [3, 4, 9] approach reconfiguration as a problem of defining dynamic transitions from one state of current workflow to another one in the new one. Without transition periods, changes can be partitioned into immediate or delayed [9]. A different take on reconfiguration is workflow versioning (e.g., [13, 29]) where multiple workflow versions such are running simultaneously. In all cases, and in contrast to our work, the notion of a transition period in which remedial activities need to be implemented that are not compliant with the current and new workflow is not considered. The notion of reconfiguration is related to that of dynamic software updates. These have also been studied in terms of the different properties that may be expected during the update (e.g. [28]).

To reason about reconfiguration, our approach assumes a declarative specification of business process requirements (rather than an operational description in the form of a workflow). Declarative modelling approaches for business processes have been studied before. The ConDec [20] language was introduced for modelling business process based on linear temporal logic (LTL [21]). In [11], an operational semantics for a declarative graph based language is proposed and a tool [15] for enacting the underlying workflow is available. Rule based descriptions of business process requirements have also been proposed (e.g., [16, 26]). Such descriptions are naturally executable. Both support changing rules during the execution of a workflow, however there is no support for understanding or guaranteeing properties of the reconfiguration. Thus, understanding if a delayed or a immediate change is needed must be done before introducing a new rule. Our approach requires a declarative description of business process requirements in a rather general language (FLTL) and provides guarantees over the reconfiguration process. The choice of DCR graphs as a starting point is accidental, we could apply a similar translation for other declarative languages.

Automatic construction of operational or executable models from declarative requirements has also been studied extensively, including work on supervisory control [23], synthesis of reactive designs [22] and automated planning [5]. This paper builds on the synthesis of discrete event controllers and in particular the work presented in [8] that uses LTS and FLTL as the input for synthesis. We strongly build on the result presented in [19] where a general technique for updating at runtime a controller. In this paper we adapt and apply this technique in the context of business process reconfiguration for DCR graph specifications.

7 Conclusions

We address the problem of business process reconfiguration by providing an automatic technique that builds a reconfiguration workflow that is guaranteed to preserve any reconfiguration transition requirements provided by a user. The technique requires a declarative description of the current and new business process requirements: in this paper we start from DCR graphs, and an LTL property

that describes the properties that must hold during the reconfiguration. The approach allows immediate and delayed changes, and also reconfigurations in which there is a period between business processes in which additional domain specific preparatory or remedial activities can be executed. The result is a workflow that can be hotswapped with the current one and actively manages the transition to the new business process requirements ensuring correctness.

Acknowledgement. This project has received funding from the European Union's Horizon 2020 research, innovation programme under the Marie Skłodowska-Curie grant agreement No. 778233, PPL CAIS-0204-11, Pict 2014 No. 1656, Pict 2015 No. 3638 and Ubacyt 2018-0297BA.

References

1. Business process management academic initiative. <https://bpmmai.org/>
2. MTSA synthesis tool and examples. <http://mtsa.dc.uba.ar>
3. van der Aalst, W.M.: Exterminating the dynamic change bug: a concrete approach to support workflow change. *Inf. Syst. Front.* **3**(3), 297–317 (2001)
4. Badouel, E., Oliver, J.: Reconfigurable nets, a class of high level Petri nets supporting dynamic changes within workflow systems. Ph.D. thesis, Inria (1998)
5. Cimatti, A., Pistore, M., Roveri, M., Traverso, P.: Weak, strong, and strong cyclic planning via symbolic model checking. *Artif. Intell.* **147**, 35–84 (2003)
6. D'Ippolito, N., Fischbein, D., Chechik, M., Uchitel, S.: MTSA: the modal transition system analyser. In: *ASE 2008*, pp. 475–476 (2008)
7. D'Ippolito, N., Braberman, V., Piterman, N., Uchitel, S.: Synthesising non-anomalous event-based controllers for liveness goals. *ACM TOSEM* **22**(1) (2013)
8. D'Ippolito, N.R., Braberman, V., Piterman, N., Uchitel, S.: Synthesis of live behaviour models. In: *FSE 2010*, pp. 77–86. ACM, New York (2010)
9. Ellis, C., Keddara, K., Rozenberg, G.: Dynamic change within workflow systems. In: *COOCS 1995*, pp. 10–21. ACM (1995)
10. Giannakopoulou, D., Magee, J.: Fluent model checking for event-based systems. In: *ESEC/SIGSOFT FSE 2003*, pp. 257–266. ACM, New York (2003)
11. Hildebrandt, T., Mukkamala, R.R.: Declarative event-based workflow as distributed dynamic condition response graphs. In: *PLACES 2010*, vol. 69, pp. 59–73 (2010)
12. Hildebrandt, T., Mukkamala, R.R., Slaats, T.: Nested dynamic condition response graphs. In: *Arbab, F., Sirjani, M. (eds.) FSEN 2011. LNCS*, vol. 7141, pp. 343–350. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29320-7_23
13. Kradolfer, M., Geppert, A.: Dynamic workflow schema evolution based on workflow type versioning and workflow migration. *Int. J. Coop. Info. Syst.* (1999)
14. Magee, J., Kramer, J.: *State Models and Java Programs*. Wiley, Hoboken (1999)
15. Marquard, M., Shahzad, M., Slaats, T.: Web-based modelling and collaborative simulation of declarative processes. In: *Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) BPM 2015. LNCS*, vol. 9253, pp. 209–225. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_15
16. Mejia Bernal, J.F., Falcarin, P., Morisio, M., Dai, J.: Dynamic context-aware business process: a rule-based approach supported by pattern identification. In: *SAC 2010*, pp. 470–474 (2010)

17. Milner, R. (ed.): *A Calculus of Communicating Systems*. LNCS, vol. 92. Springer, Heidelberg (1980). <https://doi.org/10.1007/3-540-10235-3>
18. Mukkamala, R.R.: A formal model for declarative workflows. Ph.D. thesis, IT University of Copenhagen (2012)
19. Nahabedian, L., et al.: Dynamic update of discrete event controllers. *IEEE TSE* 1 (2018, early access)
20. Pesic, 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
21. Pnueli, A.: The temporal logic of programs. In: *FOCS 1977*, pp. 46–57 (1977)
22. Pnueli, A., Rosner, R.: On the synthesis of a reactive module. In: *POPL 1989* (1989)
23. Ramadge, P.J., Wonham, W.M.: The control of discrete event systems. *Proc. IEEE* **77**(1), 81–98 (1989)
24. Rinderle, S., Reichert, M., Dadam, P.: Correctness criteria for dynamic changes in workflow systems—a survey. *Data Knowl. Eng.* **50**(1), 9–34 (2004)
25. Schonenberg, H., Mans, R., Russell, N., Mulyar, N., van der Aalst, W.M.: Towards a taxonomy of process flexibility. In: *CAiSE 2008*, vol. 344, pp. 81–84 (2008)
26. Vasilecas, O., Kalibatiene, D., Lavbič, D.: Rule-and context-based dynamic business process modelling and simulation. *J. Syst. Softw.* **122**, 1–5 (2016)
27. Van Der Aalst, W.M., Stefan, J.: Dealing with workflow change: identification of issues and solutions. *CSSE* **15**(5), 267–276 (2000)
28. Zhang, J., Cheng, B.H.: Model-based development of dynamically adaptive software. In: *ICSE 2006*, pp. 371–380 (2006)
29. Zhao, X., Liu, C.: Version management in the business process change context. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 198–213. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_15



A First-Order Logic Semantics for Communication-Parametric BPMN Collaborations

Sara Houhou^{1,2}, Souheib Baair^{1,3}, Pascal Poizat^{1,3(✉)}, and Philippe Quéinnec⁴

¹ Sorbonne Université, CNRS, LIP6, 75005 Paris, France
{sara.houhou,souheib.baair,pascal.poizat}@lip6.fr

² LINFI Laboratory, Biskra University, Biskra, Algeria

³ Université Paris Lumières, Université Paris Nanterre, 92000 Nanterre, France

⁴ IRIT - Université de Toulouse, 31000 Toulouse, France
philippe.queinnec@irit.fr

Abstract. BPMN is suitable to model not only intra-organization workflows but also inter-organization collaborations. There has been a great effort in providing a formal semantics for BPMN, and then in building verification tools on top of this semantics. However, communication aspects are often discarded in the literature. This is an issue since BPMN has gained interest outside its original scope, e.g., for the IoT, where the configuration of communication modes plays an important role. In this paper, we propose a formal semantics for a subset of BPMN, taking into account inter-process communication and parametric verification with reference to communication modes. As opposed to transformational approaches, that map BPMN into some formal model such as transition systems or Petri nets, we give a direct formalization in First-Order Logic that is then implemented in TLA⁺ to enable formal verification. Our approach is tool supported. The tool, as well as the TLA⁺ theories, and experiment models are available online.

Keywords: BPMN · Formal semantics · Collaboration · Communication · Verification · TLA⁺ · Tool

1 Introduction

BPMN collaboration diagrams [1] provide an efficient way to describe how several business entities, each one with its own internal process, can interact with one another to reach objectives. The BPMN standard defines an execution semantics using natural language, that could be qualified as semi-formal. This leaves room for interpretation and hampers formal analysis of the models. This issue has been addressed in the last decade in different proposals for a formalization of the BPMN execution semantics (see Sect. 4), some of them with available tools. But

This work was supported by project PARDI ANR-16-CE25-0006.

© Springer Nature Switzerland AG 2019

T. Hildebrandt et al. (Eds.): BPM 2019, LNCS 11675, pp. 52–68, 2019.

https://doi.org/10.1007/978-3-030-26619-6_6

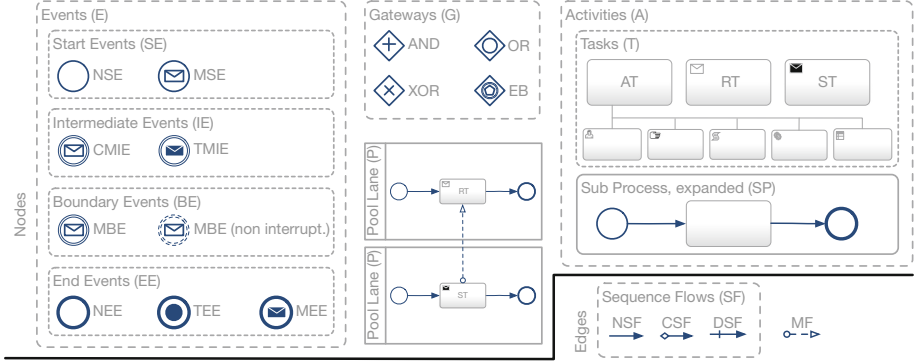


Fig. 1. BPMN subset being supported, with types used in the formalization (*e.g.*, MSE stands for Message Start Events and SE for Start Events). Due to lack of space, the support for boundary events is presented in [13].

these proposals often leave apart features related to communication. Meanwhile, BPMN is gaining interest as a modeling language for the Internet of Things (IoT) [4, 18], There, the communication between the nodes of the system, and the configuration of different communication modes, is an issue.

Contribution. The contribution of this paper is twofold. First, (1) we provide a formalization of a subset of BPMN execution semantics that *supports interaction* and that is *parametric with reference to the properties of the communication* between participants, and (2) we support this formalization with *tools that automatically perform the verification of correctness properties* for BPMN collaboration models.

As far as (1) is concerned, we chose to define a direct First-Order Logic (FOL) semantics for BPMN. Instead of using an intermediary formal model, *e.g.*, Petri nets or process algebra, this choice of a simple yet expressive framework enables one to get a formal semantics that is amenable to implementation in different formal frameworks while still being close to the semi-formal semantics of the standard (hence it can be related to it). We implement our FOL semantics in TLA^+ [17] as a set of TLA^+ theories. This corresponds to a pure syntactic transformation of FOL into the corresponding TLA^+ fragment. Our semantics supports the six point-to-point communication models that exist when considering local and global message ordering, and it is easily extensible. As far as the subset of BPMN is concerned, we have first based our choice on the analysis of the 825 BPMN processes available in the BIT process library, release 2009 [12], given in [15]. We have then taken more constructs into account, mainly relative to our focus on communication: creation and termination of processes using messages, message-related tasks and intermediary events, and event-based gateways. The whole subset of the notation that we support is given in Fig. 1.

With respect to (2), our approach relies on two steps. First, one uses the `fbpmn` tool that we have developed to get a TLA^+ representation of the model

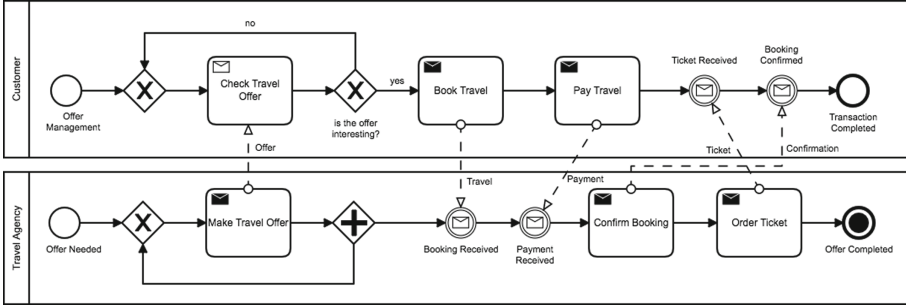


Fig. 2. Travel agency case study (slightly adapted from an example in [7]).

to verify. Then, one uses the TLC model-checker from the TLA⁺ tool-suite to perform the verification. The properties of interest are encoded in the TLA⁺ theories we have implemented. They include usual correctness properties for workflows as well as ones (proposed more recently [7]) that are more specific to BPMN. Both tools are open source and freely available online. Furthermore, the models we have used for evaluation in Sect. 3 are also available online. To get the tools and the models, see the `fbpmn` repository [3].

Due to space considerations, we assume the reader has a basic knowledge of the BPMN notation. We refer to [1] if this is not the case.

Case Study. Figure 2 presents the case study we use to illustrate some definitions and to perform verification. The outcomes of verification on more examples, including ones from the literature, are synthesized in Sect. 3.

The collaboration involves two participants: a customer and a travel agency. The agency sends offers to the client. The client may accept or decline the offers (loop using two exclusive gateways). Once it has accepted an offer, the client will not be ready to receive another one and it proceeds to the exchange part relative to the offer of interest. On the other side, the agency (through the use of a parallel gateway) is able both to send other offers and begin the exchange part: at the parallel gateway a token is generated both to get back to offer sending and to wait for the booking. This case study is interesting for several reasons. First, due to the agency behavior, the collaboration is possibly unsafe: one can have an unbounded number of tokens on the right of the parallel gateway. Second, observe that the partners do not agree on the order of confirmation *wrt.* ticket reception. Depending on the communication model, this may cause a deadlock.

Overview. The formal part of the paper is developed in Sect. 2, with Subsects. 2.1 and 2.2, respectively addressing the presentation of the model underlying the semantics, and then the semantics itself. The implementation of the semantics in TLA⁺, verification, and evaluation are then presented in Sect. 3. This section also includes a short introduction to the TLA⁺ language and verification framework. Related work is given in Sect. 4, and we end with conclusions and perspectives in Sect. 5.

2 Formal Semantics

In this section, we first present the model on which we base the definition of the communication-parametric semantics for BPMN collaborations. This model is used to represent collaborations as typed graphs. In a second step, we present the semantic itself. It follows the “token game” of the standard [1, Ch. 13], with a notion of state that evolves with activation and completion of graph nodes.

2.1 A Typed Graph Representation of BPMN Collaborations

In our work, a BPMN model is seen as a typed graph (Definition 1), where nodes and edges are associated to types corresponding to the BPMN syntax (see Fig. 1): $T_{Nodes} = \{AT, RT, ST, SP, NSE, MSE, CMIE, TMIE, NEE, TEE, MEE, AND, OR, XOR, EB, P\}$ and $T_{Edges} = \{NSF, CSF, DSF, MF\}$. The hierarchical structure of collaborations, with processes and sub-processes is dealt with by using specific types for nodes, P and SP , and a relation denoting containment, R . From our example, Fig. 2, we would then have two nodes of type P (say n_1 for Customer, n_2 for Travel Agency) with their respective contents related by R , e.g., a node n_3 for Make Travel Offer of type ST with $n_3 \in R(n_2)$.

Definition 1 (BPMN Graph). *A BPMN graph is a tuple $\widehat{G} = (N, E, \mathbb{M}, cat_N, cat_E, source, target, R, msg_t)$ where N is the set of nodes, E ($N \cap E = \emptyset$) is the set of edges, \mathbb{M} is the set of message types, $cat_N : N \rightarrow T_{Nodes}$ gives the type of a node, $cat_E : E \rightarrow T_{Edges}$ gives the type of an edge, $source/target : E \rightarrow N$ give the source/target of an edge, $R : N^{\{SP, P\}} \rightarrow 2^{N \cup E}$ gives the set of nodes and edges which are directly contained in a container (process or sub-process), and $msg_t : E^{MF} \rightarrow \mathbb{M}$ gives the message associated to a message flow.*

Notation. We use N^T (resp. E^T) to denote the subset of nodes (resp. edges) of type T , e.g., $N^T = \{n \in N \mid cat_N(n) \in T\}$. By abuse of notation, we may write N^t instead of $N^{\{t\}}$, e.g., N^{NSE} instead of $N^{\{NSE\}}$. We use for cat_E the same simplified notations as for cat_N .

We then define some auxiliary functions that will be used in the semantics.

Auxiliary Functions. For a graph $\widehat{G} = (N, E, \mathbb{M}, cat_N, cat_E, source, target, R, msg_t)$, we introduce the following auxiliary functions:

- $in/out : N \rightarrow 2^E$ give the incoming/outgoing edges of a node, $in(n) = \{e \in E \mid target(e) = n\}$ and $out(n) = \{e \in E \mid source(e) = n\}$.
- a family of functions in^T (resp. out^T) : $N \rightarrow 2^E$ is used to combine in (resp. out) with E^T , $in^T(n) = in(n) \cap E^T$ and $out^T(n) = out(n) \cap E^T$.
- $procOf : N \rightarrow N^P$ gives the container process of a given node, $procOf(n) = p$ if and only if $n \in R^+(p)$, with R^+ being the transitive closure of R .

It is desirable to enforce that models respect some well-formedness rules before performing verification. Due to a lack of space, we give the rules that we impose (most coming from [1]) for a BPMN graph to be well-formed in [13].

2.2 A FOL Semantics for BPMN Collaborations

In order to maintain traceability with the standard, we use a token-based approach. The movement of tokens is based on node types. We define an execution model based on two predicates (St , Ct) for each node type which correspond to, respectively, the enabling of the node to start its execution, and the enabling of the node to complete its execution. Some nodes only have a start transition (e.g., end events), and others only have a completion transition (e.g., gateways). The definition of these predicates relies a notion of *state* of the BPMN Graph.

Definition 2 (State). *The state of a BPMN graph is given by a couple of functions $s = (m_n, m_e)$, $m_n : N \rightarrow \mathbb{N}$ and $m_e : E \rightarrow \mathbb{N}$, that associate a number of tokens to nodes (gateways are always 0) and edges, respectively. The set of all states of a BPMN graph is denoted by *States*.*

Definition 3 (Initial state). *The initial state of a BPMN graph, denoted by $s_o = (m_{n0}, m_{e0})$, associates a token to the *NSE* nodes of the processes, all the other nodes and edges being unmarked: $\forall e \in E, m_{e0}(e) = 0$, and $\forall n \in N, m_{n0}(n) = 1$ if $\exists p \in N^P, n \in N^{NSE} \cap R(p)$, and 0 otherwise.*

The properties of communication between two participants (process nodes) for a given type of message are abstracted with two predicates, *send* and *receive*. These predicates specify when a communication action is enabled and the effect of this communication. For instance, with FIFO asynchronous communication (`NetworkFifo`, Sect. 3.3), messages must be delivered in the order they were sent. Thus $send(p_1, p_2, m)$ is always enabled and $receive(p_1, p_2, m)$ is true only if m is the oldest message and thus the next one to be delivered. Observe that the value of these predicates evolve as the processes send and receive messages.

Definition 4 (Communication Model). *The communication model is characterized by two predicates $send/receive : N^P \times N^P \times \mathbb{M} \rightarrow Bool$.*

The formal execution semantics is given in Tables 1 and 2. We consider that m_n and m'_n (resp. m_e and m'_e) denote two successive markings of a node (resp. edge) in the execution semantics. Δ is a predicate that denotes marking equality but for nodes and edges given as parameter, $\Delta(X)$ means “nothing changes but for X ”: $\Delta(X) \stackrel{def}{=} \forall n \in N \setminus X, m'_n(n) = m_n(n) \wedge \forall e \in E \setminus X, m'_e(e) = m_e(n)$.

Starting and Terminating. The behavior of an *NSE* node is defined only through completion: it consumes one token and generates one token on all its outgoing sequence flow edges. If it is the initial node of a process p , it activates it by generating a token on p . For a sub-process, it is the *SP* starting predicate that will perform activation (see below). The behavior of an *NEE* node is defined only through a starting predicate: it is enabled if it has at least one token on one of its incoming edges, that is added to the node. A *TEE* node is also defined only through a starting predicate: it is enabled if it has at least one token on one of its incoming edges, then it drops down all the remaining tokens of the process or sub-processes to which it belongs. The behavior of an activity node

Table 1. FOL semantics (part 1 – events)

Events	$n \in N^{NSE}$	$ \begin{aligned} Ct(n) \stackrel{def}{=} & (m_n(n) \geq 1) \wedge (m'_n(n) = m_n(n) - 1) \\ & \wedge (\forall e \in out^{SF}(n), (m'_e(e) = m_e(e) + 1)) \\ & \wedge ((\exists p \in N^P, n \in R(p) \wedge (m_n(p) = 0) \wedge (m'_n(p) = 1) \\ & \quad \wedge \Delta(\{n, p\} \cup out^{SF}(n))) \\ & \quad \vee (\exists p \in N^{SP}, n \in R(p) \wedge \Delta(\{n\} \cup out^{SF}(n)))) \end{aligned} $
	$n \in N^{MSE}$	$ \begin{aligned} St(n) \stackrel{def}{=} & (m_n(n) = 0) \wedge (m'_n(n) = 1) \\ & \wedge (\exists e \in in^{MF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \\ & \quad \wedge receive(procOf(source(e)), procOf(n), msg_t(e))) \\ & \quad \wedge \Delta(\{n, e\})) \end{aligned} $
		$ \begin{aligned} Ct(n) \stackrel{def}{=} & (m_n(n) = 1) \wedge (m'_n(n) = m_n(n) - 1) \\ & \wedge (\exists p \in N^P, n \in R(p) \wedge (m_n(p) = 0) \wedge (m'_n(p) = 1) \\ & \quad \wedge (\forall e \in out^{SF}(n), (m'_e(e) = m_e(e) + 1)) \\ & \quad \wedge \Delta(\{n, p\} \cup out^{SF}(n))) \end{aligned} $
	$n \in N^{TMIE}$	$ \begin{aligned} St(n) \stackrel{def}{=} & (\exists e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \\ & \quad \wedge (\exists e' \in out^{MF}(n), (m'_e(e') = m_e(e') + 1) \\ & \quad \wedge send(procOf(n), procOf(target(e')), msg_t(e'))) \\ & \quad \wedge \forall e'' \in out^{SF}(n), (m'_e(e'') = m_e(e'') + 1) \\ & \quad \wedge \Delta(\{e, e'\} \cup out^{SF}(n))) \end{aligned} $
	$n \in N^{CMIE}$	$ \begin{aligned} St(n) \stackrel{def}{=} & (\exists e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \\ & \quad \wedge (\exists e' \in in^{MF}(n), (m_e(e') \geq 1) \wedge (m'_e(e') = m_e(e') - 1) \\ & \quad \wedge receive(procOf(source(e)), procOf(n), msg_t(e'))) \\ & \quad \wedge (\forall e'' \in out^{SF}(n), (m'_e(e'') = m_e(e'') + 1)) \\ & \quad \wedge \Delta(\{e, e'\} \cup out^{SF}(n))) \end{aligned} $
	$n \in N^{NEE}$	$ \begin{aligned} St(n) \stackrel{def}{=} & (\exists e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \\ & \quad \wedge (m'_n(n) = m_n(n) + 1) \wedge \Delta(\{n, e\})) \end{aligned} $
	$\forall n \in N^{TEE}$	$ \begin{aligned} St(n) \stackrel{def}{=} & (m'_n(n) = 1) \\ & \wedge (\exists e \in in^{SF}(n), (m_e(e) \geq 1)) \\ & \wedge (\exists p \in N^{\{P, SP\}}, n \in R(p) \\ & \quad \wedge (\forall nn \in ((R^+(p) \cap N) \setminus \{n\}), m'_n(nn) = 0) \\ & \quad \wedge (\forall ee \in (R^+(p) \cap E), m'_e(ee) = 0) \\ & \quad \wedge \Delta(R^+(p))) \end{aligned} $
$n \in N^{MEE}$	$ \begin{aligned} St(n) \stackrel{def}{=} & (\exists e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \\ & \quad \wedge (\exists e' \in out^{MF}(n), send(procOf(n), procOf(target(e')), msg_t(e'))) \\ & \quad \wedge (m'_e(e') = m_e(e') + 1) \\ & \quad \wedge (m'_n(n) = m_n(n) + 1) \wedge \Delta(\{n, e, e'\})) \end{aligned} $	

AT is defined by a starting and a completion predicate. The node is started by the arrival of at least one token on one of its incoming edges. Completion is realized by adding one token on each of its outgoing edges. The behavior of a *SP* node extends the one of an *AT* node with additional conditions: when enabled, a sub-process adds a token to the start event it contains. It completes when at least one end event it contains has some tokens and neither one of its edges nor one of its non end event nodes is still active (*i.e.*, owning a token).

Table 2. FOL Semantics (part 2 – gateways and activities)

Gateways	$n \in N^{XOR}$	$Ct(n) \stackrel{def}{=} (\exists e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \wedge (\exists e' \in out^{SF}(n), (m'_e(e') = m_e(e') + 1)) \wedge \Delta(\{e, e'\}))$
	$n \in N^{AND}$	$Ct(n) \stackrel{def}{=} (\forall e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1)) \wedge (\forall e' \in out^{SF}(n), (m'_e(e') = m_e(e') + 1)) \wedge \Delta(in^{SF}(n) \cup out^{SF}(n))$
	$n \in N^{OR}$	$Ct(n) \stackrel{def}{=} (In^+(n) \neq \emptyset) \wedge (\forall e \in In^+(n), (m'_e(e) = m_e(e) - 1)) \wedge (\forall ez \in In^-(n), \forall ee \in (Pre_E(n, ez) \setminus ignore_E(n)), (m_e(ee) = 0) \wedge (\forall nn \in (Pre_N(n, ez) \setminus ignore_N(n)), (m_n(nn) = 0))) \wedge ((\exists Outs \subset (out^{SF}(n) \cap E^{\{NSF, CSF\}}), (Outs \neq \emptyset) \wedge (\forall e \in Outs, (m'_e(e) = m_e(e) + 1) \wedge \Delta(In^+(n) \cup Outs)) \vee (\exists e \in out^{DSF}(n), (m'_e(e) = m_e(e) + 1) \wedge \Delta(In^+(n) \cup \{e\}))))$
	$n \in N^{EB}$	$Ct(n) \stackrel{def}{=} (\exists e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \wedge (\exists e' \in out^{SF}(n), \exists e'' \in in^{MF}(target(e')), (m_e(e'') \geq 1) \wedge (m'_e(e') = m_e(e') + 1) \wedge \Delta(e, e')))$
Activities	$n \in N^{AT}$	$St(n) \stackrel{def}{=} (\exists e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \wedge (m'_n(n) = m_n(n) + 1) \wedge \Delta(\{n, e\}))$
		$Ct(n) \stackrel{def}{=} (m_n(n) \geq 1) \wedge (m'_n(n) = m_n(n) - 1) \wedge (\forall e \in out^{SF}(n), (m'_e(e) = m_e(e) + 1)) \wedge \Delta(\{n\} \cup out^{SF}(n))$
	$n \in N^{ST}$	$St(n) \stackrel{def}{=} (\exists e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \wedge (m'_n(n) = m_n(n) + 1) \wedge \Delta(\{n, e\}))$
		$Ct(n) \stackrel{def}{=} (m_n(n) \geq 1) \wedge (m'_n(n) = m_n(n) - 1) \wedge (\forall e \in out^{SF}(n), (m'_e(e) = m_e(e) + 1)) \wedge (\exists e' \in out^{MF}(n), send(procOf(n), procOf(target(e')), msg_t(e')) \wedge (m'_e(e') = m_e(e') + 1) \wedge \Delta(\{n, e'\} \cup out^{SF}(n)))$
	$n \in N^{RT}$	$St(n) \stackrel{def}{=} (\exists e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \wedge (m'_n(n) = m_n(n) + 1) \wedge \Delta(\{n, e\}))$
		$Ct(n) \stackrel{def}{=} (m_n(n) \geq 1) \wedge (m'_n(n) = m_n(n) - 1) \wedge (\forall e \in out^{SF}(n), (m'_e(e) = m_e(e) + 1)) \wedge (\exists e' \in in^{MF}(n), (m_e(e') \geq 1) \wedge (m'_e(e') = m_e(e') - 1) \wedge receive(procOf(source(e')), procOf(n), msg_t(e')) \wedge \Delta(\{n, e'\} \cup out^{SF}(n)))$
$n \in N^{SP}$	$St(n) \stackrel{def}{=} (\exists e \in in^{SF}(n), (m_e(e) \geq 1) \wedge (m'_e(e) = m_e(e) - 1) \wedge (m'_n(n) = m_n(n) + 1) \wedge (\forall n_{se} \in (N^{SE} \cap R(n)), (m'_n(n_{se}) = m_n(n_{se}) + 1)) \wedge \Delta(\{e, n\} \cup (N^{SE} \cap R(n))))$	
	$Ct(n) \stackrel{def}{=} (m_n(n) \geq 1) \wedge (m'_n(n) = 0) \wedge (\forall e \in R(n) \cap E, (m_e(e) = 0)) \wedge (\exists n_{ee} \in (N^{EE} \cap R(n)), (m_n(n_{ee}) \geq 1)) \wedge (\forall nn \in R(n) \cap N, (m_n(nn) \geq 1 \Rightarrow nn \in N^{EE})) \wedge (\forall nn \in (R(n) \cap N^{EE}), (m'_n(nn) = 0)) \wedge (\forall e \in out^{SF}(n), (m'_e(e) = m_e(e) + 1)) \wedge \Delta(\{n\} \cup (R(n) \cap N^{EE}) \cup out^{SF}(n))$	

Communication. The communication elements (MSE , $TMIE$, $CMIE$, MEE , ST , RT) require additional conditions for starting and completing due to the presence of sending/reception behaviors. When enabled by a token on one of their incoming edges, $TMIE$, MEE , and ST send a message on their outgoing message flow and a token on all their outgoing sequence flows. MSE , $CMIE$, and RT require a message offer on one of their incoming message flows. They receive the message and produce tokens on their outgoing edges. MSE and RT have both starting and completion transitions while $CMIE$ is an instantaneous event with only a starting transition.

Gateways. Gateways are atomic and define only the completion behavior. An AND gateway is ready to complete if it has at least one token on all its incoming edges. It completes by removing one token on each of these edges, and producing one on all its outgoing edges. An XOR gateway is ready to complete if it has at least one token on one of its incoming edges. It completes by removing this token, and producing one on one of its outgoing edges, depending on conditions. Since we abstract from data, we proceed choosing non-deterministically the concerned edge. An EB gateway behaves as an XOR gateway in that it consumes and produces a token from and to only one edge. However, its completion relies on the presence of external message triggers. The outgoing edge activation depends on the enabledness of the $CMIE$ or RT which is the target of this edge. The activation of an OR gateway g is more complex [1, Chap. 13]. It is activated if (1) it has at least one token on one of its incoming edges, and (2) for each node or edge x such that there is a path – that does not pass through g – from x to an unmarked incoming edge of g , there must be also a path – that does not pass through g – from x to a marked incoming edge of g . This is illustrated in Fig. 3 where gateway g cannot be activated since there is a path from marked e^{pre} to unmarked e^- and no path from e^{pre} to some marked e^+ . The OR gateway completes by adding a token either to all its outgoing edges whose condition it true, or else to its default sequence flow edge. Again, since we abstract from data, we chose non-deterministically to add a token either to a combination (1 or more) of the outgoing non default edges, or to the default edge.

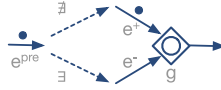


Fig. 3. Non activable inclusive gateway. It has to wait for the token on e^{pre} .

To formalize the semantics of an OR gateway we use several functions:

- $Pre_N : N \times E \rightarrow 2^N$ gives the predecessor nodes of an edge such that n^{pre} is in $Pre_N(n, e)$ if there is a path from n^{pre} to e that never visits n . Accordingly, $Pre_E : N \times E \rightarrow 2^E$ gives predecessor edges.
- $In^- : N \rightarrow E$ gives the unmarked incoming edges of a node, $In^-(n) = \{e^- \in in^{SF}(n) \mid m_e(e^-) = 0\}$.

- $In^+ : N \rightarrow E$ gives the marked incoming edges of a node, $In^+(n) = \{e^+ \in in^{SF}(n) \mid m_e(e^+) \geq 1\}$.
- $ignore_E : N \rightarrow 2^E$ gives the predecessor edges of the marked incoming edges of a given node: $ignore_E(n) \stackrel{def}{=} \bigcup_{e^+ \in In^+(n)} Pre_E(n, e^+)$.
- $ignore_N : N \rightarrow 2^N$ gives the predecessor nodes of the marked incoming edges of a given node: $ignore_N(n) = \bigcup_{e^+ \in In^+(n)} Pre_N(n, e^+)$.

3 Implementation and Verification

In this section, we present our encoding of the FOL semantics in TLA^+ . This allows one to easily parameter the properties of the communication, and to benefit from the efficient TLC model checker to automatically verify collaborations.

3.1 The TLA^+ Specification Language and Verification Framework

TLA^+ [17] is a formal specification language based on untyped Zermelo-Fraenkel set theory for specifying data structures, and on the temporal logic of actions (TLA) for specifying dynamic behaviors. TLA^+ allows one to specify symbolic transition systems with variables and *actions*. An action is a transition predicate between a state and a successor state. It is an arbitrary first-order predicate with quantifiers, set and arithmetic operators, and functions. In an action, x denotes the value of a variable x in the origin state, and x' denotes its value in the next state. Functions are primitive objects in TLA^+ . The application of function f to an expression e is written as $f[e]$. The expression $[x \in X \mapsto e]$ denotes the function with domain X that maps any $x \in X$ to e . The expression $[f \text{ EXCEPT } ![e_1] = e_2]$ is a function that is equal to the function f except at point e_1 , where its value is e_2 . A system specification is usually a disjunction of actions. Fairness, usually expressed as a conjunction of weak or strong fairness on actions, or more generally as an LTL (Linear Temporal Logic) property, ensures progression. The TLA^+ toolbox, freely available at <http://lampton.azurewebsites.net/tla/tla.html>, contains the TLC model checker, the TLAPS proof assistant, and various other tools.

3.2 Encoding of FOL Semantics in TLA^+

The expression and action fragment of TLA^+ contains FOL, and the encoding of the semantics in TLA^+ is straightforward (459 lines of TLA^+ formulae). The resulting theories are available in the fbpmn repository [3] under `theories/tla`.

Module `PWSTypes` defines the abstract constants that correspond to the node and edge types. Module `PWSDefs` specifies the constants that describe a BPMN graph (Definition 1): `Node` (for N), `Edge` (for E), `Message` (for \mathbb{M}), `CatN` (for cat_N), `CatE` (for cat_E), `ContainRel` (for R)... This module also defines auxiliary functions such as in^T . Module `PWSWellFormed` encodes the well-formedness predicates for BPMN graphs. Last, module `PWSSemantics` contains the semantics. It defines the variables for the marking: `nodemarks` ($\in [Node \rightarrow Nat]$) and `edgemarks` ($\in [Edge \rightarrow Nat]$). Then it contains a translation of the FOL formulas given in Tables 1 and 2. Each rule yields one TLA⁺ action. For instance, the St predicate of the SP nodes becomes:

$$\begin{aligned}
& subprocess_start(n) \triangleq \\
& \quad CatN[n] = SubProcess \\
& \quad \wedge \exists e \in intype(SeqFlowType, n) : edgemarks[e] \geq 1 \\
& \quad \quad \wedge edgemarks' = [edgemarks \text{ EXCEPT } ![e] = edgemarks[e] - 1] \\
& \quad \wedge nodemarks' = [nn \in DOMAIN nodemarks \mapsto \\
& \quad \quad \text{IF } nn = n \text{ THEN } nodemarks[nn] + 1 \\
& \quad \quad \text{ELSE IF } CatN[nn] \in StartEventType \wedge nn \in ContainRel[n] \\
& \quad \quad \quad \text{THEN } nodemarks[nn] + 1 \\
& \quad \quad \quad \text{ELSE } nodemarks[nn]] \\
& \quad \wedge Network!unchanged
\end{aligned}$$

The $Next$ predicate specifies a possible transition between a starting state and a successor state. It is a disjunction of all the actions. The full specification is then, as usual in TLA⁺, $Init \wedge \square[Next]_{vars} \wedge Fairness$, where $Init$ specifies the initial state (Definition 3), and $\square[Next]$ specifies that $Next$ (or stuttering) is verified along all the execution steps. In our case, $Fairness$ is a temporal property that ensures that any permanently enabled transition eventually occurs. This means that no process may progress forever while others are never allowed to do so if they can. Moreover, we include in the fairness that no choice is infinitely often ignored: if a XOR , OR , or EB gateway is included in a loop, the fairness forbids the infinite executions that never use some output edges.

3.3 Communication as a Parameter

One of the objectives of our FOL semantics is to be able to specify the communication behavior as a parameter of the verification. To achieve this, all operations related to communication are isolated in a `Network` module. This module is a proxy for several implementations that correspond to communication models with different properties, such as their delivery order.

We provide six communication models which differ in the order messages can be sent or received, and are all the possible point-to-point models when considering local ordering (per process) and global ordering (absolute time). They are formally defined in [5] and informally are: unordered (modelled by a bag of message), first-in-first-out between each couple of processes (modelled by an array of queues), fifo inbox (each process has an input queue where senders add their messages), fifo outbox (each process has an output queue from where receivers fetch messages), global fifo (a unique shared queue), and RSC (realizable with synchronous communication, modelled as a unique message slot that forces alternation of send and receive tasks). If the collaboration is sound, no message is left in transit, and RSC and synchronous communication yield the same behaviors.

The state of the communication model is specified with a variable `net`, whose content depends on the communication model. The communication actions are two transition predicates `send` and `receive` which are true when the action is enabled. These actions take three parameters, the sender process, the destination process and the message. Their specification depends on the communication model. For instance, `NetworkFifo` specifies a communication model where the delivery order is globally first-in-first-out: messages are delivered in the order they have been sent. Its realization is a queue and the two predicates are:

$$\begin{aligned} \text{send}(\text{from}, \text{to}, m) &\triangleq \text{net}' = \text{Append}(\text{net}, \langle \text{from}, \text{to}, m \rangle) \\ \text{receive}(\text{from}, \text{to}, m) &\triangleq \text{net} \neq \langle \rangle \wedge \langle \text{from}, \text{to}, m \rangle = \text{Head}(\text{net}) \wedge \text{net}' = \text{Tail}(\text{net}) \end{aligned}$$

3.4 Mechanized Verification

A specific BPMN diagram is described by instantiating the constants in `PWSDefs` (`Node`, `Edge`...) from the BPMN collaboration. This is automated using our `fbpmn` tool. Regarding the well-formedness of the BPMN diagram, the predicates from `PWSWellFormed` are *assumed* in the model. Before checking a model, The TLA^+ model checker checks these assumptions with the instantiated constants that describe the diagram, and reports an error if an assumption is violated. Otherwise, this proves that the diagram is well-formed.

The TLA^+ model checker, `TLC`, is an explicit-state model checker that checks both safety and liveness properties specified in *LTL*. This logic includes operators \square and \diamond that respectively denote that, in all executions, a property F must always hold ($\square F$) or that it must hold at some instant in the future ($\diamond F$). `TLC` builds and explores the full state space of the diagram to verify if the given properties are verified. These properties are generic properties related to any Business Process diagram, or specific properties for a given diagram. Some of the generic properties are safe collaboration, sound collaboration and message-relaxed sound collaboration [7]. A collaboration is safe if no sequence flow holds more than one token:

$$\square(\forall e \in E^{SF}, m_e(e) \leq 1) \tag{1}$$

A collaboration is sound if all processes are sound and there are no undelivered messages. A process is sound if there are no token on its inside edges, and one token only on its end events.

$$\begin{aligned} \text{SoundProc}(p) &\stackrel{\text{def}}{=} \forall e \in R(p) \cap E^{SF}, m_e(e) = 0 \\ &\quad \wedge \forall n \in R(p) \cap N, (m_n(n) = 0 \vee (m_n(n) = 1 \wedge n \in N^{EE})) \\ \text{Soundness} &\stackrel{\text{def}}{=} \diamond(\forall p \in N^P, \text{SoundProc}(p) \wedge \forall e \in E^{MF}, m_e(e) = 0) \quad (2) \end{aligned}$$

A collaboration is message-relaxed sound (3) if it is sound when ignoring messages in transit, *i.e.*, when ignoring the Message Flow edges. This is the same as (2) without the second conjunction.

Other generic properties are available, such as the absence of undelivered message or the possible activation which states there does not exist a task node (Abstract Task, Send Task, Receive Task) that is never activated in any execution. From a business process point of view, it means that there are no tasks in the diagram that are never used. This is expressed as $\forall n \in N^T : \mathbf{EF}(m_n(n) \neq 0)$ (TLC can check for the invalidity of the negation of this CTL formula).

Last, the user can also define business model properties concerning a specific diagram. For instance, one can check that the marking of a given node is bounded by a constant (*i.e.*, $\square(\text{nodemarks}[\text{"Confirm Booking"}]) \leq 1$), or that the activation of one node necessarily leads to the activation of another node ($\square(\text{nodemarks}[\text{"Book Travel"}] \neq 0 \Rightarrow \diamond(\text{nodemarks}[\text{"Offer Completed"}] \neq 0))$).

When the model checker finds that a property is invalid, it outputs a counterexample trace that we animate on the BPM graphical model to help the user understand it. As TLC uses a breadth-first algorithm, this trace is minimal for safety properties.

3.5 Experiments

Experiments were conducted on a laptop with a 2.1 GHz Intel Core i7 processor (quad core) with 8 GB of memory. Results are presented in Table 3. The first column is the reference of the example in our archive. The characteristics of a model are: number of participants, number of nodes (incl. gateways), number of flow edges (sequence or message flows), whether the model is well-balanced (for each gateway with n diverging branches we have a corresponding gateway with n converging branches) and whether it includes a loop. The communication model is asynchronous (bag), fifo-ordered between each couple of processes (fifo pair), globally fifo (fifo all), or synchronous-like (RSC). The results of the verification then follow. First, data on the resulting transition system are given: number of states, number of transitions, and depth (length of the longest sequence of transitions that the model checker had to explore). For each of the three correctness properties presented above, we indicate if the model satisfies it. Lastly, the accumulated time for the verification of the three properties is given. Our tool supports more verifications (see Table 4) and can be easily extended with new properties. We selected these three ones since they are more BPMN specific [7].

Table 3. Experimental Results.

ref.	Characteristics					Com. model	LTS size			validity			total time
	proc.	nodes (gw.)	SF/MF	B	L		states	trans.	depth	(1)	(2)	(3)	
001	2	17 (2)	14/3	✓	×	bag	93	173	25	✓	✓	✓	5.17s
						fifo pair	85	161	21	✓	×	×	5.19s
						RSC	77	147	19	✓	×	×	5.09s
002	2	16 (2)	13/3	✓	×	bag	83	154	24	✓	✓	✓	4.93s
						fifo pair	75	142	20	✓	×	×	4.74s
						RSC	67	128	18	✓	×	×	4.48s
003	1	14 (6)	16/0	×	✓	none	41	59	15	×	✓	✓	2.89s
006	2	20 (4)	18/5	×	✓	bag	470	966	43	×	×	✓	8.98s
						fifo all	522	932	40	×	×	×	8.58s
						RSC	247	420	38	×	×	×	6.58s
007	1	8 (2)	7/0	×	×	none	44	73	15	×	×	×	2.52s
008	1	11 (2)	9/0	×	×	none	48	77	19	×	✓	✓	2.60s
009	2	12 (2)	9/1	×	×	bag	170	395	19	×	×	×	4.86s
010	2	15 (2)	11/1	×	×	bag	186	423	23	×	×	✓	5.32s
011	2	15 (2)	11/1	×	×	bag	100	209	21	×	✓	✓	4.73s
012	1	15 (8)	17/0	✓	✓	none	71	137	15	✓	✓	✓	3.16s
013	1	17 (8)	21/0	✓	✓	none	407	1049	15	✓	✓	✓	5.93s
018	1	19 (8)	25/0	✓	✓	none	4631	15513	18	✓	✓	✓	28.46s
015	2	14 (2)	10/2	×	×	bag	68	117	11	✓	×	×	4.67s
016	2	14 (2)	10/2	×	×	bag	36	53	11	✓	✓	✓	4.25s
017	1	32 (12)	36/0	×	×	none	93	141	37	✓	✓	✓	4.03s
020	4	39 (6)	34/8	×	✓	bag	4648	14691	54	✓	✓	✓	53.05s
						fifo all	2564	6872	54	✓	✓	✓	28.25s
						RSC	1224	3271	54	✓	×	×	18.77s

Table 3 presents the results for a selection from our repository [3] for a variety of gateways and activities. These illustrative examples include realistic business process models (001 and 002 two client-supplier models, 006 from Fig. 2, 017 from [15], and 020 from [9]), and models dedicated to specific concerns: termination end events and sub processes (007–011 from [7]), inclusive gateways (003, 012, 013 and 018), exclusive and event-based gateways (015 and 016).

A first conclusion is that verification is rather fast: the verification of one property generally takes just a few seconds per model, the longest being for model 020 that takes up to 53s of accumulated time for the three properties (5s for the construction of the state space). Experiments also show the effect of the communication model on property satisfaction (models 1, 2, 6, 20), the use of TLA⁺ fairness to avoid infinite loops (12, 13, 18, 20), and the use of terminate end events combined with user given constraints [3] to deal with unsafety (6).

Table 4. Comparison of tool-supported approaches for the analysis of communication in BPMN.

approach	transformation						direct					
	[10]	[14]	[20]	[21]	[22]	[11]	[9]	[7]	[8]	[16]	ours	
reference year	2018	2017	2013	2011	2008	2008	2008	2018	2012	2012	2019	
formalism	CPN	ECATNets	In-Place	CSP	YAWL Nets	RDP		LTS	LTL	FOL		
tool	not avail. CP4PBMN	not avail. BPMNChecker	yes GrGen	yes OWorkflow	yes BPMN2YAWL	yes transformer	yes MIDA	yes BPProVe	not avail. prototype	yes fBPMN		
<i>AND</i> gateway	•	•	•	•	•	•	•	•	•	•	•	
<i>XOR</i> gateway	•	•	•	•	•	•	•	•	•	•	•	
<i>OR</i> gateway	•	–	•	•	•	•	•	•	•	•	•	
<i>EB</i> gateway	•	•	•	–	•	•	•	•	•	•	•	
<i>ST</i>	•	•	•	•	•	•	•	•	•	•	•	
<i>RT</i>	•	•	•	•	•	•	•	•	•	•	•	
<i>MSE</i>	•	•	•	–	•	•	•	•	•	•	•	
<i>TMIE</i>	•	–	•	–	–	•	•	•	•	•	•	
<i>CMIE</i>	•	–	•	–	–	•	•	•	•	•	•	
<i>MEE</i>	•	•	•	–	–	–	•	•	•	•	•	
<i>TEE</i>	–	–	•	•	–	–	•	•	•	•	•	
sub-processes	•	•	•	–	•	•	•	–	–	•	•	
data	•	•	•	–	–	–	•	–	–	–	–	
multi instance (pools)	–	–	•	–	–	–	•	–	–	–	–	
multi instance(activities)	•	•	•	•	•	•	•	–	–	–	–	
option to compl.	–	•	–	–	•	–	–	–	–	•	•	
proper compl.	–	–	–	–	•	•	–	–	–	•	•	
no dead activity	–	–	–	–	•	•	–	–	–	•	•	
safety	–	–	–	•	–	–	–	•	•	•	•	
collaboration soundness	–	–	–	–	–	–	–	•	–	–	•	
msg relaxed soundness	–	–	–	–	–	–	–	•	–	–	•	
undelivered msg	–	–	–	–	–	–	–	–	–	–	•	
general-purp ose	•	•	•	•	•	•	–	•	•	•	•	
communication models	–	–	–	–	–	–	–	–	–	–	•	

supported elements

verification

4 Related Work

The formalization of the BPMN execution semantics, and on a wider scale the formal study of business processes, is a very active field of research. We here focus on recent work that provides tool support for the verification of collaboration diagrams and communication features in BPMN [7–10, 14, 16, 20]. We add [11] due to its role as a seminal paper and [21, 22] as recent representatives for the formal model they use. Table 4 gives a synthetic presentation of a comparison between these proposals and ours.

Some works are based on transformations, while others provide a direct semantics. The former rely on an intermediary model, while the later have the benefit to provide a direct link between BPMN constructs and the verification formalism. Our work is in this line. Further, the choice of FOL lets one implement the semantics in different tools, *e.g.*, TLA⁺ as here or SMT solvers.

As far as the BPMN coverage criteria is concerned, we can observe that we are among the approaches with a high coverage. To make verification tractable, we have abstracted from the data and the multi-instance constructs, that are often related to data. Works taking data into account in verification either require to bound domains or operate on the basis of configurations (a state and a substitution from variables to closed terms). The former is still subject to state-space explosion, while the later is closer to animation than to full-fledged verification, and makes it impossible to verify a model for any possible initial value of the data. A perspective of our work is to rely on symbolic techniques instead [19].

The work in [9] provides a very rich formal semantics with animation capacities but does not enable verification in the large. Most of the work, still, support the verification of BP correctness properties or, at least, all-purpose formal properties (reachability, deadlock). The use of FOL to define the semantics, and its implementation in TLA⁺, made it possible for us to implement quite easily these correctness checks as temporal logic properties to be checked against the model.

5 Conclusion and Future Work

In this work we have proposed a formalization of a subset of the BPMN execution semantics. It supports interaction and is parametric with reference to the properties of the communication between participants. We have seen in Table 3 that these properties can have an effect on the satisfaction of correctness properties by the collaboration model. Communication-parametric verification techniques for BPMN are therefore helpful when it comes to use this standard in contexts such as the IoT [4, 18] where communication may vary. Our proposal is equipped with open source tools that are freely available and automatically perform the transformations and verification steps. A direct perspective of this work is the integration of our proposal as a plug-in of a platform for business processes that goes beyond modeling, namely ProM [2].

An on-going work is to replace the network theories that specify the communication model with a more general and versatile solution based on a communication framework we have developed [6]. This framework allows a large variety of

configurations for the communication model, including which ordering policies are to be applied per participant, per couples of communicating participants, using priorities, or bounds on the number of messages in transit.

Some BPMN features that play a role in full-fledged executable collaboration have been discarded, in a first step, when selecting a relevant BPMN subset. This is the case of the support for data and the multi-instances activity, that are supported for example in [9] to provide the business process designers with model animation. However, to keep verification tractable, we had to abstract from these features. Our main perspective goes in this direction, using techniques from symbolic execution and symbolic transition systems [19] while keeping a direct semantics, *i.e.*, relatable to the parts of the BPMN informal semantics.

References

1. Business Process Modeling Notation. <http://www.omg.org/spec/BPMN/2.0.2/>
2. ProM Framework. <http://www.processmining.org/prom>
3. fbpmn repository. <https://github.com/pascalpoizat/fbpmn>, v0.3.0 (c389b6d)
4. Casati, F., et al.: Towards business processes orchestrating the physical enterprise with wireless sensor networks. In: Proceedings of ICSE (2012)
5. Chevrou, F., Hurault, A., Quéinnec, P.: On the diversity of asynchronous communication. *Form. Asp. Comput.* **28**(5), 847–879 (2016)
6. Chevrou, F., Hurault, A., Quéinnec, P.: A Modular Framework for Verifying Versatile Distributed Systems. *Journal of Logical and Algebraic Methods in Programming* (to appear)
7. Corradini, F., et al.: A classification of BPMN collaborations based on safeness and soundness notions. In: Proceedings of EXPRESS/SOS (2018)
8. Corradini, F., et al.: A formal approach to modeling and verification of business process collaborations. *Sci. Comput. Program.* **166**, 35–70 (2018)
9. Corradini, F., Muzi, C., Re, B., Rossi, L., Tiezzi, F.: Animating multiple instances in BPMN collaborations: from formal semantics to tool support. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *BPM 2018*. LNCS, vol. 11080, pp. 83–101. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_6
10. Dechsupa, C., Vatanawood, W., Thongtak, A.: Transformation of the BPMN design model into a colored Petri net using the partitioning approach. *IEEE Access* **6**, 38421–38436 (2018)
11. Dijkman, R.M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in BPMN. *Inf. Softw. Technol.* **50**(12), 1281–1294 (2008)
12. Fahland, D., et al.: Instantaneous soundness checking of industrial business process models. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) *BPM 2009*. LNCS, vol. 5701, pp. 278–293. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03848-8_19
13. Houhou, S., et al.: A first-order logic semantics for communication-parametric BPMN collaborations (extended version). <http://pardi.enseeiht.fr/BPM19>
14. Kheldoun, A., Barkaoui, K., Ioualalen, M.: Formal verification of complex business processes based on high-level Petri nets. *Inf. Sci.* **385**, 39–54 (2017)
15. Krishna, A., Poizat, P., Salaün, G.: Checking business process evolution. *Sci. Comput. Program.* **170**, 1–26 (2019)
16. Lam, V.S.: A precise execution semantics for BPMN. *Int. J. Comput. Sci.* **39**(1), 20–33 (2012)

17. Lamport, L.: *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison Wesley, Boston (2002)
18. Meyer, S., Ruppen, A., Hilty, L.: The things of the internet of things in BPMN. In: Persson, A., Stirna, J. (eds.) *CAiSE 2015. LNBIP*, vol. 215, pp. 285–297. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19243-7_27
19. Nguyen, H.N., Poizat, P., Zaïdi, F.: A symbolic framework for the conformance checking of value-passing choreographies. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) *ICSOC 2012. LNCS*, vol. 7636, pp. 525–532. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34321-6_36
20. Van Gorp, P., Dijkman, R.: A visual token-based formalization of BPMN 2.0 based on in-place transformations. *Inf. Softw. Technol.* **55**(2), 365–394 (2013)
21. Wong, P.Y., Gibbons, J.: Formalisations and applications of BPMN. *Sci. Comput. Program.* **76**(8), 633–650 (2011)
22. Ye, J., Sun, S., Song, W., Wen, L.: Formal semantics of BPMN process models using YAWL. In: *Proceedings of IITA* (2008)



Modeling and Enforcing Blockchain-Based Choreographies

Jan Ladleif¹(✉), Mathias Weske¹, and Ingo Weber²

¹ Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
{jan.ladleif,mathias.weske}@hpi.de

² Technische Universität Berlin, Berlin, Germany
ingo.weber@tu-berlin.de

Abstract. Distributed Ledger Technology (DLT) and blockchains in particular have been identified as promising foundations to realize inter-organizational business processes. Capabilities such as shared data and decision logic defined as smart contracts open up entirely new ways to implement process choreographies. However, current choreography modeling languages solely focus on direct interactions between organizations; they do not take into account the conceptually new features of blockchains, like shared data and smart contracts. To bridge the gap between choreography modeling and implementation, this paper critically analyzes the assumptions of choreography languages. We propose new language concepts specifically targeting blockchain capabilities, and we define their operational semantics. Our work is evaluated with a proof-of-concept implementation and an analysis of three real-world case studies from the private and the corporate sectors.

Keywords: Choreography · Blockchain · Interacting processes · BPMN

1 Introduction

Recent developments in blockchain technology promise to fundamentally impact various domains. Business Process Management (BPM) is no exception. It has been shown that decision management [4], model design [15] or process execution [6, 16] can be facilitated using smart contracts, a type of user-defined program running on a blockchain system. Especially when mutually distrustful participants want to cooperate, the trustless architecture of blockchains unfolds its potential. This has made the blockchain a central piece of interest for inter-organizational processes, which can be modeled as choreographies. Issues such as enactment, enforcement and monitoring that have largely been solved for the intra-organizational case [18] could now be solved for choreographies as well [17].

A number of challenges towards that goal have been identified [9]. Most notably, there is no consensus on the level of abstraction required for the design of blockchain-based choreographies. Established and widely used standards like

Business Process Model and Notation (BPMN) 2.0 collaboration diagrams are not well equipped to express choreographies in a succinct fashion. A more promising solution is provided by Business Process Model and Notation (BPMN) 2.0 choreography diagrams [13]. Abstracting from internal orchestration details, choreography diagrams represent business contracts on a purely interaction-centric level. As such, they are emerging as a prime contender for modeling processes in a way befitting the blockchain [9].

However, choreography diagrams are widely regarded to be of a purely descriptive nature. Lacking a central orchestrator, they also lack the executable character that model-driven process enactment approaches aim for. Further, they do not reflect important blockchain capabilities such as on-chain data storage, or shared logic. Local observability and vague ownership assumptions make it hard to properly represent a participant’s responsibilities. These limitations are obstacles in establishing choreography diagrams as the standard for blockchain-based choreography design.

In this paper, we critically analyze and question the limitations and decisions in the BPMN 2.0 choreography standard against the possibilities given by blockchain technology. We provide suggestions to refine and extend the standard where suitable, so as to elevate choreography diagrams to be enforceable artifacts using execution semantics founded in blockchain technology. We developed a proof-of-concept implementation, and use it to test the expressiveness, practical feasibility, and a number of properties of our proposal.

The paper is structured as follows: We start by establishing the basic concepts of BPMN 2.0 choreography diagrams, how they pertain to the blockchain, and discuss prior work in Sect. 2. Our approach is introduced in Sect. 3. We evaluate and discuss our contribution in Sect. 4 before concluding in Sect. 5.

2 Background and Motivation

In this section, we will introduce the necessary background information in choreography modeling and how blockchain and smart contract technology ultimately motivates our approach. We will also give an overview of relevant related work on choreography modeling and enactment.

2.1 Choreography Modeling

BPMN 2.0 choreography diagrams model the interactions between two or more individual parties to reach a common business goal [13]. The interfaces and communication between the participants are in focus rather than the work done within them. In that sense, choreography diagrams represent business contracts specifying the expected behavior of each participant.

For the remainder of this paper, we will use the running example given in Fig. 1. The choreography models a part of the rental process for private residences, specifically the interactions around bond refund that follow the termination of a tenancy. A rental bond is a deposit of the tenant to cover any damage

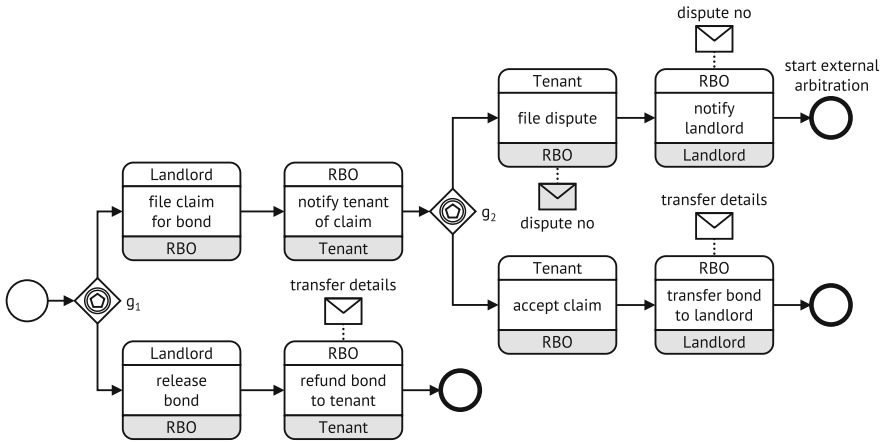


Fig. 1. A BPMN 2.0 choreography diagram of the interactions between a landlord, a tenant and a rental bond agency for rental bond handling after tenancy termination

to the property, and the process is modeled based on guidelines by the state of New South Wales, Australia.¹ A government service called Rental Bond Online (RBO) lodges the bond at the start of the tenancy. After the termination, the landlord may either release the bond and trigger its refund to the tenant, or file a claim for the bond in which case the tenant gets notified. The latter case is further subdivided: the tenant may file a dispute, which starts an external arbitration process; otherwise the bond is transferred to the landlord.

The example shows the main components of BPMN 2.0 choreography diagrams. Each choreography task (rounded rectangles) represents an interaction between an initiator and a respondent (shaded background). An initiating message exchange from the initiator to the respondent is always implied and may be refined by an attached envelope. A response message may be indicated with a shaded envelope. Choreography tasks are connected via sequence flows which incorporate gateways and events similar to BPMN 2.0 collaboration diagrams [13].

When modeling choreographies, two limiting criteria have to be kept in mind: ownership (who enforces which element) and observability (who knows what). Since “there is no central controller, responsible entity, or observer” [13, p. 23] in standard BPMN 2.0 choreography diagrams, all elements need to be owned by one specific participant. In the running example (see Fig. 1), for instance, the event-based gateways are implicitly owned by the landlord (g_1) and the tenant (g_2), respectively, based on a set of rules defined by the standard. The modeler must make sure that a participant responsible for a diagram element has sufficient information to correctly enforce it by being included in all relevant previous interactions.

¹ <https://www.fairtrading.nsw.gov.au/housing-and-property/renting/rental-bonds-online>, accessed 2019-03-14.

Another limitation is the absence of concepts like data objects which are frequently used in process diagrams. The standard pragmatically states that “there is no mechanism for maintaining any central process (choreography) data” [13, p. 319]. Choreography diagrams follow an interaction-centric paradigm, and storage of choreography data was not considered for conceptual reasons.

2.2 Choreographies and Blockchain

Since its inception for Bitcoin [12], blockchain technology has seen a rapid development. Conceptually, “a blockchain is [an append-only store of transactions which is distributed across many machines] that is structured into a linked list of blocks. Each block contains an ordered set of transactions.” [19, Ch. 1] In their core, most blockchains share a set of fundamental properties such as immutability, integrity and transparency, largely eliminating trust issues between cooperating parties [19, Ch. 3]. Various configuration options enable users to tailor blockchains to their needs, e.g., incorporating access permissions or off-chain data storage. Second-generation blockchains like Ethereum expand on the core idea and cater for expressive smart contracts, which are deterministic “programs deployed as data in the blockchain ledger and executed in transactions on the blockchain.” [19, Ch. 1] Since it is ledger data, smart contract code is immutable once deployed.

As such, a blockchain architecture may provide the crucial components needed to eventually facilitate secure and trustless inter-organizational process enactment. This is highlighted by Mendling et al. in their discussion of opportunities and challenges for BPM with blockchains [9]. They state that the “whole area of choreographies may be re-vitalized by [blockchain] technology.” While BPMN 2.0 choreography diagrams have not found wide-spread industry adoption, Mendling et al. still consider them one of the promising approaches for designing inter-organizational business processes. The limitations in expressiveness regarding ownership and observability mentioned in the previous section still present a major obstacle, even though they can be mostly lifted when using a blockchain as a decentralized entity driving the choreography execution: In a sense, a choreography can be regarded as a smart contract itself. The smart contract may store and manipulate data, own decisions and their logic and keep track of the overall execution state. Interactions within the choreography become transactions. Observability broadens considerably as each participant has access to the whole distributed ledger of transactions attached to the choreography.

The aim of this paper is to tackle the challenges mentioned and extend and refine BPMN 2.0 choreography diagrams to these changes in assumptions. Models should be expressive enough to deduce interfaces for each participant and enforce the order of all message exchanges. It should be obvious which elements of the choreography and which data is stored and executed on the blockchain. The approach should not impose technical restrictions on the participants’ internal orchestration processes and should not introduce additional trust to the interactions. On the other hand, the approach should reflect the set of capabilities common to second-generation blockchains, rather than specific implementations,

to remain as platform-independent as possible. This strategy aims at minimizing barriers to the adoption of blockchain-based choreography diagrams.

2.3 Prior Work on Choreographies

Utilizing blockchains to monitor and enforce business processes is a relatively new development in BPM. Research has been focused on exploring whether blockchains can extend or even replace traditional Business Process Management Systems (BPMS). On a modeling basis, these approaches use interpretations of BPMN 2.0 process diagrams [16], collaboration diagrams [6] or other automata [15]. In the declarative space, contract formalizations [3] as well as Dynamic Condition Response (DCR) graphs [8] are employed. To the best of our knowledge, there have been no systematic efforts to adapt BPMN 2.0 choreography diagrams to suit the needs of blockchain-based process execution.

There has been some work on a more conceptual level. Weber et al. [17] discuss the general impact of blockchain on collaborative processes, such as conflict resolution, monitoring and auditing. They use choreography diagrams for their argumentation, but do not challenge the standard's underlying assumptions regarding ownership and observability. In this respect, this paper takes the next step towards using blockchains for choreography execution.

Establishing a shared understanding of data in choreographies has been the subject of previous work by Meyer et al. [10, 11]. The authors extend BPMN 2.0 collaboration diagrams with an explicit data perspective, introducing the notion of a global data model which establishes the concrete structure of all message exchanges. Participants then use the global data model to implement or generate adapters to their local data model. Since Meyer et al. assume that there is no central management entity, messages remain transient and there is no shared choreography data, which makes this approach inapplicable to blockchains.

A different approach to collaborative process modeling for the blockchain is presented by López-Pintado et al. [7]. Their idea is that if participants share the same execution infrastructure, i.e., the blockchain, they may act as if they were located within the same organization. The choreography is thus modeled like an intra-organizational orchestration process which uses a traditional BPMS. There are no explicit choreography concepts such as message exchange. While this approach has the advantage of using modeling elements familiar to most process modelers, it is also prone to issues that sparked the development of choreography modeling in the first place, namely an insufficient level of abstraction.

Lastly, Breu et al. introduce the concept of living inter-organizational processes [1]. Taking into account developments in cloud infrastructure, such processes are enacted by multiple organizations sharing control and power. The authors identify perspectives of distribution of power, behavior, data and resources, but do not provide a concrete solution for modeling or enacting them. The extension proposed in this paper could be motivated as a concrete realization of living inter-organizational processes powered by the blockchain.

Table 1. Categories of modeling elements in blockchain-based choreography diagrams

Category	BPMN 2.0 model elements	Ownership	Status
<i>Interaction</i>	Choreography tasks with message flow	Participants	Adapted
<i>Decomposition</i>	Sub-choreographies, call choreographies	Smart contract	Adapted
<i>Data</i>	Data objects, data input, data output	Smart contract	Introduced
<i>Logic</i>	Script tasks	Smart contract	Introduced
<i>Control flow</i>	Sequence flow, gateways	Smart contract	Adapted
<i>Events</i>	Start, intermediate, and end events	Smart contract	Adapted

3 Blockchain-Based Choreographies

We propose an extension and refinement of BPMN 2.0 choreography diagrams with a specific focus on the capabilities of blockchain technology. In particular, blockchain allows us to store choreography data and evaluate logic in a shared environment that is not controlled by a single participant. This presents a major shift in the understanding of choreographies. Following the approach of the BPMN standard, we describe the intended semantics of our proposal textually.

3.1 Modeling Elements and Their Ownership

We avoided introducing entirely new modeling elements in the extensions. Instead, we opted to reuse as many familiar BPMN 2.0 concepts as possible to facilitate the adoption of the proposal. The proposal is backward-compatible: all valid choreography diagrams are still valid in our extended notation. Table 1 lists the six categories of modeling elements that we considered in this paper and whether they were changed. These include previously available elements and elements from other BPMN 2.0 diagrams, e.g., script tasks or data objects.

Ownership is a central issue in choreography diagrams. The owner of an element is responsible for all managerial tasks that need to be completed for triggering and enforcing that element in an actual execution. Whereas in standard choreography diagrams only participants could be owners, the smart contract now represents another entity capable of that. Most notably, only the interaction group contains elements that are owned by or require any input from participants, namely choreography tasks. All other elements are entirely managed by the smart contract, eliminating most of the confusion and ambiguities regarding element ownership present in the standard (see Sect. 2.1).

Figure 2 shows an example of a blockchain-based choreography diagram modeled using our approach. It extends the running example from Sect. 2.1 with additional elements covering the whole duration of the tenancy. While the syntax looks very similar to the diagram shown before, there are fundamental differences. For example, the event-based gateway e_2 is followed by two choreography tasks that can be initiated by a tenant who was not involved in the task immediately preceding the gateway. In standard choreographies, this would not be

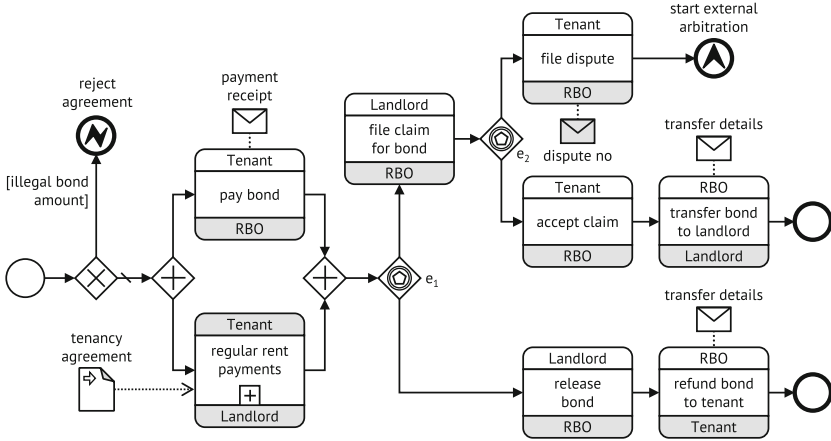


Fig. 2. Extended choreography for the interactions between a landlord, a tenant and an online rental bond agency during the entire duration of a tenancy

realizable because of observability issues. Furthermore, the ‘regular rent payments’ sub-choreography does not have an initiating participant. Instead, it is implicitly owned by the smart contract, which is not displayed in a band but manages the sub-choreography. These and other changes will be explained in the following sections.

3.2 Shared Data Model and Storage

Introducing shared data in choreographies comes with multiple challenges. For one, a shared data model is essential [11]. Participants need to have a common understanding of which data is stored and how it is mapped to properly connect their local data model and assess security and privacy concerns. This means that all data in the model, including messages, needs to be refined with concrete data structures provided by the modeler. Implementations of blockchain-based choreographies may require specific data mappings based on their underlying technology. Secondly, data needs to be suitably represented in the model itself, in our case with messages and data objects. The union of all messages and data objects with their associated structures constitutes the shared data model.

Messages are a peculiar instance of data in that they used to be an inherently transient concept. Message flows were defined as a bilateral interaction between two participants, with no provision to store or share specific message content. This situation changes in blockchain-based choreographies as message exchanges become transactions, which are automatically logged and permanently saved in the blockchain. All participants can observe these transactions, and interpret the payload according to the shared message structure.

In addition to messages, we chose to adopt the notion of data objects as specified for BPMN 2.0 process diagrams. Data objects can be regarded as variables in a concrete choreography instance, and their values are exclusively stored

within the smart contract. Figure 2 shows an example of a data object used to store the details of a tenancy agreement. The main conceptual difference between messages and data objects is that messages are owned and triggered by participants through transactions, while data objects are owned and managed by the smart contract. The blockchain guarantees a consistent view on all data.

3.3 Observability Constraints

In standard choreographies, observability as introduced in Sect. 2.1 was purely local and based on message exchanges. Each participant could only see the details of interactions they were directly involved in, implicitly capturing a very restricted view of the overall process state. In blockchain-based choreographies this assumption does not hold true anymore as each participant has complete visibility of the state. In certain scenarios, this might not be a satisfying solution. For example, some messages and data objects might contain sensitive information that should only be visible to a subset of participants.

We propose using sub-choreographies and call choreographies as a means of decomposition and to restrict observability. As they have similar semantics, we will focus on the former in the following. Every level of decomposition opens a new scope, and all the participants referenced by the sub-choreography through its bands are part of it. This allows modelers to partition a choreography into different areas with different sets of participants. Data objects can be passed to and from sub-choreographies with data inputs and data outputs, special kinds of data objects marked with a stroked and filled arrow, respectively.

Figure 2 shows an example: the tenancy agreement data input is passed to the ‘regular rent payments’ sub-choreography using a dotted data association. Because RBO is not attached to the sub-choreography, they are not able to participate in its execution. Neither can they observe its current state, nor can they inspect the content of messages or data objects contained in it. The instantiation of the sub-choreography and management of data inputs and outputs is handled by the smart contract. The implementation of observability constraints depends on the concrete blockchain chosen, and is discussed further in Sect. 4.3.

3.4 Control Flow and Embedded Logic

The handling of control flow and of gateways in particular presents a major shift from the standard. As mentioned above, there is no need to have overlapping sets of participants in tasks anymore; this also applies to pairs of tasks with a gateway between one another. In our approach, gateways are not manifested through the internal orchestrations of one or more controlling participants, but through the smart contract directly. The strict sequencing rules in the standard (ensuring that involved participants have a common knowledge of all relevant data and the choreography state) are no longer needed given the shared data storage and global observability. Instead, *data-based exclusive gateways* are transparently evaluated using formal condition expressions attached to the model. For *event-based gateways*, the smart contract ensures that only one outgoing sequence

flow is activated. It also keeps track of *parallel* execution branches and enforces correct join behavior. The execution semantics of control flow logic elements will be discussed in Sect. 3.5.

These mechanisms are based on one of the main novelties of smart contracts, the decentralized execution of shared logic in the form of smart contract code. The shared logic is used to enforce and manage all elements owned by the smart contract. Modelers may also directly exploit these capabilities in two ways: through condition expressions attached to data-based exclusive gateways as mentioned above, and script tasks. Both script tasks and conditions can be embedded in our extended choreography diagrams by refining the respective elements with concrete code which can be evaluated in the smart contract. They may access all data that is available in the scope at the time of encountering the specific element. For example, the conditions of the exclusive data-based gateway in Fig. 2 may only access the values of the ‘tenancy agreement’ data object; no other data object or message exists at this point.

3.5 Transaction-Driven Semantics

When choreography diagrams were initially designed, they were intended to be primarily descriptive, hence no specific execution semantics were defined. Instead, a mapping to equivalent collaboration diagrams was given that distributes power and trust to participants based on the implicit ownership of elements (see Sect. 2.1). That definition-by-mapping fails for blockchain-based choreographies as there is no way to model shared data or logic in collaboration diagrams.

For that reason, we introduce the notion of transaction-driven semantics for blockchain-based choreographies. Because of the nature of current blockchain technology, smart contracts are never invoked outside the context of a transaction. The smart contract can only advance the state of a choreography within a transaction from some participant or the call of another smart contract, never just by itself. In the context of enacting a choreography, there are two types of

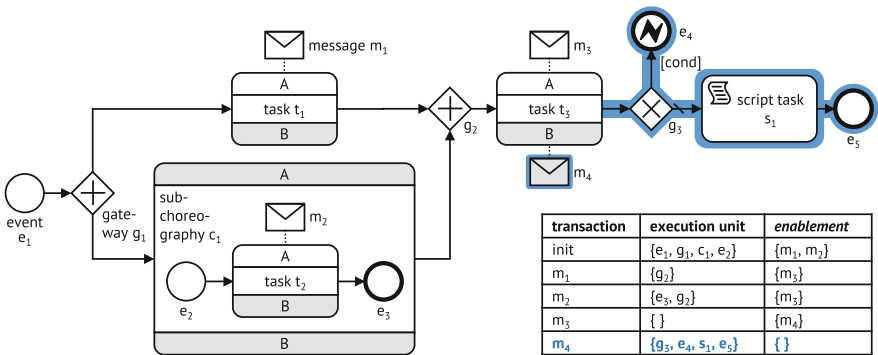


Fig. 3. Execution units of a sample blockchain-based choreography

transactions: one transaction for initiating the smart contract, and one for each message exchange. We use these transactions to bundle choreography elements into so-called execution units, which contain all elements that may possibly be executed within the respective transaction. The extent of an execution unit is delimited by elements necessitating another transaction, i.e., choreography tasks.

Figure 3 shows an example for a simple blockchain-based choreography with all execution units. For instance, the execution unit of the transaction belonging to message exchange m_4 is $\{g_3, e_4, s_1, e_5\}$ (see the shaded highlights). This means that when participant B sends message m_4 , g_3 is executed by evaluating the outgoing sequence flows' conditions. If `cond` evaluates to true, the end event e_4 is executed. Otherwise, the script task s_1 is executed, followed by e_5 . The order of evaluation and execution is based on the constraints defined by sequence flow. The example also illustrates that not all elements in the execution units necessarily have to be executed. Rather, the execution unit contains all elements that could possibly be executed as a part of that transaction subject to sequence flow. Elements can be in multiple execution units, e.g., when the order of the transactions determines which one will be able to execute a parallel join gateway (see m_1 and m_2 in the example).

Naturally, there is an ordering to messages. Message m_3 will only be accepted after both m_1 and m_2 have been sent. We express this with the notion of enablement which can be visualized as a kind of token system. Within a transaction, tokens are consumed and propagated through the model based on the execution units. A message attached to a task t is only enabled and can be sent when t is in possession of a token and, for response messages, the initiating message has been sent. A transaction may consume multiple tokens, for example if it directly follows an event-based gateway. In that case, the transaction also consumes the tokens of all the other “competing” elements following the gateway. Race conditions are automatically resolved by the blockchain’s ordering consensus: blocks and the transactions they contain are totally ordered. Figure 3 shows the potential enablement after each transaction, as per the sequence flow.

These semantics have some direct conceptual consequences. The initiating and response message of a choreography task do not form an atomic unit anymore as postulated in the BPMN 2.0 standard. Instead, both messages constitute separate transactions, with separate execution units. Further, data objects will never delay the execution of an element. Transactions are executed in an isolated environment in which blockchain data is instantly available, while external data sources cannot be used due to the closed-world nature of blockchain.

4 Implementation, Evaluation, and Discussion

We show the feasibility of our approach by a proof-of-concept implementation. Furthermore, we implement and analyze three concrete use cases and critically discuss our proposed solution’s properties.

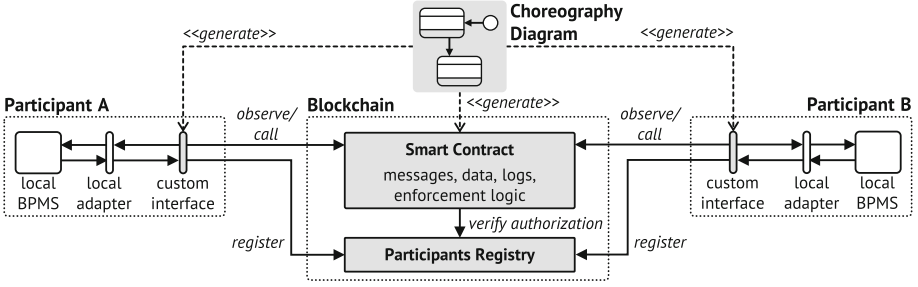


Fig. 4. Architecture for executing choreographies as smart contracts (based on [17])

4.1 Proof-Of-Concept Implementation

The proof-of-concept implements our approach for the Ethereum blockchain and smart contracts written in the Solidity language.

Execution Architecture. One of the requirements for our choreography modeling approach was to not impose any restrictions on the internal software landscape of any of the participants. The architecture of the proof-of-concept as shown in Fig. 4 reflects this goal, and is generally based on the layout proposed by Weber et al. [17]. The main artifact is the choreography model. It is used to generate several components, most prominently the smart contract encapsulating the enforcement logic and storing messages, data and logs. Additionally, a custom interface component can be deduced for each participant. This interface component hides the complexity of the blockchain and can connect to local adapters, which in turn communicate with the local BPMS. Another smart contract is used as a registry for the participants. Before executing the choreography, participants register for specific roles with their addresses. The choreography smart contract can then authorize actions which are to be taken by certain roles only.

Refinement. The choreography models need to be expressive enough to generate concrete smart contract code. BPMN 2.0 already provides model attributes to refine certain model elements to an implementation level, namely the concept of item definitions attached to messages and data objects, scripts attached to script tasks, as well as formal expressions attached to sequence flows. We use these attributes to specify conditions and data structures within the model.

For the proof-of-concept, data structures are expected to be a flat list of variables using Solidity data types. Expressions must be given as Solidity expressions evaluating to a boolean value. Script tasks must contain a valid block of Solidity code. A snippet of the refinement for the running example (see Fig. 2) is given in Table 2. For example, the tenancy agreement contains the bond amount as well as the weekly rent, and the bond may only be at most four times the weekly rent (as per official guidelines in New South Wales, Australia).

Table 2. Refinement of the running example (snippet)

Tenancy agreement	<code>uint16 bond; uint16 weeklyRent;</code>
Transfer details	<code>int32 timestamp; uint32 transferID;</code>
[illegal bond amount]	<code>agreement_bond > 4 * agreement_weeklyRent</code>

Contract Generation. We implemented a token system which generates control flow logic from the sequence flows specified in the model, similar to the Petri net-approach presented by García-Bañuelos et al. [2]. Each sequence flow can potentially possess a token that is used to enable the target element. Tokens are propagated through the model whenever a valid transaction arrives (initially deploying/starting the choreography or sending a message). The smart contract propagates the tokens as far as possible until a participant-owned element is encountered, which emulates the execution units described in Sect. 3.5.

```

1 pragma solidity ^0.4.23;
2 interface Choreography { [...]
3     function start() external;
4     function sendRequest(uint8 task, bytes message) external;
5     function sendResponse(uint8 task, bytes message) external;
6 }
```

Listing 1. Common interface (excerpt) of all choreography smart contracts

We opted for a minimal common interface shared by all choreography contracts partly shown in Listing 1. The `start` function executes the initial token propagation from the start events. `sendRequest` and `sendResponse` correspond to the request and response messages of specific tasks, respectively. Tasks are assigned an ID and all message data is encoded in a single byte array. This simple interface facilitates inter-contract communication, e.g., between sub-choreographies and their parents. It also enables the use of generic oracles sending data through requests without requiring specific code for each choreography.

4.2 Case Studies

We have modeled, executed, and analyzed three case studies that cover all the elements discussed in this paper. Apart from the rental bond example already discussed in the course of this paper we discuss two corporate case studies.

The **grain delivery** case study describes the interactions between a producer and a seller of grain, including payment using an escrow agent and analysis of the grain by a silo and laboratory. The general use-case is modeled after an example by AgriDigital² enriched with legal details obtained from Grain Trade Australia (GTA). A distinctive feature of this choreography is that script tasks are used to communicate with an external smart contract representing the grain

² <https://www.agridigital.io/>, accessed 2019-03-11.

Table 3. Number of elements in each use case

Element	Rental agreement	Grain delivery	Interline agreement
<i>Roles</i>	3	5	2
<i>Choreography tasks</i>	7	9	11
<i>Sub/call choreographies</i>	1/0	1/3	0/2
<i>Data objects</i>	1	9	5
<i>Script tasks</i>	0	6	4
<i>Gateways</i>	5	7	12
<i>Events</i>	5	13	11

title, allowing secure transfer of titles. In case of conflicts, an external arbitration entity is assigned. This shows how our approach can be integrated with other smart contracts running on the same blockchain, and can be used to implement applications that go beyond what can be achieved with choreography models alone. For example, external oracles could be introduced using this technique.

The **bilateral interline traffic agreement** case study describes a contract between two airlines when booking passengers for flights with the other airline, e.g., in case of a cancellation. Prices are tallied and a monthly settlement is performed. The use case is modeled after guidelines of the International Air Transport Association (IATA) and is an example for a long-running contract which can only be canceled via mutual agreement. The interline agreement case study especially showcases the degree of parallelism that can be achieved. Multiple branches of the choreography are active at the same time, sharing data through data objects. The case study also showcases an example of timing constraints in that the gap between two settlements needs to be at least 30 days.

The case studies and derived artifacts can be found in an online appendix.³ All examples were fully modeled, refined and successfully executed with our proof-of-concept implementation. Table 3 shows the numbers of elements in the case studies as indicators of their complexity. We can represent sophisticated processes and sequences with relatively few elements, which supports the claim

Table 4. Cost metrics of a sample trace for each case study

Case study	Factory deployment	Root deployment	Transactions		
			Count	Avg. cost	
<i>Rental agreement</i>	1,195,765 gas	2,022,705 gas	8	151,794 gas	(64% deploym.)
<i>Grain delivery</i>	4,933,814 gas	2,191,847 gas	15	278,368 gas	(78% deploym.)
<i>Interline agreement</i>	2,825,324 gas	2,093,638 gas	13	368,792 gas	(80% deploym.)

³ <https://github.com/jan-ladleif/bpm19-blockchain-based-choreographies>.

that choreography diagrams are a suitable means of modeling diverse inter-organizational processes. Also, the newly introduced elements such as script tasks and data objects are heavily used and fit well into the overall diagram structure.

Furthermore, we investigated the cost of our approach. Cost in Ethereum is measured in gas, which is determined based on the computational complexity and memory intensity of a transaction. Table 4 shows the preliminary results for one sample trace of each case study (more detailed data is available in the online appendix). There are three relevant stages. In a first stage, factory contracts for sub and call choreographies are deployed. This has to be done once and factories can be reused for multiple instances. Second, the root contract and a participant registry contract need to be deployed for each instance of the choreography. Third, based on the concrete conversation, a variable number of transactions, i.e., initializations and message exchanges, are performed. These transactions potentially involve deploying new contract instances of sub or call choreographies, which largely dominates the cost of the average transaction (see the share in Table 4). As yet, there are no provisions to evenly distribute costs between participants. Presumably, realizations of blockchain-based choreographies in production would rely on permissioned blockchains for their access controls. For those types of blockchains, transaction cost is more relevant regarding throughput and usually not directly equated to fiat currency.

4.3 Discussion

Our proof-of-concept implementation shows the feasibility of our approach in practice, while the case studies show its conceptual expressiveness. While there still is space for further extensions and a broader empirical analysis of the impact of blockchain-based choreography modeling, we present a first step into the direction laid out by Mendling et al. [9].

Our approach was partly motivated by previous work posing challenges in inter-organizational process modeling. Breu et al. postulate four challenges: flexibility, correctness, traceability and scalability [1]. Our blockchain-based solution performs well regarding correctness and traceability. The smart contract already provides runtime verification with an explicit data perspective, and known static verification techniques could be applied to our extended choreography diagrams. The blockchain's log makes choreography executions traceable and auditable.

Flexibility is not the strong suit of the proposed approach. In general, blockchains focus on immutable storage and dynamically changing or evolving smart contracts do not fit well into that paradigm. For business contracts, however, this might even be the desired behavior as a deal is usually fixed once negotiated. Explicit upgrade mechanisms may technically still be incorporated in smart contracts, though, and general architectural patterns and paradigms could be adapted to the specifics of choreographies [19]. Regarding scalability, our solution is dependent on the underlying blockchain's performance.

Not all elements are currently supported by our implementation or blockchains in general. For example, the monthly settlement in the interline agreement

case study cannot be triggered by the smart contract itself through timer events due to current limitations in the underlying technology. Until that changes, we have to rely on a workaround where either of the airlines requests payment, with the smart contract checking the time conditions when that occurs. It also became apparent that choreography models cannot capture all the intricacies of a contractual agreement, signified by the external arbitration and litigation pattern used in the rental agreement as well as the grain delivery case studies. The degree to which business contracts can be formalized with smart contracts is a topic of ongoing discussion [5]. Given that smart contracts and legal contracts are different types of “things” [14], this is unlikely to be fully resolvable.

Privacy and confidentiality pose further challenges. While the transparency of a blockchain is a requirement for some use cases, and a feature for others, many organizations would hesitate to give up confidentiality of their business contracts and associated data. While we model observability constraints using sub and call choreographies, the proof-of-concept implementation on Ethereum cannot yet realize these restrictions. Techniques like symmetric encryption or zero-knowledge proofs may be incorporated for trade-offs in cost, transparency or performance. Other blockchain platforms, like Quorum or Hyperledger Fabric, have native support for confidentiality. They provide private contracts which could be used to shield instances of sub-choreographies from certain participants as specified in the model. At a loss of trustlessness, such permissioned blockchains might be a promising solution to these issues.

Lastly, in all of the case studies, payments between participants need to be tracked. This is done through escrow agents (grain delivery), transfer receipts (rental agreement) or simple counting (interline agreement). Payments in cryptocurrencies are an inherent capability of blockchains. As blockchain-based choreographies are realized using smart contracts, they could technically hold cryptocurrency as well. As suggested by Weber et al., the choreography itself could be the escrow account, cutting out middlemen and allowing payment processing directly in the choreography [17]. However, Weber et al. handled this by custom parameter extensions. Ideally, the concept should be implemented with dedicated modeling elements and techniques to prevent locking or losing funds.

5 Conclusion

In this paper, we critically assessed capabilities and limitations of current choreography modeling approaches. We discussed how these limitations restrict the expressiveness of choreographies, and how they can be lifted using blockchain technology. To do so, we gave a backward-compatible extension and refinement of BPMN 2.0 choreography diagrams and provided operational semantics based on blockchain concepts. We implemented a proof of concept based on the Ethereum blockchain and Solidity smart contracts to show the practical feasibility of our approach. Three case studies allowed us to examine different aspects of the approach, while highlighting its versatility and remaining limitations. We see this paper as an initial step towards re-vitalizing the area of choreography modeling and adapting for new technologies such as blockchain.

In future work, we plan to address the current limitations, including model elements for payments and escrow, and how to implement observability constraints. Preliminary investigations using Quorum showed that the latter is not a straightforward task, and deserves significant additional attention.

References

1. Breu, R., et al.: Towards living inter-organizational processes. In: IEEE Conference on Business Informatics (CBI) (2013)
2. García-Bañuelos, L., Ponomarev, A., Dumas, M., Weber, I.: Optimized execution of business processes on blockchain. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNCS, vol. 10445, pp. 130–146. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_8
3. Governatori, G., Idelberger, F., Milosevic, Z., Riveret, R., Sartor, G., Xu, X.: On legal contracts, imperative and declarative smart contracts, and blockchain systems. *Artif. Intell. Law* **26**(4), 377–409 (2018). <https://doi.org/10.1007/s10506-018-9223-3>. ISSN 1572-8382
4. Haarmann, S., Batoulis, K., Nikaj, A., Weske, M.: DMN decision execution on the ethereum blockchain. In: Krogstie, J., Reijers, H.A. (eds.) CAiSE 2018. LNCS, vol. 10816, pp. 327–341. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91563-0_20
5. Kõlvart, M., Poola, M., Rull, A.: Smart contracts. In: Kerikmäe, T., Rull, A. (eds.) *The Future of Law and eTechnologies*, pp. 133–147. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-26896-5_7
6. López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I.: Caterpillar: a blockchain-based business process management system. In: *International Conference on Business Process Management (BPM), Demo Track* (2017)
7. López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I., Ponomarev, A.: Caterpillar: a business process execution engine on the Ethereum blockchain. *CoRR abs/1808.03517* (2018)
8. Madsen, M.F., Gaub, M., Høgnason, T., Kirkbro, M.E., Slaats, T., Debois, S.: Collaboration among adversaries: distributed workflow execution on a blockchain. In: *Symposium on Foundations and Applications of Blockchain* (2018)
9. Mendling, J., Weber, I., et al.: Blockchains for business process management - challenges and opportunities. *ACM Trans. Manag. Inf. Syst. (TMIS)* **9**(1), 41–416 (2018). <https://doi.org/10.1145/3183367>. ISSN 2158-656X
10. Meyer, A., Pufahl, L., Batoulis, K., Fahland, D., Weske, M.: Automating data exchange in process choreographies. *Inf. Sys.* **53**, 296–329 (2015)
11. Meyer, A., et al.: Data perspective in process choreographies: modeling and execution. Technical report BPM-13-29, BPMcenter.org (2013)
12. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
13. OMG: Business Process Model and Notation (BPMN), Version 2.0.2, December 2013. <http://www.omg.org/spec/BPMN/2.0.2/>
14. Staples, M., et al.: Risks and opportunities for systems using blockchain and smart contracts. Technical report, Data61 (CSIRO) (2017)
15. Sturm, C., Szalanczi, J., Schönig, S., Jablonski, S.: A lean architecture for blockchain based decentralized process execution. In: Daniel, F., Sheng, Q.Z., Motahari, H. (eds.) BPM 2018. LNBIP, vol. 342, pp. 361–373. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11641-5_29

16. Tran, A.B., Lu, Q., Weber, I.: Lorikeet: a model-driven engineering tool for blockchain-based business process execution and asset management. In: International Conference on Business Process Management (BPM), Demo Track (2018)
17. Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., Mendling, J.: Untrusted business process monitoring and execution using blockchain. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 329–347. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_19
18. Weske, M.: Business Process Management, 2nd edn. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-3-642-28616-2>
19. Xu, X., Weber, I., Staples, M.: Architecture for Blockchain Applications. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-03035-3>



Formal Reasoning on Natural Language Descriptions of Processes

Josep Sànchez-Ferreres¹(✉), Andrea Burattin², Josep Carmona¹,
Marco Montali³, and Lluís Padró¹

¹ Universitat Politècnica de Catalunya, Barcelona, Spain
{jsanchezf, jcarmona, padro}@cs.upc.edu

² Technical University of Denmark, Kongens Lyngby, Denmark
andbur@dtu.dk

³ Free University of Bozen-Bolzano, Bolzano, Italy
montali@inf.unibz.it

Abstract. The existence of unstructured information that describes processes represents a challenge in organizations, mainly because this data cannot be directly referred into process-aware ecosystems due to ambiguities. Still, this information is important, since it encompasses aspects of a process that are left out when formalizing it on a particular modelling notation. This paper picks up this challenge and faces the problem of ambiguities by acknowledging its existence and mitigating it. Specifically, we propose a framework to partially automate the elicitation of a formal representation of a textual process description, via text annotation techniques on top of natural language processing. The result is the ATDP language, whose syntax and semantics are described in this paper. ATDP allows to explicitly cope with several interpretations of the same textual description of a process model. Moreover, we link the ATDP language to a formal reasoning engine and show several use cases. A prototype tool enabling the complete methodology has been implemented, and several examples using the tool are provided.

1 Introduction

Organizing business processes in an efficient and effective manner is the overarching objective of *Business Process Management* (BPM). Classically, BPM has been mainly concerned with the quantitative analysis of key performance dimensions such as time, cost, quality, and flexibility [10] without considering in depth the analysis of textual data that talks about processes.

Hence, textual descriptions of processes in organizations are a vast and rather unexploited resource. Not neglecting the information that is present in natural language texts in a organization brings opportunities to complement or correct process information in conceptual models. In spite of this, only very recently *Natural Language Processing* (NLP)-based analysis has been proposed in the BPM context, as reported in [12, 14, 15, 22].

This paper is a first step towards the challenge of unleashing formal reasoning on top of textual descriptions of processes. By relying on *textual annotations*, we propose ATDP, a multi-perspective language that can be connected to a reasoner so that a formal analysis is possible. From a raw textual description, annotations can be introduced manually, or selected from those inferred by NLP analysis (e.g., from libraries like [16]), thus alleviating considerably the annotation effort. Remarkably, our perspective differs from the usual trend in conceptual modelling, i.e., ATDP specifications can contain several interpretations, so ambiguity is not forced to be ruled out when modelling, for those cases when the process is under-specified, or when several interpretations are equally valid.

We formalize ATDP, and describe its semantics using linear temporal logic (LTL), with relations defined at two different levels, thanks to the notion of *scopes*. Then we show how to cast reasoning on such a specification as a model checking instance, and provide use cases for BPM, such as model consistency, compliance checking and conformance checking. Notably, such reasoning tasks can be carried out by adopting the standard infinite-trace semantics of LTL, or by considering instead finite traces only, in line with the semantics adopted in declarative process modeling notations like Declare [17]. Finally, a tool to convert ATDP specifications into a model checking instance is reported.

The paper is organized as follows: in the next section we provide the work related to the contributions of this paper. Then Sect. 3 contains the preliminaries needed for the understanding of the paper content. Section 4 describes a methodology to use ATDP in organizations. In Sect. 5 we provide intuition, syntax and semantics behind the ATDP language. Then in Sect. 6 it is shown how reasoning on ATDP specification can be done through model checking and finally Sect. 7 concludes the paper.

2 Related Work

In order to automatically reason over a natural language process description, it is necessary to construct a formal representation of the actual process. Such generation of a formal process model starting from a natural language description of a process has been investigated from several angles in the literature. We can project these techniques into a spectrum of support possibilities to automation: from fully manual to automatic.

The first available option consists in converting a textual description into a process model by manually modeling the process. This approach, widely discussed (e.g., [9, 10]), has been thoroughly studied also from a psychological point of view, in order to understand which are the challenges involved in such process of process modeling [4, 18]. These techniques, however, do not provide any automatic support and the possibility for automatic reasoning is completely depending on the result of the manual modeling. Therefore, ambiguities in the textual description are subjectively resolved.

On the opposite side of the spectrum, there are approaches that autonomously convert a textual description of a process model into a formal

representation [13]. Such representation can be a final process model (e.g., as BPMN) [7] and, in this case, it might be possible to automatically extract information. The limit of these techniques, however, is that they need to resolve ambiguities in the textual description, resulting in “hard-coded” interpretations.

In the middle of the spectrum, we have approaches that automatically process the natural language text but they generate an intermediate artifact, useful to support the manual modeling by providing intermediate diagnostics [8, 20]. The problem of having a single interpretation for ambiguities is a bit mitigated in this case since a human modeler is still in charge of the actual modeling. However, it is important to note that the system is biasing the modeler towards a single interpretation.

The approach presented in this paper drops the assumption of resolving all ambiguities in natural language texts. Therefore, if the text is clear and no ambiguities are manifested, then the precise process can be modeled. However, if this is not the case, instead of selecting one possible ambiguity resolution, our solution copes with the presence of several interpretations for the same textual description.

3 A Recap on Linear Temporal Logics

In this paper, we use Linear Temporal Logic (LTL) [19] to define the semantics of the ATDP language. In particular, we use the standard interpretation of temporal logic formulae over *infinite traces*.

LTL formulae are built from a set \mathcal{P} of propositional symbols and are closed under the boolean connectives, the unary temporal operator \bigcirc (*next-time*) and the binary temporal operator U (*until*):

$$\varphi ::= a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 U \varphi_2 \quad \text{with } a \in \mathcal{P}$$

Intuitively, $\bigcirc\varphi$ says that φ holds at the *next* instant, $\varphi_1 U \varphi_2$ says that at some future instant φ_2 will hold and *until* that point φ_1 always holds. Common abbreviations used in LTL include the ones listed below:

- Standard boolean abbreviations, such as \top , \perp , \vee , \rightarrow .
- $\diamond\varphi = \top U \varphi$ says that φ will *eventually* hold at some future instant.
- $\square\varphi = \neg\diamond\neg\varphi$ says that from the current instant φ will *always* hold.
- $\varphi_1 W \varphi_2 = (\varphi_1 U \varphi_2 \vee \square\varphi_1)$ is interpreted as a *weak until*, and means that either φ_1 holds until φ_2 or forever.

Recall that the same syntax can also be used to construct formulae of LTL interpreted over *finite traces* [6]. Later on in the paper we show how our approach can also accommodate this interpretation. Recall however that the intended meaning of an LTL formula may radically change when moving from infinite to finite traces [5].

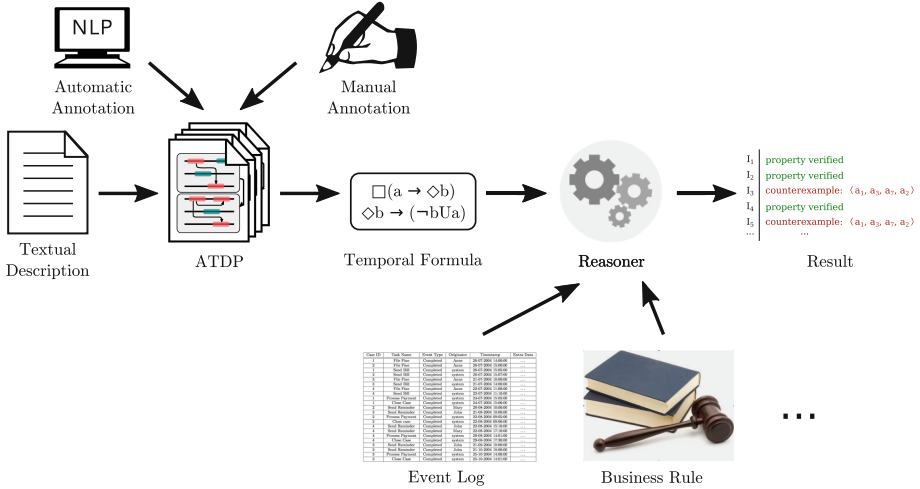


Fig. 1. Annotation framework overview

4 A Framework for Semantic Reasoning of Natural Language Descriptions of Processes

We briefly describe our envisioned framework for process modelling and management based on natural language. Figure 1 overviews the framework. Given a textual description of a process, automatic or manual annotation (or a combination of both) is used to obtain an Annotated Textual Description of a Process (ATDP), which contains all the interpretations of the original text. This specification can then be automatically transformed into temporal formula that encompasses the semantics of the process. The temporal formula can then be queried with the help of a reasoner (e.g., a model checker). Typical use cases may require the encoding of additional inputs, e.g., traces of an event log, compliance rules, among others. The result of the reasoner is the satisfaction or rebuttal (with the corresponding counterexample) of the query. Notice that query results may not hold in all possible interpretations of the text.

5 Processes as Annotated Textual Descriptions

We now propose ATDP, a language for annotated textual descriptions of processes starting with a gentle introduction relying on a real-world example. Specifically, we use the textual description of the examination process of a Hospital extracted from [21]. Figure 2 shows the full text, while Fig. 3 contains a fragment of the visualization for an ATDP specification of the description.

One of the key features of the ATDP approach is the ability to capture *ambiguity*. In our example, we can see this at the topmost level: the text is associated

to three different interpretations I_1 , I_2 and I_3 , providing three different process-oriented semantic views on the text. Each interpretation is a completely unambiguous specification of the process, which fixes a specific way for understanding ambiguous/unclear parts. Such parts could be understood differently in another interpretation. A specification in ATDP then consists of the union of all the valid interpretations of the process, which may partially overlap but also contradict each other.

Each interpretation consists of a hierarchy of *scopes*, providing a recursive mechanism to isolate parts of text that correspond to “phases” in the process. Each scope is thus a conceptual block inside the process, which is in turn decomposed as a set of lower-level scopes. Each scope dictates how its inner scopes are linked via control-flow relations expressing the allowed orderings of execution of such inner scopes. In our example, I_1 contains two scopes. A sequential relation indicates that the second scope is always executed when the first is completed, thus reconstructing the classical flow relation of conventional process modeling notation. All in all, the scope hierarchy resembles that of a process tree, following the variant used in [1].

Inside leaf scopes, *text fragments* are highlighted. There are different types of fragments, distinguished by color in our visual front-end. Some fragments (shown in red) describe the atomic units of behavior in the text, that is, activities and events, while others (shown in blue) provide additional perspectives beyond control flow. For example, `outpatient physician` is labelled as a *role* at the beginning of the text, while `informs` is labelled as an *activity*. Depending on their types, fragments can be linked by means of *fragment relations*. Among such relations, we find:

- Fragment relations that capture background knowledge induced from the text, such as for example the fact that the `outpatient physician` is the role responsible for performing (i.e., is the **Agent** of) the `informs` activity.
- Temporal constraints linking activities so as to declaratively capture the acceptable courses of execution in the resulting process, such as for example the fact that `informs` and `signs an informed consent` are in **succession** (i.e., `informs` is executed if and only if `signs an informed consent` is executed afterwards).

As for temporal relations, we consider a relevant subset of the well-known patterns supported by the Declare declarative process modeling language [17]. In this light, ATDP can be seen as a multi-perspective variant of a process tree where the control-flow of leaf scopes is specified using declarative constraints over the activities and events contained therein. Depending on the adopted constraints, this allows the modeler to cope with a variety of texts, ranging from loosely specified to more procedural ones. At one extreme, the modeler can choose to nest scopes in a fine-grained way, so that each leaf scope just contains a single activity fragment; with this approach, a pure process tree is obtained. At the other extreme, the modeler can choose to introduce a single scope containing all activity fragments of the text, and then add temporal constraints relating

The process starts when the female patient is examined by an outpatient physician, who decides whether she is healthy or needs to undertake an additional examination. In the former case, the physician fills out the examination form and the patient can leave. In the latter case, an examination and follow-up treatment order is placed by the physician, who additionally fills out a request form. Furthermore, the outpatient physician informs the patient about potential risks. If the patient signs an informed consent and agrees to continue with the procedure, a delegate of the physician arranges an appointment of the patient with one of the wards. Before the appointment, the required examination and sampling is prepared by a nurse of the ward based on the information provided by the outpatient section. Then, a ward physician takes the sample requested. He further sends it to the lab indicated in the request form and conducts the follow-up treatment of the patient. After receiving the sample, a physician of the lab validates its state and decides whether the sample can be used for analysis or whether it is contaminated and a new sample is required. After the analysis is performed by a medical technical assistant of the lab, a lab physician validates the results. Finally, a physician from the outpatient department makes the diagnosis and prescribes the therapy for the patient.

Fig. 2. Textual description of a patient examination process.

arbitrary activity fragments from all the text; with this approach, a pure declarative process model is obtained.

5.1 ATDP Models

ATDP models are defined starting from an input text, which is separated into *typed text fragments*. We now go step by step through the different components of our approach, finally combining them into a coherent model. We then move into the semantics of the model, focusing on its temporal/dynamic parts and formalizing them using LTL.

Fragment Types. Fragments have no formal semantics associated by themselves. They are used as basic building blocks for defining ATDP models. We distinguish fragments through the following types.

Activity. This fragment type is used to represent the atomic units of work within the business process described by the text. Usually, these fragments are associated with verbs. An example activity fragment would be `validates` (from `validates the sample state`). Activity fragments may also be used to annotate other occurrences in the process that are relevant from the point of view of the control flow, but are exogenous to the organization responsible for the execution of the process. For instance, `(the sample) is contaminated` is also an activity fragment in our running example.

Role. The role fragment type is used to represent types of autonomous actors involved in the process, and consequently responsible for the execution of activities contained therein. An example is `outpatient physician`.

Business Object. This type is used to mark all the relevant elements of the process that do not take an active part in it, but that are used/manipulated by activities contained in the process. An example is the (medical) **sample** obtained and analyzed by physicians within the patient examination process.

When the distinction is not relevant, we may refer to fragments as the entities they represent (e.g. *activity* instead of *activity fragment*).

Given a set F of text fragments, we assume that the set is partitioned into three subsets that reflect the types defined above. We also use the following dot notation to refer to such subsets: (i) F .activities for activities; (ii) F .roles for roles; (iii) F .objects for business objects.

Fragment Relations. Text fragments can be related to each other by means of different non-temporal relations, used to express multi-perspective properties of the process emerging from the text. We consider the following relations over a set F of fragments.

Agent. An *agent relation* over F is a partial function

$$agent_F : F.activities \rightarrow F.roles$$

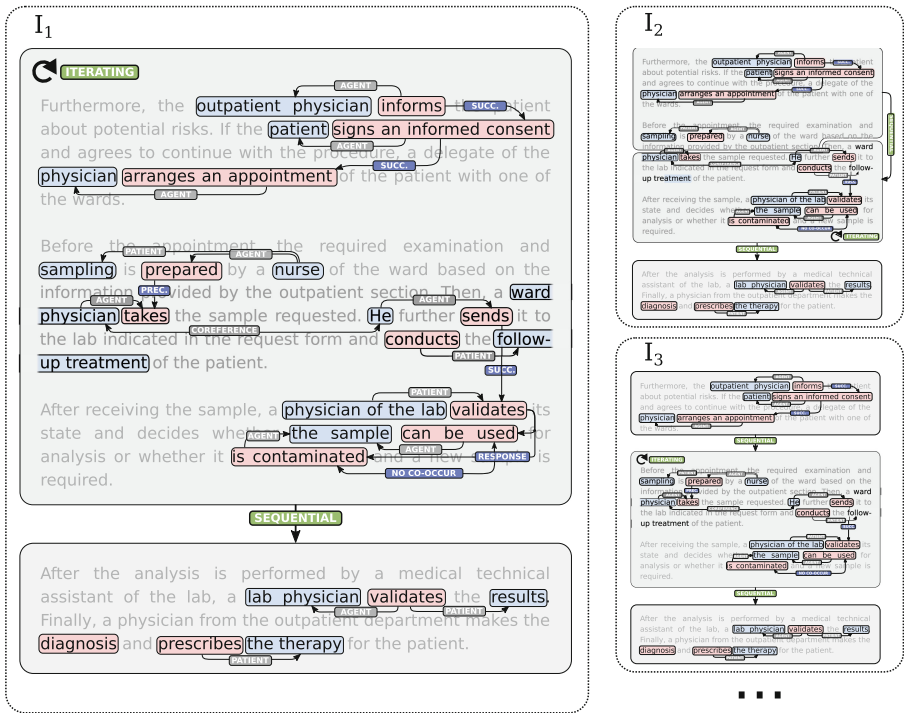


Fig. 3. Example annotation of a textual process description with multiple ambiguous interpretations. Some relations are omitted for brevity. (Color figure online)

indicating the role responsible for the execution of an activity. For instance, in our running example we have $agent(\text{informs}) = \text{physician}$, witnessing that informing someone is under the responsibility of a physician.

Patient. A *patient relation* over F is a partial function

$$patient_F : F.activities \rightarrow F.roles \cup F.objects$$

indicating the role or business object constituting the main recipient of an activity. For instance, in our running example we have $patient(\text{prepare}) = \text{sample}$, witnessing that the `prepare` activity operates over a `sample`.

Coreference. A *coreference relation* over F is a (symmetric) relation

$$coref_F \subseteq F.roles \times F.roles \cup F.objects \times F.objects$$

that connects pairs of roles and pairs of business objects when they represent different ways to refer to the same entity. It consequently induces a coreference graph where each connected component denotes a distinct process entity. In our running example, all text fragments pointing to the `patient` role corefer to the same entity, whereas there are three different physicians involved in the text: the `outpatient physician`, the `ward physician` and the `physician of the lab`. These form disconnected coreference subgraphs.

Text Scopes. To map the text into a process structure, we suitably adjust the notion of process tree used in [1]. In our approach, the blocks of the process tree are actually *text scopes*, where each scope is either a *leaf scope*, or a branching scope containing a one or an ordered pair¹ of (leaf or branching) sub-scopes.

Each activity is associated to one and only one leaf scope, whereas each leaf scope contains one or more activities, so as to non-ambiguously link activities to their corresponding process phases.

Branching scopes, instead, are associated to a corresponding control-flow operator, which dictates how the sub-scopes are composed when executing the process. At execution time, each scope is enacted possibly multiple times, each time taking a certain amount of time (marked by a punctual scope start, and a later completion). We consider in particular the following scope relation types:

Sequential (\rightarrow) A sequential branching scope s with children $\langle s_1, s_2 \rangle$ indicates that each execution of s amounts to the sequential execution of its sub-scopes, in the order they appear in the tuple. Specifically: *(i)* when s is started then s_1 starts; *(ii)* whenever s_1 completes, s_2 starts; *(iii)* the completion of s_2 induces the completion of s .

Conflicting (\times) A conflicting branching scope s with children $\langle s_1, s_2 \rangle$ indicates that each execution of s amounts to the execution of one and only one of its children, thus capturing a choice. Specifically: *(i)* when s is started, then one among s_1 and s_2 starts; *(ii)* the completion of the selected sub-scope induces the completion of s .

¹ We keep a pair for simplicity of presentation, but all definitions carry over to n -ary tuples of sub-blocks.

Inclusive (\vee) An inclusive branching scope s with children $\langle s_1, s_2 \rangle$ indicates that each execution of s amounts to the execution of at least one of s_1 and s_2 , but possibly both.

Interleaving (\wedge) An interleaving branching scope s with children $\langle s_1, s_2 \rangle$ indicates that each execution of s amounts to the interleaved, parallel execution of its sub-scopes, without ordering constraints among them. Specifically: (i) when s is started, then s_1 and s_2 start; (ii) the latest, consequent completion of s_1 and s_2 induces the completion of s .

Iterating (\odot) An iterating branching scope s with child s_1 indicates that each execution of s amounts to the iterative execution of s_1 , with one or more iterations. Specifically: (i) when s is started, then s_1 starts; (ii) upon the consequent completion of s_1 , then there is a non-deterministic choice on whether s completes, or s_1 is started again.

All in all, a *scope tree* T_F over the set F of fragments is a binary tree whose leaf nodes S_l are called *leaf scopes* and whose intermediate/root nodes S_b are called *branch nodes*, and which comes with two functions:

- a total *scope assignment* function $parent : F.activities \rightarrow S_l$ mapping each activity in F to a corresponding leaf scope, such that each leaf scope in S_l has at least one activity associated to it;
- a total *branching type* function $btype : S_b \rightarrow \{\rightarrow, \times, \vee, \wedge, \odot\}$ mapping each branching scope in S_b to its control-flow operator.

Temporal Constraints Among Activities. Activities belonging to the same leaf scope can be linked to each other by means of temporal relations, inspired by the Declare notation [17]. These can be used to declaratively specify constraints on the execution of different activities within the same leaf scope. Due to the interaction between scopes and such constraints, we follow here the approach in [11], where, differently from [17], constraints are in fact *scoped*.²

We consider in particular the following constraints:

Scoped Precedence Given activities a_1, \dots, a_n, b , $Precedence(\{a_1, \dots, a_n\}, b)$ indicates that b can be executed only if, within the same instance of its parent scope, at least one among a_1, \dots, a_n have been executed *before*.

Scoped Response Given activities a, b_1, \dots, b_n , $Response(a, \{b_1, \dots, b_n\})$ indicates that whenever a is executed within an instance of its parent scope, then at least one among b_1, \dots, b_n has to be executed *afterwards*, within the same scope instance.

Scoped Non-Co-Occurrence Given activities a, b , $NonCoOccurrence(a, b)$ indicates that whenever a is executed within an instance of its parent scope, then b *cannot* be executed within the same scope instance (and vice-versa).

Scoped Alternate Response Given activities a, b_1, \dots, b_n , $AlternateResponse(a, \{b_1, \dots, b_n\})$ indicates that whenever a is executed within an instance of its parent scope, then a *cannot be executed again* until, within the same scope, at least one among b_1, \dots, b_n is *eventually* executed.

² It is interesting to notice that Declare itself was defined by relying on the patterns originally introduced in [11].

Terminating Given activity a , $\text{Terminating}(a)$ indicates that the execution of a within an instance of its parent scope terminates that instance.

Mandatory Given activity a , $\text{Mandatory}(a)$ indicates that the execution of a must occur at least once for each execution of its scope.

Interpretations and Models. We are now ready to combine the components defined before into an integrated notion of text interpretation. An *ATDP interpretation* I_X over text X is a tuple $\langle F, \text{agent}_F, \text{patient}_F, \text{coref}_F, T_F, C_F, \rangle$, where: (i) F is a set of *text fragments* over X ; (ii) agent_F is an *agent function* over F ; (iii) patient_F is a *patient function* over F ; (iv) coref_F is a *coreference relation* over F ; (v) T_F is a *scope tree* over the activities in F ; (vi) C_F is a set of *temporal constraints* over the activities in F , such that if two activities are related by a constraint in, then they have to belong to the same leaf scope according to T_F .

An *ATDP model* M_X over text X is then simply a finite set of ATDP interpretations over X .

5.2 ATDP Semantics

We now describe the execution semantics of ATDP interpretations, in particular formalizing the three key notions of scopes, scope types (depending on their corresponding control-flow operators), and temporal constraints over activities. This is done by using LTL, consequently declaratively characterizing those execution traces that conform to what is prescribed by an ATDP interpretation. We consider execution traces as finite sequences of atomic activity executions over interleaving semantics.

Scope Semantics. To define the notion of scope execution, for each scope s , we introduce a pair of artificial activities st_s and en_s which do not belong to F .activities. The execution of s starts with the execution of st_s , and ends with the execution of en_s . The next three axioms define the semantics of scopes:

A1. An activity a inside a scope s can only be executed between st_s and en_s :

$$\neg a W st_s \wedge \Box(en_s \rightarrow \neg a W st_s)$$

A2. A scope s can only be started and ended inside of its parent s' :

$$\neg(st_s \vee en_s) W st_{s'} \wedge \Box(en_{s'} \rightarrow \neg(st_s \vee en_s) W st_{s'})$$

A3. Executions of the same scope cannot overlap in time. That is, for each execution of a scope s 's start there is a unique corresponding end:

$$\begin{aligned} & \Diamond en_s \rightarrow (\neg en_s U st_s) \wedge \Box(st_s \rightarrow \Diamond en_s) \wedge \\ & \Box(st_s \rightarrow \bigcirc(\Diamond st_s \rightarrow (\neg st_s U en_s))) \wedge \\ & \Box(en_s \rightarrow \bigcirc(\Diamond en_s \rightarrow (\neg en_s U st_s))) \end{aligned}$$

Temporal Constraint Semantics. In this section, we define the semantics of temporal constraints between activities. Note that, in all definitions we will use the subindex s to refer to the scope of the constraint.

$$\text{Precedence}_s(\{a_1, \dots, a_K\}, b) := \bigvee_{i=1}^N \Box(st_s \rightarrow (\neg b U (a_i \vee en_s)))$$

$$\text{Response}_s(a, \{b_1, \dots, b_N\}) := \bigvee_{i=1}^N \Box(st_s \rightarrow (a \rightarrow (\neg en_s U b_i)) U en_s)$$

$$\begin{aligned} \text{NonCoOccurrence}_p(a, b) &:= \Box(st_s \rightarrow (a \rightarrow (\neg b U en_s)) U en_s) \wedge \\ &\quad \Box(st_s \rightarrow (b \rightarrow (\neg a U en_s)) U en_s) \end{aligned}$$

$$\begin{aligned} \text{AlternateResponse}_p(a, b) &:= \text{Response}_p(a, b) \wedge \\ &\quad \Box(st_s \rightarrow (a \rightarrow \bigcirc(\neg a U (b \vee en_s))) U en_s) \end{aligned}$$

$$\text{Terminating}_p(a) := \Box(a \rightarrow \bigcirc en_s)$$

$$\text{Mandatory}_p(a) := \Box(st_s \rightarrow (\neg en_s U a))$$

Scope Relation Semantics. In all our definitions, let $\langle s_1, s_2 \rangle$ denote the children of a branching scope s , associated to the control-flow operator being defined. Note that by $\text{Sequence}(a, b)$ we refer to the formula $\text{Precedence}(\{a\}, b) \wedge \text{Response}(a, \{b\})$.

Sequential (\rightarrow) : $\text{Sequence}_s(en_{s_1}, st_{s_2}) \wedge \text{Mandatory}_s(st_{s_1}) \wedge \text{Mandatory}_s(st_{s_2})$

Conflicting (\times) : $\text{Mandatory}_s(st_{s_1}) \oplus \text{Mandatory}_s(st_{s_2})$

Inclusive (\vee) : $\text{Mandatory}_s(st_{s_1}) \vee \text{Mandatory}_s(st_{s_2})$

Interleaving (\wedge) : $\text{Mandatory}_s(st_{s_1}) \wedge \text{Mandatory}_s(st_{s_2})$

Iterating (\odot) : This relation is defined by negation, with any non-iterating scope s , child of s' , fulfilling the property:

$$(st_{s'} \rightarrow (\neg en_{s'} U st_s \wedge (st_s \rightarrow \bigcirc(\neg st_s U en_{s'}))) U en_{s'})$$

Additionally, iterating scopes may be affected by the presence of terminating activities, as defined by the following property: A terminating activity a_t inside an iterating scope s , child of s' , stops the iteration. That is, its execution cannot be repeated anymore inside its parent:

A4. $\Box(st_{s'} \rightarrow ((a_t \rightarrow (\neg st_s U en_{s'})) U en_{s'}))$

6 Reasoning on ATDP Specifications

A specification in ATDP is the starting point for reasoning over the described process. This section shows how to encode the reasoning as a model checking instance, so that a formal analysis can be applied on the set of interpretations of the model. Furthermore, we present three use cases in the scope of business process management: *checking model consistency*, *compliance checking* and *conformance checking*.

6.1 Casting Reasoning as Model Checking

Reasoning on ATDP specifications can be encoded as an instance of model checking, which allows performing arbitrary queries on the model. The overall system can be defined by the following formula

$$(A \wedge \mathcal{C}_F \wedge \mathcal{C}_{T_F}) \implies Q \quad (1)$$

where A is the conjunction of all LTL formulas defined by the axioms, \mathcal{C}_F is the conjunction of the activity temporal constraints, \mathcal{C}_{T_F} is the conjunction of all LTL formulas defined by the semantics of the process tree, and Q is an arbitrary query expressed in LTL (cf. Sect. 5.2).

In this paper, we present an encoding of the ATDP's semantics into NuSMV, a well-known software for model-checking [3]. First, the notion of process execution is defined using an *activity* variable, with a domain of all the activities in the ATDP. At any given step, the system may choose a single value for this variable, meaning that this activity has been executed. This ensures that simultaneous execution of activities will not happen.

The system definition for an ATDP is then split into two parts: a transition system and an LTL property specification. The transition system is a graph defining the next possible values for the *activity* variable given its current value. In our proposed encoding the transition system is specified as a complete graph, since all the behavioural constraints are specified in A , \mathcal{C}_F and \mathcal{C}_{T_F} as parts of the property specification, as seen in Eq. (1). This property specification is directly encoded as a single LTL formula.

We can adapt NuSMV, which performs model checking on infinite traces, to check properties on finite traces when necessary. In order to do that, we add a special activity value **STOP**. In the transition system, an edge is added from any possible activity to **STOP**. Additionally, the constraint $\Diamond \text{STOP}$ is added to the antecedent of the LTL property specification. This enforces that all traces accepted by the model end in an infinite loop repeating the (only) execution of the **STOP** activity, which is equivalent to terminating execution.

Non-temporal information can be introduced in the queries without increasing the problem complexity, since the information is statically defined. For example, when the text mentions that several activities are performed by a certain role, this information remains invariant during the whole model-checking phase. Thus, queries concerning roles can be translated directly into queries about the

set of activities performed by that role. A possible encoding of this into a model checker consists of adding additional variables during the system definition.

When dealing with multiple interpretations, the above framework is extended with two types of queries:

Existential: Is the proposition true in any interpretation of the process?

$$\exists I \in \text{ATDP} : (A_I \wedge \mathcal{C}_{F_I} \wedge \mathcal{C}_{T_{F_I}}) \implies Q$$

Complete: Is the proposition true in all interpretations of the process?

$$\forall I \in \text{ATDP} : (A_I \wedge \mathcal{C}_{F_I} \wedge \mathcal{C}_{T_{F_I}}) \implies Q$$

Existential and complete queries can be used to reason in uncertain or incomplete specifications of processes.

An application of complete queries would be finding invariant properties of the process. That is, a property that holds in all possible process interpretations. Existential queries, in turn, fulfill a similar role when the proposition being checked is an undesired property of the process. By proving invariant properties, it is possible to extract information from processes even if these are not completely specified or in case of contradictions. A negative result for this type of query would also contain the non-compliant interpretations of the process, which can help the process owner in gaining some insights about which are the assumptions needed to comply with some business rule.

Tool Support. The encoding technique described in Sect. 6.1 has been implemented in a prototype tool, `ATDP2NuSMV`. The tool can be used to convert an ATDP specification into a `NuSMV` instance.

`ATDP2NuSMV` is distributed as a standalone tool, that can be used in any system with a modern Java installation, and without further dependencies. A compiled version as well as the source code can be found in the following repository: <https://github.com/setzer22/atdp2nusmv>.

In the next subsection, we present use cases that have been tested with `ATDP2NuSMV` and `NuSMV`. The ATDP specifications as well as the exact query encodings can be found in the repository. The use case examples are based on a full version of the specification presented in Fig. 3.

Use Case 1: Model Consistency. An ATDP specification can be checked for consistency using proof by contradiction. Specifically, if we set $Q = \perp$, the reasoner will try to prove that $A \wedge \mathcal{C}_F \wedge \mathcal{C}_{T_F} \rightarrow \perp$, that is, whether a false conclusion can be derived from the axioms and constraints describing our model. Since this implication only holds in the case $\perp \rightarrow \perp$, if the proof succeeds we will have proven that $A \wedge \mathcal{C}_F \wedge \mathcal{C}_{T_F} \equiv \perp$, i.e. that our model is not consistent. On the contrary, if the proof fails we can be sure that our model does not contain any contradiction.

To illustrate this use case, we use interpretations `hosp-1` and `hosp-1-bad`, available in our repository. The first interpretation consists of a complete version of the specification in Fig. 3, where `F.activities` includes $a_1 = \text{takes (the sample)}$ and $a_2 = \text{validates (sample state)}$, and constraints in \mathcal{C}_F include: `Mandatory(a_1)`, `Precedence($\{a_1\}, a_2$)` and `Response($a_1, \{a_2\}$)`.

NuSMV falsifies the query in interpretation `hosp-1` with a counter-example. When the model is consistent, the property is false, and the resulting counter example can be any valid trace in the model.

The second specification, `hosp-1-bad` adds $\text{Precedence}(\{a_2\}, a_1)$ to the set of relations R . This relation contradicts the previously existing $\text{Precedence}(\{a_1\}, a_2)$, thus resulting in an inconsistent model. Consequently, NuSMV cannot find a counter-example for the query in interpretation `hosp-1-bad`. This result can be interpreted as the model being impossible to fulfill by any possible trace, and thus inconsistent.

Use Case 2: Compliance Checking. Business rules, as those arising from regulations or SLAs, impose further restrictions that any process model may need to satisfy. On this regard, compliance checking methods assess the adherence of a process specification to a particular set of predefined rules.

The presented reasoning framework can be used to perform compliance checking on ATDP specifications. An example rule for our running example might be: “An invalid sample can never be used for diagnosis”. The relevant activities for this property are annotated in the text: $a_3 = (\text{the sample})$ can be used, $a_4 = (\text{the sample})$ is contaminated, $a_5 = \text{makes the diagnosis}$, and the property can be written in LTL as: $Q = \Box(a_4 \rightarrow (\neg a_5 U a_3))$.

In the examples from our repository, interpretations `hosp-2-i`, with $i=\{1,2,3\}$, correspond to the three interpretations of the process shown in Fig. 3. Particularly, the ambiguity between the three interpretations is the scope of the repetition when the taken sample is contaminated. The three returning points correspond to: `sign an informed consent`, `sampling is prepared` and `take the sample`. NuSMV finds the property true for all three interpretations, meaning that we can prove the property $\Box(a_4 \rightarrow (\neg a_5 U a_3))$ without resolving the main ambiguity in the text.

Use Case 3: Conformance Checking. Conformance checking techniques put process specifications next to event data, to detect and visualize deviations between modeled and observed behavior [2]. On its core, conformance checking relies on the ability to find out whereas an observed trace can be reproduced by a process model.

A decisional version of conformance checking can be performed, by encoding traces inside Q as an LTL formulation. Given a trace $t = \langle a_1, a_2, \dots, a_N \rangle$, we can test conformance against an ATDP interpretation with the following query³:

$$Q = \neg(a_1 \wedge O(a_2 \wedge O(\dots \wedge O(a_N \wedge \text{OSTOP}))))$$

This query encodes the proposition “Trace t is not possible in this model”. This proposition will be false whenever the trace is accepted by the model.

³ The proposed query does not account for the start and end activities of scopes, which are not present in the original trace. A slightly more complex version can be crafted that accounts for any invisible activity to be present between the visible activities of the trace. We do not show it here for the sake of simplicity.

Other variants of this formulation allow for testing trace patterns: partial traces or projections of a trace to a set of activities. In this case, the counter-example produced will be a complete trace which fits the model and the queried pattern.

As an example of this use-case, we provide the example `ATDP` interpretation `hosp-3` in our repository. We project the set of relevant activities to the set `a6 = informs (the patient)`, `a7 = signs (informed consent)` `a8 = arranges (an appointment)`. Two trace patterns are tested, the first: $t_1 = \langle \dots a_6, a_7, a_8, \dots \rangle$ and $t_2 = \langle \dots a_7, a_6, a_8, \dots \rangle$. NuSMV finds the trace pattern t_1 fitting the model, and produces a full execution trace containing it. On the other hand, t_2 does not fit the model, which is successfully proven by NuSMV.

7 Conclusions and Future Work

This paper proposes `ATDP`, a novel multi-perspective language for the representation of processes based on textual annotation. On the control-flow dimension, `ATDP` is a mixture of imperative constructs at general level via scopes, and declarative constructs inside each scope. In a way, the language generalizes process trees, allowing declarative relations instead of atomic activities in the leaf nodes. The paper also shows how to translate `ATDP` specifications into temporal formulas that are amenable for reasoning. Three use cases in the context of BPM are shown, illustrating the potential of the ideas in this paper.

Several avenues for future work are under consideration. First, to explore alternatives or refinements of the encoding in Eq. (1) to make it more suitable in a model-checking context. Second, to validate the proposed language against more examples and use cases, specifically by testing how the `ATDP` primitives accommodate to different document styles. Finally, studying the connection between `ATDP` and other process model notations may serve as a bridge between textual descriptions and their operationalization within an organization.

Acknowledgments. This work has been supported by MINECO and FEDER funds under grant TIN2017-86727-C2-1-R and by Innovation Fund Denmark project [EcoKnow.org](https://doi.org/10.1007/978-3-319-99414-7) (7050-00034A).

References

1. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: A genetic algorithm for discovering process trees. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2012)
2. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance Checking - Relating Processes and Models. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-99414-7>
3. Cimatti, A., Clarke, E.M., Giunchiglia, F., Roveri, M.: NUSMV: a new symbolic model checker. *STTT* **2**(4), 410–425 (2000)
4. Claes, J., Vanderfeesten, I., Pinggera, J., Reijers, H.A., Weber, B., Poels, G.: A visual analysis of the process of process modeling. *Inf. Syst. e-Bus. Manag.* **13**(1), 147–190 (2015)

5. De Giacomo, G., De Masellis, R., Montali, M.: Reasoning on LTL on finite traces: insensitivity to infiniteness. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 1027–1033. AAAI Press (2014)
6. De Giacomo, G., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), pp. 854–860. IJCAI/AAAI (2013)
7. Delicado, L., Sanchez-Ferreres, J., Carmona, J., Padró, L.: NLP4BPM - natural language processing tools for business process management. In: BPM Demo Track (2017)
8. Delicado, L., Sanchez-Ferreres, J., Carmona, J., Padró, L.: The model judge - a tool for supporting novices in learning process modeling. In: BPM 2018 Demonstration Track (2018)
9. Dijkman, R., Vanderfeesten, I., Reijers, H.A.: Business process architectures: overview, comparison and framework. *Enterp. Inf. Syst.* **10**(2), 129–158 (2016)
10. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.: *Fundamentals of Business Process Management*, 2nd edn. Springer, Heidelberg (2018). <https://doi.org/10.1007/978-3-662-56509-4>
11. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Patterns in property specifications for finite-state verification. In: Proceedings of the 1999 International Conference on Software Engineering (ICSE), pp. 411–420. ACM (1999)
12. Leopold, H.: *Natural Language in Business Process Models*. Springer, Cham (2013). <https://doi.org/10.1007/978-3-319-04175-9>. Ph.D. thesis
13. 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
14. Mendling, J., Baesens, B., Bernstein, A., Fellmann, M.: Challenges of smart business process management: an introduction to the special issue. *Decis. Support Syst.* **100**, 1–5 (2017)
15. Mendling, J., Leopold, H., Pittke, F.: 25 challenges of semantic process modeling. *Int. J. Inf. Syst. Softw. Eng. Big Co.* **1**(1), 78–94 (2015)
16. Padró, L., Stanilovsky, E.: Freeing 3.0: towards wider multilinguality. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC), pp. 2473–2479 (2012)
17. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: Proceedings of the Eleventh IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), pp. 287–298. IEEE Computer Society (2007)
18. Pinggera, J.: *The process of process modeling*. Ph.D. thesis, University of Innsbruck, Department of Computer Science (2014)
19. Pnueli, A.: The temporal logic of programs, pp. 46–57. IEEE (1977)
20. Sánchez-Ferreres, J., Carmona, J., Padró, L.: Aligning textual and graphical descriptions of processes through ILP techniques. In: Dubois, E., Pohl, K. (eds.) *CAiSE 2017*. LNCS, vol. 10253, pp. 413–427. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_26
21. Semmelrodt, F.: *Modellierung klinischer prozesse und compliance regeln mittels BPMN 2.0 und eCRG*. Master’s thesis, University of Ulm (2013)
22. van der Aa, H., Carmona, J., Leopold, H., Mendling, J., Padró, L.: Challenges and opportunities of applying natural language processing in business process management. In: Proceedings of the 27th International Conference on Computational Linguistics (COLING), pp. 2791–2801 (2018)



Goal-oriented Process Enhancement and Discovery

Mahdi Ghasemi^(✉)  and Daniel Amyot 

EECS, University of Ottawa, Ottawa, Canada
{mghasemi, damyot}@uottawa.ca

Abstract. Process mining practices are mainly activity-oriented and they seldom consider the (often conflicting) goals of stakeholders. Involving goal-related factors, as often done in requirements engineering, can improve the rationality and interpretability of mined models and lead to better opportunities to satisfy stakeholders. This paper proposes a new Goal-oriented Process Enhancement and Discovery (GoPED) method to align discovered models with stakeholders' goals. GoPED first adds goal-related attributes to traditional event characteristics (case identifier, activities, and timestamps), selects a subset of cases with respect to a goal-related criterion, and finally discovers a process model from that subset. We define three types of criteria that suggest desired satisfaction levels from a (i) case perspective, (ii) goal perspective, and (iii) organization perspective. For each criterion, an algorithm is proposed to enable selecting the best subset of cases where the criterion holds. The resulting process models are expected to reproduce the desired level of satisfaction. A synthetic event log is used to illustrate the proposed algorithms and to discuss their results.

Keywords: Business process management · Process mining · Goal modeling · Requirements engineering · Event logs · Performance indicators

1 Introduction

The process mining community has developed various algorithms and tools to enable the analysis of event logs to discover process models and improve their underlying processes. Process mining activities involve: (1) *Discovery*, where a model is being created from event logs; (2) *Conformance checking*, where differences between the model and reality are detected; and (3) *Enhancement*, where an existing process model is improved or extended using some additional desired data from different aspects [16].

Event logs, resulting from the execution of processes, are the main input of process discovery activities. However, process mining approaches usually do not consider specific goals that individual cases pursue and satisfaction levels that traces yielded for different stakeholders' goals [7]. This situation not only threatens the *rationality* behind the discovered models, but also often results in unstructured “*spaghetti-like*” process models. Although such models reflect reality, they cover many exceptions and many traces misaligned with goals [12]. Process mining practitioners have to deal with such problems especially in flexible environments that allow multiple alternatives within process execution.

There are currently some strategies that deal with unstructured discovered processes often taking into account the *frequency* of activities and transitions. For example, keeping the activities that occur at least for 20% of cases is a way to simplify the model. In contrast to strategies that change logs, abstraction techniques such as fuzzy mining [16] are applied to the resulting process graphs. Also, current *declarative* approaches, e.g., based on linear temporal logic, exist to enforce some constraints (e.g., on sequencing) and discover complying models at the *activity level* [12].

Goal modeling, a requirements engineering approach that enables the description of the interrelated (and often conflicting) goals of systems and stakeholders, can be leveraged for addressing the aforementioned problems. Goal modeling is used to support heuristic, qualitative, or formal reasoning about goals, and ultimately trade-off analysis, what-if analysis, and decision making. In contrast to process mining where “*how*”, “*what*”, “*where*”, “*who*”, and especially “*when*” questions are answered, goal modeling focuses mainly on complementary “*why*” questions [2].

We hypothesize that a goal-oriented approach combined to process mining enables leveraging goals to improve process models and their realization. Process models that are discovered with respect to different goals are aligned with such goals and hence more likely to produce high levels of satisfaction.

The objective in this paper is to offer a process mining method concerned not only with the sequencing of activities, but also with processes’ goals and satisfaction indicators. To this end, we propose a *goal-oriented process enhancement and discovery* (GoPED) method that adds satisfaction levels of different goals to event logs and considers traces of activities beside their contribution to predefined goals. Goal satisfaction levels are derived from a model capturing goals, stakeholders, and their relationships. Note that the “*enhancement*” part of GoPED is about enhancing logs with goal information to produce higher-quality and simpler process models, and not about improving processes after their discovery.

As an example, a trace of activities in a healthcare process may take a very short time (i.e., it satisfies the goal “*to decrease process time*”) but may end up with a wrong diagnosis (i.e., it violates the goal “*to diagnose correctly*”). Inversely, a trace may take a long time and impose an unaffordable cost but may end up with a correct diagnosis. GoPED takes advantage of goal models to manage such conflicting goals and to support trade-off analysis. With GoPED, good historical experiences will be found within the whole event log to be used as a basis for inferring good models and bad experiences will be found to be avoided. The *goodness* of traces and models is defined with regards to three categories of goal-related criteria: satisfaction of individual cases in terms of some goals (*case perspective*), overall satisfaction of some goals rather than individual cases (*goal perspective*), and a comprehensive satisfaction level for all goals over all cases (*organization perspective*). GoPED is expected to guide process discovery approaches towards specific goal-related properties of interest.

The paper is structured as follows. Section 2 explains the fundamentals of GoPED and highlights its contribution to current process mining approaches. Section 3 explicitly describes three algorithms for selecting traces according to the three categories of criteria discussed in the previous paragraph. Then, in Sect. 4, the GoPED method is applied to an illustrative example of a healthcare process, with a discussion of the results. Related work at the intersection of process mining and goal modeling is briefly discussed in Sect. 5. Finally, a summary and conclusions are provided in Sect. 6.

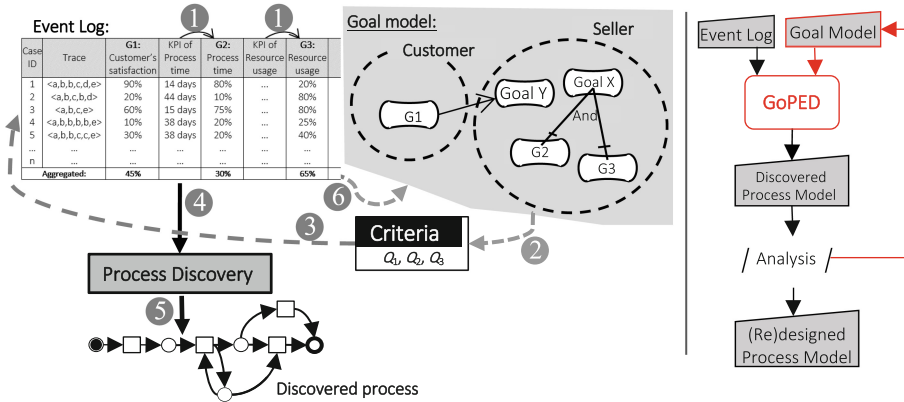


Fig. 1. Overview of goal-oriented process enhancement and discovery (GoPED)

2 GoPED Method

Figure 1 gives an overview of the proposed method through an example that exploits the Goal-oriented Requirement Language (GRL) standard [2]. Let us assume that there are three leaf goals (G1, G2 and G3), with G1 contributing to Goal Y and Goal X being AND-decomposed by G2 and G3. Each goal may be fed by its Key Performance Indicators (KPIs), allowing to quantify its satisfaction level. Let us also assume that the event logs store the value of each goal-related KPI, e.g., “process time” associated to the goal G2 “to take a short process time”. Such KPIs and how they contribute to goal satisfaction (e.g., by providing a function that converts a current, observable value to an abstract satisfaction value between 0/violated and 100/satisfied) are also defined in the goal model (see [1] for details). Based on this scheme, the satisfaction level of the leaf goals in Fig. 1 are computed from corresponding KPIs (arrows 1). We define the satisfaction level of Goal i as $Sat(G_i)$. The satisfaction of actors (dashed circles in Fig. 1) and of the whole models can be computed in a similar fashion [1].

After finding the current satisfaction of considered goals, GoPED defines some criteria related to the goals (arrow 2). The main objective is to design a process model that fulfills one or many such criteria. The goal-related criteria are defined from three perspectives as follows:

- The resulting process model achieves (based on current evidence) a minimum satisfaction level for every single case in terms of one or multiple goals (*row or case perspective*). For example, in Fig. 1, $Sat(G_2)$ should be more than 60 for all cases.
- The resulting process model achieves a threshold for the *aggregated* satisfaction level of one or multiple goals rather than the level for individual cases (*overall column or goal perspective*). For example, in Fig. 1, the aggregated satisfaction level of G2 (where the aggregation function is defined as the *average* here) should be higher than 70 (but is currently 45).
- The resulting process model achieves a threshold for the comprehensive satisfaction level of many goals over all cases (*table or organization perspective*, arrow 6).

This may be computed through the goal model (e.g., in GRL) or through a function derived from that model. For example, according to the structure of the model in Fig. 1, the satisfaction of the stakeholder “Seller” is the average of Goal Y (computed from G1) and Goal X (the minimum of G1 and G2).

The basis of process mining is generally to use historical event logs and infer valuable insights. Following this general approach, GoPED selects a *subset* of the input traces that have already fulfilled the given criteria and uses them to find process models of interest (arrow ④). For example, if the objective is to secure at least 50 as a satisfaction level for G2 for all the customers, the cases #1 and #3 will be selected because they have a satisfaction level over 50 for goal G2. After such a selection, a process model is mined through the selected traces using a process discovery algorithm (arrows ④ and ⑤). The discovered model does not represent all existing behaviours, but rather represents the desired behaviours towards the goals. Different model mined through different goal-related criteria can shed some light on different aspects and alternatives involved in the real or the discovered model. Such criteria are purposely defined by a domain expert in collaboration with a modeller. Moreover, an analyst can compare the model discovered from the whole log with the model discovered by GoPED. Such a comparison can also reveal some valuable insights from potential discrepancies. Goal-oriented conformance checking approaches [5, 6] can also suggest some way of reconsidering the goal model with respect to misalignments between the process and goal perspectives, as shown in the right graph of Fig. 1.

The process model resulting from GoPED is inferred from cases selected based on their goals. Therefore, irrelevant cases (that likely pursue goals different from the expected goals) are filtered out. The discovered model will be more likely well-structured as GoPED intentionally decreases the number of variations of traces and, in turn, decreases the chance of producing a spaghetti-like process model.

Another benefit of GoPED relates to the quality dimensions considered in usual process mining activities. In addition to the *fitness*, *precision*, *generalization*, and *simplicity* dimensions [16], GoPED brings into consideration a new *intention* dimension, formalized by the goal model.

3 Algorithms to Select Cases

In process mining, the three attributes (columns) that must minimally exist in an event table are *case identifier*, *activity* and *timestamp*. There might be some other event attributes stored in such a table that can be used for the analysis of discovered models (e.g., *resource*). Similarly, there might be some attributes about the case (e.g., age) or about a case’s trace (e.g., total process time). In GoPED, we add some new case attributes related to goals, which are usually absent in process mining practice. Table 1 shows the architecture of event logs enhanced with goal-related attributes, used as input of GoPED.

Table 1. Event log enhanced with n goal-related attributes (*EnhancedLog*)

Case	Trace	Goal 1	Goal 2	...	Goal n	Overall
c_1	t_1	$s_{1,1}$	$s_{1,2}$...	$s_{1,n}$	$s_{1,Ove}$
c_2	...	$s_{2,1}$	$s_{2,2}$...	$s_{2,n}$	$s_{2,Ove}$
...
c_m	t_m	$s_{m,1}$	$s_{m,2}$...	$s_{m,n}$	$s_{m,Ove}$
<i>Aggregated satisfaction:</i>		$s_{Agg.1}$	$s_{Agg.2}$...	$s_{Agg.n}$	s_{Comp}

3.1 Preliminaries

The notations that are used through this paper are defined as follows:

Definition 1. Basic concepts (*activity, trace, case, event log*).

- A is the set of all experienced *activities* labelled a_i .
- A *trace* is a finite sequence of activities $t = \langle a_1, \dots, a_k \rangle$, where $k \in \mathbb{N}^+$ is the trace length. T is the set of all observed traces.
- A *case* $c = \langle id, t \rangle$ has a case identifier $id \in \mathbb{N}^+$ and contains a trace $t \in T$.
- $trace(c) = t$ is a shorthand to indicate that the trace of the case c is $t \in T$.
- $L = \langle c_1, \dots, c_m \rangle$ is an *event log* consisting of a finite sequence of cases of size m .
- C is the set of all possible cases (with traces) represented in the log L .

Definition 2. *EnhancedLog* structure.

To select the best subset of cases in a log through GoPED's algorithms, an event log enhanced with additional goal-related attributes is needed. The structure of such log, shown in Table 1, and the elements of that structure are defined as follows:

- *EnhancedLog* is the event log L enhanced with goal-related attributes. This log is a table of all cases $c \in C$ beside their traces $t \in T$. The satisfaction levels of all considered goals (including KPIs and actors) are stored in the next columns. \mathbb{G} is the set of all considered goals, i.e., $\mathbb{G} = \{G_1, G_2, \dots, G_n\}$. We assume that *EnhancedLog* consists of m cases and n considered goals.
- $s_{i,j}$ in *EnhancedLog* shows the level of satisfaction of *case* $_i$ in terms of *Goal* $_j$.
- $s_{i,Ove}$, found in the last column of the table *EnhancedLog*, is the overall satisfaction level for *case* $_i$. This represents the satisfaction level of the whole goal model. The satisfaction level of the goal model is evaluated through bottom-up analysis as elaborated in the goal-oriented modeling literature [1]. This evaluation is based on AND/OR refinements, contribution links, the importance level of a goal to its actor, and the actor importance in the whole model.
- Function g is derived from the goal model to compute the overall satisfaction level based on satisfaction levels of all sub-goals in \mathbb{G} [4]. Therefore, as $s_{i,j}$ is the satisfaction level of *Goal* $_j$ for *case* $_i$, we have $s_{i,Ove} = g(s_{i,1}, s_{i,2}, \dots, s_{i,n})$.
- $s_{Agg.i}$, in the last row of *EnhancedLog*, show the aggregated satisfaction level of each goal based on the satisfaction level of all cases in terms of that goal. $s_{Agg.i}$ is a function (e.g., average, median, etc.) of satisfaction levels of all m cases for *Goal* $_j$.

- Function f_j is the aggregation function of $Goal_j$. For each goal $G_j \in \mathbb{G}$ we have $s_{Agg,j} = f_j(s_{1,j}, s_{2,j}, \dots, s_{m,j})$. F is a tuple of functions (f_1, f_2, \dots, f_n) that keeps aggregation functions of all goals.
- s_{Comp} is the comprehensive satisfaction level that the process has yielded. This factor can be defined either by composing the aggregated satisfactions (last row) or by aggregating the overall satisfaction levels (last column) using some function.

Definition 3. GoPED offers three types of goal-related criteria, discussed in Sect. 3.1:

- Q_{case} is a set of criteria q_j . Each q_j is a tuple composed of one goal $G_j \in \mathbb{G}$ and a threshold, sl_j , for the satisfaction level of that goal, $Q_{case} = \{q_j = (G_j, \overline{sl}_j) | G_j \in \mathbb{G} \wedge 0 \leq \overline{sl}_j \leq 100\}$. A confidence level, $0 \leq conf \leq 1$, together with Q_{case} , constitute the whole criteria. Such criteria represent that (with a confidence $conf$) the satisfaction level of every single case in terms of the considered goals G_j will be at least \overline{sl}_j . It is noteworthy that all goals in \mathbb{G} are not necessarily considered by Q_{case} . For example, when $\mathbb{G} = \{G_1, G_2, G_3\}$ and $Q_{case} = \{(G_2, 90), (G_3, 75)\}$ and $conf = 0.8$, GoPED is looking for a process model that will yield minimum satisfaction levels of 90 for G_2 and 75 for G_3 , for at least 80% of the cases (i.e., confidence of 0.8).
- Q_{goal} refers to the second type of goal-related criteria. Q_{goal} consists of a set of criteria q_j composed of one goal $G_j \in \mathbb{G}$ and a satisfaction level for that goal. $Q_{goal} = \{q_j = (G_j, \overline{sl}_j) | G_j \in \mathbb{G} \wedge 0 \leq \overline{sl}_j \leq 100\}$. Q_{goal} is looking for a process model that can deliver an *aggregated* satisfaction level for the considered $G_j \in \mathbb{G}$ of at least \overline{sl}_j . Again, all goals in \mathbb{G} are not necessarily considered by Q_{goal} . For example, when $\mathbb{G} = \{G_1, G_2, G_3\}$ and $Q_{goal} = \{(G_2, 90), (G_3, 75)\}$, Q_{goal} is looking for a process model that will yield minimum *aggregated* satisfaction levels of 90 for G_2 and 75 for G_3 .
- Q_{Comp} consists of one value between 0 and 100 called \overline{sl}_{Comp} . This criterion looks for a process model that can yield a *comprehensive* satisfaction of at least \overline{sl}_{Comp} .

Definition 4. $SelectedCases \subseteq C$ is the main output of GoPED algorithms and the set of selected cases that satisfy one of the aforementioned criteria.

3.2 GoPED Algorithms

As the goal-related criteria are based on three different viewpoints of *EnhancedLog*, three different algorithms for trace selection are required. The main idea in all three algorithms is to select the largest subset of cases that satisfy the selected criterion.

Searching for the *largest* subset of cases is needed because if one simply selects very few cases that meet the desired criteria, the discovered model will be based on an event log suffering from potential *incompleteness* problems. When the event log consists of too few events, the discovered model is less realistic and risks becoming overfitted.

Another feature of our search approach is that we look over the *cases* $\in C$ rather than the *traces* $\in T$. This is because many cases might have a same trace but different

levels of satisfaction for the goals. Moreover, the frequency of each trace contains very important knowledge about real-world behaviors. Therefore, we need to end up with a subset that consists of variations of traces together with their frequencies.

Algorithm 1 Selecting a subset of an event log to infer a process model that guarantees a minimum satisfaction level for one or multiple goals in each selected case

Input: *EnhancedLog*: An enhanced structured event log \triangleright explained in Definition 2
 Q_{case} : A set of criteria; \triangleright explained in Definition 3
 conf : a confidence level \triangleright explained in Definition 3
Output: *SelectedCases* \triangleright a subset of cases selected according to the criteria and the all-or-none rule

```

1  sort_by_trace(EnhancedLog)  $\triangleright$  sort the cases based on their traces
2   $\text{trace}(\text{case}_0) \leftarrow \langle \rangle$   $\triangleright \langle \rangle$  is an empty trace, which cannot happen in reality
3   $\text{trace}(\text{case}_{\text{NumberOfCases}+1}) \leftarrow \langle \rangle$   $\triangleright$  also flag the end of the log
4  SelectedCases  $\leftarrow \emptyset$ 
5  index  $\leftarrow 1$ 
6  while index  $\leq$  NumberOfCases  $\triangleright$  NumberOfCases is  $m$  in Table 1
7  |   SameTraceCases  $\leftarrow \emptyset$   $\triangleright$  a set of cases whose traces are the same
8  |   NumberOfSatisfiedCasesOfTrace  $\leftarrow 0$   $\triangleright$  counts the satisfied cases of a trace
9  |   do
10 |     SameTraceCases  $\leftarrow$  SameTraceCases  $\cup$  {caseindex}
11 |     if caseindex meets all criteria of  $Q_{\text{case}}$  then
12 |       NumberOfSatisfiedCasesOfTrace ++
13 |     end if
14 |     index ++
15 |     while  $\text{trace}(\text{case}_{\text{index}}) = \text{trace}(\text{case}_{\text{index}-1})$ 
16 |     if NumberOfSatisfiedCasesOfTrace / size(SameTraceCases)  $\geq$   $\text{conf}$  then
17 |       SelectedCases = SelectedCases  $\cup$  SameTraceCases
18 |     end if
19 |   end while
20 return SelectedCases  $\triangleright$  the resulting subset of cases

```

One consequence of searching within cases is that there might be some cases with trace t_k that are eligible to be selected and, simultaneously, some cases with the same trace that are not. Although including the former cases and excluding the latter ones appears to be a simple solution, it would not be correct. The reason is that a discovered model either allows a trace (and all its cases) or avoids it. We respect an “all-or-none” rule, i.e., the *SelectedCases* should have either all cases of a same trace or none of them. Based on the above explanation, we define the three algorithms for selecting the best subset of cases regarding the three types of goal criteria and the all-or-none rule.

Algorithm 1: Guaranteeing One or Multiple Goals for All Cases. This type of goal-related criterion is looking for a model that guarantees (with a given confidence level) a predefined satisfaction level for one or multiple goals for all cases. This criterion considers every single case in a row viewpoint, therefore each case will be assessed against all goals considered in the criterion. There might be cases with trace t_k that meets all $q_i \in Q_{\text{case}}$ and some cases with the same trace that do not. Algorithm 1 checks all cases of one trace against Q_{case} (line 11). If the proportion of complying cases is not

inferior to the given confidence level $conf$, all the cases with that trace will be selected, otherwise all of them will be filtered out (lines 16–17). For example, assume $conf$ is 0.8 and the event log has 100 cases with trace $\langle a, b, c, g \rangle$, including 83 cases that meet Q_{case} and 17 cases that do not. As 83% of these cases comply with the criterion, which is above the confidence level of 80%, all 100 cases are selected. Algorithm 1 first sorts all cases according to their trace (line 1). Searching within all cases of a trace and checking them against the criteria is hence efficient (lines 9–15).

Algorithm 2 Selecting a subset of an event log to infer a process model that guarantees the overall satisfaction level(s) of one or multiple goals

Input: *EnhancedLog*: An enhanced structured event log \triangleright explained in Definition 2

Q_{goal} : A set of criteria (some goals and thresholds for their satisfaction level) \triangleright Definition 3

g : A function computing the satisfaction of the whole goal model \triangleright Definition 3

Output: *SelectedCases* \triangleright a subset of all cases selected regarding the criteria and the all-or-none rule, $SelectedCases \subseteq C$, Definition 4

1 $SelectedCases \leftarrow \emptyset$

2 **Solve the binary optimization below:** (x_i is a flag for either selecting case i or not)

$Max z = \sum_{i=1}^m x_i \triangleright$ this is to find the largest subset

s.t.

$\forall r, t \ 1 \leq r < t \leq m : \text{if } trace(c_r) = trace(c_t) \ x_r = x_t \triangleright$ all-or-none rule

$\forall j \text{ where } G_j \in \mathcal{G} : \frac{\sum_{i=1}^m x_i \cdot s_{i,j}}{\sum_{i=1}^m x_i} \geq \overline{s}_j \triangleright |Q_{goal}| \text{ constraints}$

$x_i = 0, 1 \triangleright$ if $x_i = 1$, case i should be selected

3 **end of binary optimization**

4 **for** $i = 1$ to $NumberOfCases$ **do** $\triangleright NumberOfCases$ is m in Table 1

5 **if** $x_i = 1$ **then**

6 $SelectedCases \leftarrow SelectedCases \cup \{c_i\}$

7 **end**

8 **end**

9 **return** *SelectedCases* \triangleright the resulting subset of cases that meets the criteria

Algorithm 2: Guaranteeing the Aggregated Satisfaction Levels of Goals. Here, in a column perspective, the focus is on the *aggregated* satisfaction level of one or multiple goals. Logically, the largest subset that simultaneously meets all criteria is the intersection of the largest subsets that separately meet all criteria. Therefore, one can focus on all considered goals individually and find the largest subsets for each G_j regarding \overline{s}_j , then use their intersection as *SelectedCases*. However, finding the largest subset is not trivial because the largest subset that satisfies the criterion of one goal might not be unique and different subsets with similar (and largest) sizes may satisfy the condition. This might be the case even when the aggregation function is simple (e.g., *average*). In this situation, a subset that makes the largest intersection of all subsets (generated by all considered goals) should be selected. If this situation happens for several goals, we have to deal with the challenge of selecting one combination that finally makes the largest set.

Addressing such a difficulty, Algorithm 2 generates a binary optimization whose number of variables (x_i) equals the number of cases (m). The binary variable x_i is a flag variable associated to case c_i . If $x_i = 1$, then the case c_i will be selected and, if not, it will be excluded. As we are looking for the largest subset, $\sum x_i$ should be as large as possible. There are two categories of constraints for the optimization problem. The first aims to preserve the *all-or-none* rule, i.e., the selected subset should have either all cases of a same trace or none of them. The second category of constraints takes care of the threshold for the aggregated satisfaction level of each goal, i.e., \overline{sl}_j . This category of constraints is based on f_j , i.e., the aggregation function. In Algorithm 2 we assumed that all f_j are the *average* function (but others could be defined).

Algorithm 3 Selecting the largest subset of an event log to infer a process model that guarantees a comprehensive satisfaction level

Input: *EnhancedLog*: An enhanced structured event log; \triangleright as explained by Definition 2

$Q_{Comp} : \overline{sl}_{Comp}$ \triangleright a minimum threshold for comprehensive satisfaction level

F : a tuple of functions $(f_1, f_2, \dots, f_n) \mid s_{Agg,j} = f_j(s_{1,j}, s_{2,j}, \dots, s_{m,j}), j = 1, \dots, n$

g : a function derived from the goal model. $s_{i,ove} = g(s_{i,1}, s_{i,2}, \dots, s_{i,n})$

Output: *SelectedCases* \triangleright a subset of all cases selected regarding the criteria and the all-or-none rule, $SelectedCases \subseteq C$

```

1  SelectedCases  $\leftarrow \emptyset$ 
2  if  $s_{Comp} = f_{n+1}(s_{1,ove}, s_{2,ove}, \dots, s_{m,ove})$  then
3      use Algorithm 2 and exit.
4  else
5      Solve the binary optimization below:

```

Max $z = \sum_{i=1}^m x_i$ \triangleright this is to find the largest subset.
s.t.
 $\forall r, t \ 1 \leq r < t \leq m : \text{if } trace(c_r) = trace(c_t) \ x_r = x_t$ \triangleright all-or-none rule
 $g\left(\frac{\sum_{i=1}^m x_i \cdot s_{i1}}{\sum_{i=1}^m x_i}, \frac{\sum_{i=1}^m x_i \cdot s_{i2}}{\sum_{i=1}^m x_i}, \dots, \frac{\sum_{i=1}^m x_i \cdot s_{in}}{\sum_{i=1}^m x_i}\right) \geq \overline{sl}_{Comp}$
 $x_i = 0, 1$

```

6  end
7  for  $i = 1$  to NumberOfCases do  $\triangleright$  NumberOfCases is  $m$  in Table 1
8      if  $x_i = 1$  then
9          SelectedCases  $\leftarrow SelectedCases \cup \{c_i\}$ 
10     end
11 end
12 return SelectedCases  $\triangleright$  the resulting subset of cases that meets the criterion

```

Algorithm 3: Guaranteeing Comprehensive Satisfaction Levels. The two above types of criteria considered the goals from a row perspective and a column perspective. The third type of criteria, however, considers the goals from a *table* perspective.

Here, the overall satisfaction level of all columns is aggregated and represented by one number as a *comprehensive satisfaction level*. Finding the largest subset of cases that guarantees a minimum threshold for comprehensive satisfaction level (\overline{sl}_{comp}) is,

Table 2. *EnhancedLog*, event log and satisfaction level of goals for the DGD process

Case	Trace	G ₁ : To decrease process time	G ₂ : To decrease cost	G ₃ : To do a smooth process	G ₄ : To screen accurately	Overall (To satisfy the patient)
Patient#1	$\langle a, b, c, g \rangle$	100	100	88	100	97
Patient#2	$\langle a, b, c, g \rangle$	94	100	88	100	95
Patient#3	$\langle a, b, c, g \rangle$	94	100	88	0	0
Patient#4	$\langle a, b, c, d, e, c, g \rangle$	61	59	75	100	64
Patient#5	$\langle a, b, c, d, e, c, g \rangle$	72	59	63	100	65
Patient#6	$\langle a, b, c, d, e, c, g \rangle$	67	59	75	100	66
Patient#7	$\langle a, b, c, f, b, c, g \rangle$	78	82	63	100	76
Patient#8	$\langle a, b, c, f, b, c, d, e, c, g \rangle$	41	20	50	100	36
Patient#9	$\langle a, b, c, f, b, c, d, e, c, g \rangle$	43	20	40	100	34
Patient#10	$\langle a, b, c, d, b, c, d, e, c, g \rangle$	9	10	30	100	15
<i>Aggregated satisfaction:</i>		65.9	60.9	66	90	64.1

also, not trivial. This is because adding a trace to the subset might increase the aggregated level of one goal and, at the same time, decrease the level of another goal.

As explained in Definition 3, the *comprehensive* level might be calculated in two different ways. It can be the overall satisfaction level of the aggregated levels of all goals, or the aggregated level of the overall satisfaction level of each case. The latter one will work only with the column of the overall satisfaction level (the right column of Table 1), therefore, the problem will be solved in a way similar to that of the second type of criteria (Algorithm 2). Accordingly, Algorithm 3. first checks the definition of *comprehensive*. If it is not like Algorithm 2, then Algorithm 3 generates a new binary optimization problem. Here, the first category of constraints aims to preserve the *all-or-none* rule, whereas the last constraint makes sure that the comprehensive level of the selected subset is not less than the given threshold \overline{sl}_{comp} .

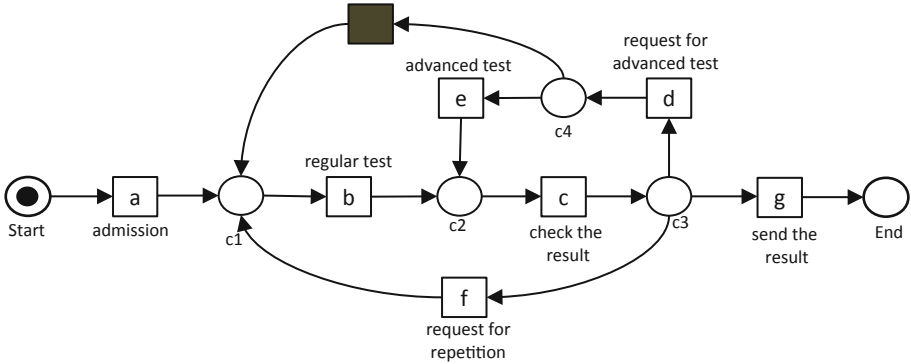
4 Illustrative Example

The process of *diagnosis of gestational diabetes* (DGD) will be used to illustrate the proposed methods. To this end, three types of goal-related criteria discussed in Sect. 2 will be taken into consideration. The main assumption here is that the log is realistic but *not real* and is used only to study the GoPED method and its algorithms.

4.1 Event Log of an Illustrative DGD Process

The event log of 10 patients who have used the DGD process is shown in Table 2. We use short names to encode the activities: a = admission, b = regular test, c = check the result, d = request for advanced test, e = advanced test, f = request for repetition, and g = send the result. According to Table 2 and the definitions described in Sect. 3.1, the event log (L) includes five different variants of traces:

$$L = [\langle a, b, c, g \rangle^3, \langle a, b, c, d, e, c, g \rangle^3, \langle a, b, c, f, b, c, g \rangle^1, \langle a, b, c, f, b, c, d, e, c, g \rangle^2, \langle a, b, c, d, b, c, d, e, c, d, e, c, g \rangle^1]$$



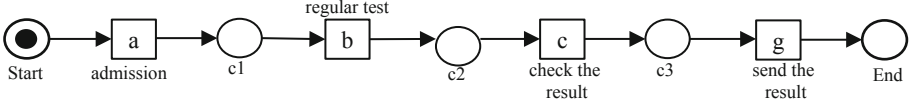
Model 1. Process model discovered from the original event log by the α -algorithm

As shown in Table 2, there are four additional goal-oriented fields related to the DGD process. Due to space limitation, we directly show the satisfaction levels of the goals in the table (without their indicators or actors), which are values in the range [0–100].

An advanced version of the α -algorithm [17] generates Model 1 using the whole event log. The main story of this DGD process is as follows: after admission of a patient, a regular blood test is done. Then, based on the result of the test the patient may need to do an advanced test, the patient may need to repeat the regular test, or the result will be sent to the related department, and then the process ends. A *silent* transition is shown in Model 1 with black color. That is a particular transition not observable in the event log, but needed to make a *sound* Petri-net [16]. Considering the traces, the source of this need is that for Patient#10, after the activity *d*, request for advanced test, the activity *b*, regular test, has executed, while the activity *e*, advanced test, was supposed to execute. Model 1 will be used as a basis for considering the resulting models from GoPED respecting three types of goal-related criteria.

4.2 Example Models Resulting from GoPED

Guaranteeing Satisfaction of One or Multiple Goals for All Cases. This goal-related criterion is looking for a model that guarantees a predefined satisfaction level for all cases in terms of one or multiple goals, with a given confidence level. For example, the condition is as follows:



Model 2. To satisfy goal criteria of Q_{case}

- *Case perspective:* generate a model that guarantees (with a confidence of 90%) that the satisfaction level for all patients in terms of goal “To decrease process time” will be at least 75 and that in terms of goal “To do a smooth process” will be at least 80.

In this case, we have $Q_{\text{case}} = \{(G_1, 75), (G_3, 80)\}$ and $\text{conf} = 0.9$. Using Algorithm 1, only all the cases of trace $\langle a, b, c, g \rangle$ are returned, i.e., Patients #1, #2 and #3. All cases of this trace meet Q_{case} , so the fraction of eligible cases of this trace is 100%, which is more than the required 90% confidence level. Such a parameter for the four remaining traces is zero, which is less than 90% by far. Therefore, we have $\text{SelectedCases} = \{\text{Patient\#1}, \text{Patient\#2}, \text{Patient\#3}\}$, resulting in the log $\{\langle a, b, c, g \rangle^3\}$. The α -algorithm [17] produces Model 2 from this log. This is the process to be encouraged in the organization to meet these goals.

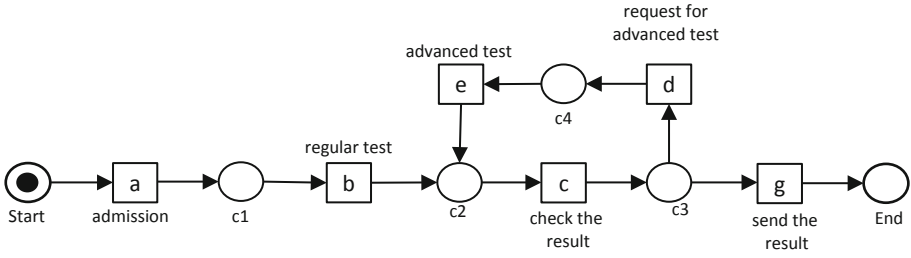
Guaranteeing the Aggregated Satisfaction Levels of Goals. Here, from a column perspective, the focus is on the aggregated satisfaction level of one or multiple goals rather than on the satisfaction of every single case.

- *Goal perspective:* Generate a model that results in an *aggregated* satisfaction levels of the goal “To decrease time process” higher than 80 and of the goal “To do a smooth process” higher than 78, simultaneously.

In this case, we have $Q_{\text{goal}} = \{(G_1, 80), (G_3, 78)\}$. Here the functions f showing how to calculate the aggregation of each column is required. Let us assume that for all goals in the DGD process, the function is the *average*. Therefore, the optimization problem of Algorithm 2 can be formalized as follows:

$$\begin{aligned}
 & \text{Max } z = \sum_{i=1}^{10} x_i \\
 & \text{s.t.} \\
 & x_1 = x_2 = x_3, \quad x_4 = x_5 = x_6, \quad x_8 = x_9 \quad \leftarrow (\text{all-or-none rule}) \\
 & \frac{100x_1 + 94x_2 + 94x_3 + 61x_4 + 72x_5 + 67x_6 + 78x_7 + 41x_8 + 43x_9 + 9x_{10}}{\sum_{i=1}^m x_i} \geq 80 \\
 & \frac{88x_1 + 88x_2 + 88x_3 + 75x_4 + 63x_5 + 75x_6 + 63x_7 + 50x_8 + 40x_9 + 30x_{10}}{\sum_{i=1}^m x_i} \geq 78 \\
 & x_i = 0, 1
 \end{aligned}$$

The answer of the above problem is unique: $x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = 1$ and $x_7 = x_8 = x_9 = x_{10} = 0$. Therefore, $\text{SelectedCases} = \{\text{Patient\#1}, \text{Patient\#2}, \text{Patient\#3}, \text{Patient\#4}, \text{Patient\#5}, \text{Patient\#6}\}$, leading to the log $\{\langle a, b, c, g \rangle^3, \langle a, b, c, d, e, c, g \rangle^3\}$. For this subset of cases, the aggregation satisfaction level for G_1 and G_3 will be 81.3 and 79.5, respectively. The α -algorithm produces Model 3 from this log.



Model 3. To satisfy goal criteria of Q_{goal}

Guaranteeing Comprehensive Satisfaction Levels. Here, from a table perspective, the focus is on the *comprehensive* satisfaction level, which we assume to be the *overall* satisfaction level of the *aggregated* levels of all goals. Therefore, the goal model should be used to evaluate s_{Comp} based on the satisfaction levels of all sub goals. Figure 2 shows the goals model related to the DGD process using the GRL language. In the graph, the root is the main goal and the sub goals are the leaves. Based on the goal model and its AND/OR refinements and the weight of contributions, the s_{Comp} is defined as follows:

$$s_{Comp} = Sat(G_6) = \text{Minimum}(s_{Agg.4}, 0.4 \times s_{Agg.1} + 0.35 \times s_{Agg.2} + 0.25 \times s_{Agg.3})$$

This kind of evaluation is known as *forward propagation* in GRL. The jUCMNav tool is an Eclipse-based graphical editor that can be used for evaluating GRL models [9].

- *Organization perspective:* Generate a model where the comprehensive satisfaction level is no less than 75.

The above criterion leads to $Q_{Comp} = 75$. Recall that $s_{Agg.j}$ is the aggregated satisfaction of goal j , in our case the *average* of column of Goal $_j$ in Table 2. According to the function derived from the goal model of Fig. 2, Algorithm 3 generates the optimization problem as follows:

$$\begin{aligned} \text{Max } z &= \sum_{i=1}^{10} x_i \\ \text{s.t.} & \\ x_1 &= x_2 = x_3, \quad x_4 = x_5 = x_6, \quad x_8 = x_9 \quad \leftarrow (\text{all-or-none rule}) \\ \text{Minimum} &\left(\frac{\sum_{i=1}^m x_i \cdot s_{i,4}}{\sum_{i=1}^m x_i}, 0.4 \times \frac{\sum_{i=1}^m x_i \cdot s_{i,1}}{\sum_{i=1}^m x_i} + 0.35 \times \frac{\sum_{i=1}^m x_i \cdot s_{i,2}}{\sum_{i=1}^m x_i} + 0.25 \times \frac{\sum_{i=1}^m x_i \cdot s_{i,3}}{\sum_{i=1}^m x_i} \right) \geq 75 \\ x_i &= 0, 1 \end{aligned}$$

(s_{ij} refers to the cells of Table 2, e.g., $s_{2,1} = 94$).

The answer of the above problem is, also, unique: $x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = x_7 = 1$ and $x_8 = x_9 = x_{10} = 0$. Therefore, *SelectedCases* = {Patient#1, Patient#2, Patient#3, Patient#4, Patient#5, Patient#6, Patient#7}, resulting in the log $\{\langle a, b, c, g \rangle^3, \langle a, b, c, d, e, c, g \rangle^3, \langle a, b, c, f, b, c, g \rangle\}$. For this subset, the *comprehensive satisfaction level* is 79.5. The α -algorithm produces Model 4 from this log.

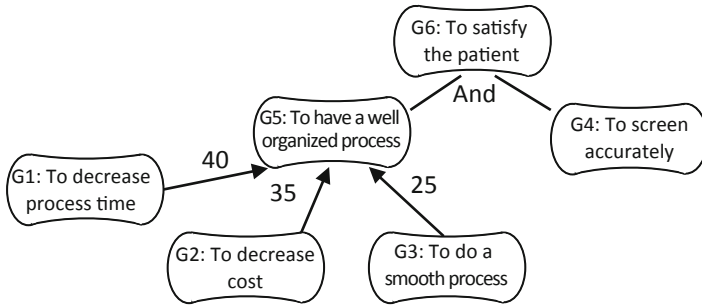
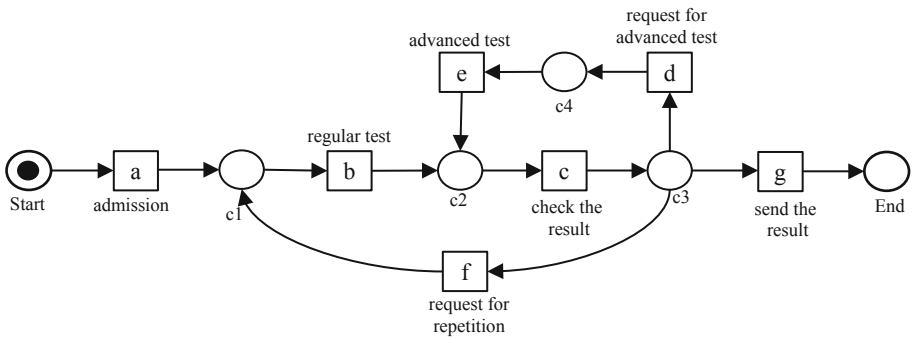


Fig. 2. Goal model showing the relations between the goals pursued by the DGD process



Model 4. To satisfy goal criteria of Q_{Comp}

4.3 Discussion

We generated three models using three types of goal-related criteria. Comparing the models, we find that the main difference between Model 2 and the other models relates to the loops. Model 2 does not allow repeating the regular test or to do an advanced test. Here, the decision point “check the result” is spurious as it will not actually make any decision. According to the goals considered in the generation of such a model, i.e., G1 and G3, one can hypothesize that doing advanced blood test or repeating the regular test are not aligned with the goals of having short process time and of providing all patients with a smooth process. Considering the trace $\langle a, b, c, g \rangle$ that Model 2 can generate, we find that there are three patients who experienced this trace. One of these patients has been diagnosed wrongly ($s_{3,4} = 0$). The goal model shown in Fig. 2 implies that the goal “To screen accurately” participates to an AND refinement; therefore, when this goal is denied, regardless of the other goals in the refinement, the main goal of the process gets denied. Using the goal model, the overall satisfaction level of those three patients are 97, 95 and 0, respectively. This suggests that although Model 2 highly satisfies all cases in terms of process time and smoothness of the process, it will end up with a third of the patients who will be dissatisfied.

The above analyses are simple examples of knowledge that GoPED can provide. Such knowledge, together with the discovered models, can help domain experts (re)design goal-aligned process models, encourage good behaviors, and discourage bad ones.

5 Related Work

Our systematic literature review of goal-oriented process mining showed that although process mining and goal modeling are growing research topics, there are only a few rare studies conducted at their intersection [7]. Therefore, this suggests that goal-oriented processes discovery can still be considered a gap to be filled between the process mining and requirements engineering communities.

From an *agent viewpoint*, the goals behind activities of agents who contribute in a process (e.g., employees) are considered by Yan et al. [18]. Their proposed approach adopts a decision tree algorithm to learn goals of agents by classifying their activities in different situations. In this viewpoint, Outmazgin and Soffer [11] used process discovery techniques to analyze different types of intentional incompliances, where employees intentionally deviate from prescribed models, to find their causes.

In addition, in a *process viewpoint* (or case/customer viewpoint) all activities constituting a trace are considered. Ponnalagu et al. [13] proposed an approach for analyzing and validating a family of variants of a single process based on a goal model. From the same view point, Horita et al. [8] proposed a method to detect and analyze the effects of disagreements between real logs and prescribed models using a goal-oriented conformance checking approach. Bernard and Andritsos [3] used process discovery in conjunction with customers' journeys and developed a tool that facilitates navigation through many different journeys in a goal-oriented fashion.

An *organization viewpoint* considers the overall goals that should be achieved by performing business processes. Santiputri et al. [14] considered the sequence of events in multi-layered event logs and proposed an approach to discover goal refinement patterns of the goal models.

Trace clustering is a solution proposed by the process mining community to improve interpretability of discovered process models by splitting different behaviors on different process perspectives into multiple sub-logs. Similar to our approach, the main idea in the existing clustering approaches is to discover models from subsets of logs. However, the clustering approach yet considers the log at an activity-level and does not bring the satisfaction of competing goals of stakeholders into account [10, 15].

6 Conclusion

This paper proposed a new method that exploits the capabilities of goal modeling and performs process discovery in a goal-oriented fashion. This method first enhances an event log by adding new goal-related information to all traces. Then, it quantifies the satisfaction level of goals using a goal model. Such a goal model shows correlations between (often conflicting) goals of different stakeholders and allows what-if analysis

and balancing trade-offs between confliction goals. Three types of goal-related criteria were introduced as the basis for generating goal-oriented models promising to achieve predefined goals. The real behaviors that are aligned with the goals and achieve desired satisfaction levels are selected. Three algorithms for such a selection were explicitly explained. The selected subset becomes the basis for conventional process discovery. The resulting model can be compared to a model discovered from the original event log to reveal new insights about the ability of different forms of process models to satisfy the goals. Learning from *good* behaviors that satisfy goals and detecting *bad* behaviors that hurt them is an opportunity to redesign models so they are better aligned with goals.

Acknowledgment. This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC, Discovery and CGS-D).

References

1. Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., Yu, E.: Evaluating goal models within the goal-oriented requirement language. *Int. J. Intell. Syst.* **25**(8), 841–877 (2010). <https://doi.org/10.1002/int.20433>
2. Amyot, D., Mussbacher, G.: User requirements notation: the first ten years, the next ten years. *J. Softw.* **6**(5), 747–768 (2011)
3. Bernard, B., Andritsos, P.: CJM-ex: goal-oriented exploration of customer journey maps using event logs and data analytics. In: *BPM Demo Track and BPM Dissertation Award (BPMD&DA)*, vol. 1920. EUR-WS (2017)
4. Fan, Y., Anda, A.A., Amyot, D.: An arithmetic semantics for GRL goal models with function generation. In: Khendek, F., Gotzhein, R. (eds.) *SAM 2018*. LNCS, vol. 11150, pp. 144–162. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01042-3_9
5. Ghasemi, M.: Towards goal-oriented process mining. In: *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pp. 484–489. IEEE CS (2018). <https://doi.org/10.1109/re.2018.00066>
6. Ghasemi, M.: What requirements engineering can learn from process mining. In: *2018 1st International Workshop on Learning from other Disciplines for Requirements Engineering (D4RE)*, pp. 8–11. IEEE (2018). <https://doi.org/10.1109/d4re.2018.00008>
7. Ghasemi, M., Amyot, D.: From event logs to goals: a systematic literature review of goal-oriented process mining. *Requir. Eng.* 1–27 (2019). <https://doi.org/10.1007/s00766-018-00308-3>
8. Horita, H., Hirayama, H., Tahara, Y., Ohsuga, A.: Towards goal-oriented conformance checking. In: *Proceedings of the International Conference on Software Engineering and Knowledge Engineering SEKE*, pp. 722–724 (2015)
9. jUCMNav (2016). <http://softwareengineering.ca/jucmnav>
10. Mannhardt, F., de Leoni, M., Reijers, H., van der Aalst, W., Toussaint, P.: Guided process discovery – a pattern-based approach. *Inf. Syst.* **76**, 1–18 (2018). <https://doi.org/10.1016/j.is.2018.01.009>
11. Outmazgin, N., Soffer, P.: A process mining-based analysis of business process work-arounds. *Softw. Syst. Model.* **15**(2), 309–323 (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. Ponnalagu, K., Ghose, A., Narendra, Nanjangud C., Dam, H.K.: Goal-aligned categorization of instance variants in knowledge-intensive processes. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) BPM 2015. LNCS, vol. 9253, pp. 350–364. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_24
14. Santiputri, M., Deb, N., Khan, M.A., Ghose, A., Dam, H., Chaki, N.: Mining goal refinement patterns: distilling know-how from data. In: Mayr, H.C., Guizzardi, G., Ma, H., Pastor, O. (eds.) ER 2017. LNCS, vol. 10650, pp. 69–76. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69904-2_6
15. Seeliger, A., Nolle, T., Mühlhäuser, M.: Finding structure in the unstructured: hybrid feature set clustering for process discovery. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 288–304. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_17
16. van der Aalst, W.: Process Mining Data Science in Action, 2nd edn. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
17. Wen, L., van der Aalst, W., Wang, J., Sun, J.: Mining process models with non-free-choice constructs. *Data Min. Knowl. Disc.* **15**(2), 145–180 (2007)
18. Yan, J., Hu, D., Liao, S., Wang, H.: Mining agents' goals in agent-oriented business processes. *ACM Trans. Manag. Inf. Syst.* **5**(4), 1–22 (2014)



Checking Regulatory Compliance: Will We Live to See It?

Silvano Colombo Tosatto, Guido Governatori, and Nick van Beest^(✉)

Data61, CSIRO, Dutton Park, Queensland, Australia
{silvano.colombo,guido.governatori,nick.vanbeest}@data61.csiro.au,
colombotosatto.silvano@gmail.com

Abstract. Checking regulatory compliance of business processes prior to deployment is common practice and numerous approaches have been developed over the last decade. However, the computational complexity of the problem itself has never received any major attention. Although it is known that the complexity of the problem is generally in **NP**-complete, many existing approaches ignore the issue using the excuse that current problems are small enough to be solved anyway. However, due to the current race towards digitalisation and automatism, the size and complexity of the problems is bound to increase. As such, this paper investigates the computational complexity of all sub-classes of the problem and categorises some of the existing approaches, providing a detailed overview of the issues that require to be tackled in order for current compliance checking solutions to remain feasible in future scenarios.

Keywords: Business process modelling · Regulatory compliance · Computational complexity

1 Introduction

Recently more and more effort is put into digitalising and automatising businesses and services in various sectors. Considering both the importance and complexity of proving regulatory compliance of such automated business models, studying and understanding the details about the computational complexity of such problems becomes of paramount importance.

Current approaches aiming at proving regulatory compliance of business process models (i.e. verifying whether their executions follow the required regulations) already exist. However, due to the limited size and complexity of the models currently used, the computational complexity of these problems has not yet been an insurmountable issue. Nevertheless, due to the increase in automation and digitalisation, the size of the problems and their complexity are also bound to increase. This, in turn, will lead the computational complexity to be an issue in the future.

With this in mind, we study the computational complexity of some variants of the problem identified by considering some properties, namely the number of

regulations to be verified, their expressivity and whether they affect the process partially or entirely. Given these properties, we identify 8 variants. The computational complexity for some of these variants has already been studied by Colombo Tosatto et al. [22,23], showing that the upper bound complexity of the variants of the problem considered in the present paper is **NP**-complete, while the lower bound is in **P**.

While being aware that the regulatory framework is not the only source of computational complexity in the regulatory compliance problem, we have decided to explore and study the variants of the problem obtainable by considering a relatively simple business process model, acyclic structured business processes, and varying the properties of the regulatory framework. Although the computational complexity study provided in this paper does not investigate the whole problem, it still provides an analysis of many variants of the problem. As such, it can be used in the future to start a further computational complexity analysis, such as for instance how the properties of a business process model contribute to the computational complexity of the problem.

Additionally, we analyse some of the existing solutions, and assign them to the variants of the problem identified in this paper. Our goal is to show in this way, which of the currently available solutions are bound to hit the computational complexity wall as automatising compliance proving gains popularity.

The remainder of this paper is structured as follows. Section 2 introduces the problem of proving regulatory compliance of business process model and its semantics. Subsequently, Sect. 3 introduces the classification and acronyms of the variants of the problem studied in this paper, and discusses some of the existing computational complexity results. Next, Sect. 4 extends the computational complexity results and provides additional ones for some of the variants. Section 5 shows how some of the existing approaches aiming at solving the regulatory compliance problem can be assigned to the problem’s variants identified and studied. Finally, Sect. 6 concludes the paper. In order to preserve the readability of the paper, we have collected the computational complexity proofs in the Appendix.

2 Proving Regulatory Compliance

In this section, we introduce the problem of proving regulatory compliance of business process models analysed in this paper. The problem consists of two components: (i) the business process model compactly describing a set of possible executions, and (ii) a regulatory framework, describing the compliance requirements.

2.1 Structured Business Processes

We limit our study to structured process models, as the soundness¹ of such models can be verified in polynomial time with respect to their size, and allows

¹ A process is sound, as defined by van der Aalst [1,24], if it avoids livelocks and deadlocks.

us to reuse some existing results to prove our claims. These processes are similar to structured workflows defined by Kiepuszewski et al. [15].

Definition 1 (Process Block). *A process block B is a directed graph: the nodes are called elements and the directed edges are called arcs. The set of elements of a process block are identified by the function $V(B)$ and the set of arcs by the function $E(B)$. The set of elements is composed of tasks and coordinators. There are 4 types of coordinators: `and_split`, `and_join`, `xor_split` and `xor_join`. Each process block B has two distinguished nodes called the initial and final element. The initial element has no incoming arc from other elements in B and is denoted by $b(B)$. Similarly the final element has no outgoing arcs to other elements in B and is denoted by $f(B)$.*

A directed graph composing a process block is defined inductively as follows:

- *A single task constitutes a process block. The task is both initial and final element of the block.*
- *Let B_1, \dots, B_n be distinct process blocks with $n > 1$:*
 - *SEQ(B_1, \dots, B_n) denotes the process block with node set $\bigcup_{i=0}^n V(B_i)$ and edge set $\bigcup_{i=0}^n (E(B_i) \cup \{(f(B_i), b(B_{i+1})) : 1 \leq i < n\})$. The initial element of SEQ(B_1, \dots, B_n) is $b(B_1)$ and its final element is $f(B_n)$.*
 - *XOR(B_1, \dots, B_n) denotes the block with vertex set $\bigcup_{i=0}^n V(B_i) \cup \{\text{xsplit}, \text{xjoin}\}$ and edge set $\bigcup_{i=0}^n (E(B_i) \cup \{(\text{xsplit}, b(B_i)), (f(B_i), \text{xjoin}) : 1 \leq i \leq n\})$ where `xsplit` and `xjoin` respectively denote an `xor_split` coordinator and an `xor_join` coordinator, respectively. The initial element of XOR(B_1, \dots, B_n) is `xsplit` and its final element is `xjoin`.*
 - *AND(B_1, \dots, B_n) denotes the block with vertex set $\bigcup_{i=0}^n V(B_i) \cup \{\text{asplit}, \text{ajoin}\}$ and edge set $\bigcup_{i=0}^n (E(B_i) \cup \{(\text{asplit}, b(B_i)), (f(B_i), \text{ajoin}) : 1 \leq i \leq n\})$ where `asplit` and `ajoin` denote an `and_split` and an `and_join` coordinator, respectively. The initial element of AND(B_1, \dots, B_n) is `asplit` and its final element is `ajoin`.*

By enclosing a process block as defined in Definition 1 along with a start and end task in a sequence block, we obtain a *structured process model*. The effects of executing the tasks contained in a business process model are described using annotations as shown in Definition 2.

Definition 2 (Annotated process). *Let P be a structured process and T be the set of tasks contained in P . An annotated process is a pair: (P, ann) , where ann is a function associating a consistent set of literals to each task in T : $\text{ann} : T \mapsto 2^{\mathcal{L}}$. The function ann is constrained to the consistent literals sets in $2^{\mathcal{L}}$.*

The update between the states of a process during its execution is inspired by the AGM² belief revision operator [2] and is used in the context of business

² The operator is named after the initials of the authors introducing it: Alchourr3n, G3rdenfors, and Makinson.

processes to define the transitions between states, which in turn are used to define the *traces*.

Definition 3 (State update). *Given two consistent sets of literals L_1 and L_2 , representing the process state and the annotation of a task being executed, the update of L_1 with L_2 , denoted by $L_1 \oplus L_2$ is a set of literals defined as follows:*

$$L_1 \oplus L_2 = L_1 \setminus \{\neg l \mid l \in L_2\} \cup L_2$$

Definition 4 (Executions and Traces). *Given a structured process model identified by a process block B , the set of its executions, written $\Sigma(B) = \{\epsilon \mid \epsilon \text{ is a sequence and is an execution of } B\}$. The function $\Sigma(B)$ is defined as follows:*

1. If B is a task t , then $\Sigma(B) = \{(\{t\}, \emptyset)\}$
2. if B is a composite block with sub-blocks B_1, \dots, B_n :
 - (a) If $B = \text{SEQ}(B_1, \dots, B_n)$, then $\Sigma(B) = \{\epsilon_1 +_{\mathcal{E}} \dots +_{\mathcal{E}} \epsilon_n \mid \epsilon_i \in \Sigma(B_i)\}$ and $+_{\mathcal{E}}$ the operator concatenating two executions.
 - (b) If $B = \text{XOR}(B_1, \dots, B_n)$, then $\Sigma(B) = \Sigma(B_1) \cup \dots \cup \Sigma(B_n)$
 - (c) If $B = \text{AND}(B_1, \dots, B_n)$, then $\Sigma(B) = \{\text{the union of the interleavings of: } \epsilon_1, \dots, \epsilon_n \mid \epsilon_i \in \Sigma(B_i)\}$

Given an annotated process (B, ann) and an execution $\epsilon = (t_1, \dots, t_n)$ such that $\epsilon \in \Sigma(B)$, a trace θ is a finite sequence of states: $(\sigma_1, \dots, \sigma_n)$. Each state of $\sigma_i \in \theta$ is a pair: (t_i, L_i) capturing what holds after the execution of a task t_i , expressed by a set of literals L_i . A set L_i is constructed as follows:

1. $L_0 = \emptyset$
2. $L_{i+1} = L_i \oplus \text{ann}(t_{i+1})$, for $1 \leq i < n$.

To denote the set of possible traces resulting from a process model (B, ann) , we use $\Theta(B, \text{ann})$.

Example 1. Annotated Process Model.

Figure 1 shows a structured process containing four tasks labelled t_1, t_2, t_3 and t_4 and their annotations. The process contains an AND block followed by a task and an XOR block nested within the AND block. The annotations indicate what has to hold after a task is executed. If t_1 is executed, then the literal a has to hold in that state of the process (Table 1).

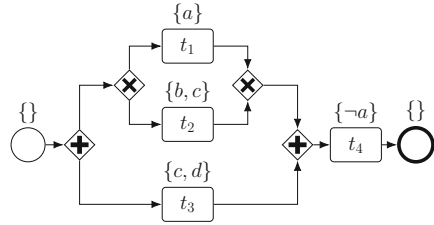


Fig. 1. An annotated process

Table 1. Executions and Traces of the annotated process in Fig. 1.

$\Sigma(B)$	$\Theta(B, \text{ann})$
(start, t_1, t_3, t_4 , end)	((start, \emptyset), ($t_1, \{a\}$), ($t_3, \{a, c, d\}$), ($t_4, \{\neg a, c, d\}$), (end, $\{\neg a, c, d\}$))
(start, t_2, t_3, t_4 end)	((start, \emptyset), ($t_2, \{b, c\}$), ($t_3, \{b, c, d\}$), ($t_4, \{\neg a, b, c, d\}$), (end, $\{\neg a, b, c, d\}$))
(start, t_3, t_1, t_4 end)	((start, \emptyset), ($t_3, \{c, d\}$), ($t_1, \{a, c, d\}$), ($t_4, \{\neg a, c, d\}$), (end, $\{\neg a, c, d\}$))
(start, t_3, t_2, t_4 end)	((start, \emptyset), ($t_3, \{c, d\}$), ($t_2, \{b, c, d\}$), ($t_4, \{\neg a, b, c, d\}$), (end, $\{\neg a, b, c, d\}$))

2.2 Regulatory Framework

We hereby introduce the regulatory framework through the use of obligations, which are the conditions that a process model needs to fulfil in order to be considered compliant. As such, we use a subset of Process Compliance Logic (PCL), introduced by Governatori and Rotolo [9]. Obligations can be either *global* or *local*. A global obligation is in force for the entire duration of an execution, while the in force interval of a local obligation is determined by its trigger and deadline conditions. Additionally, an obligation can be either an achievement or a maintenance obligation and it determines how such an obligation is fulfilled by an execution when in force.

Definition 5 (Global and Local Obligations).

Global A global obligation $\mathcal{O}^o\langle c \rangle$, where $o \in \{a, m\}$ represents whether the obligation is achievement or maintenance. The element c represents the condition of the obligation.

Local A local obligation $\mathcal{O}^o\langle c, t, d \rangle$, where $o \in \{a, m\}$ represents whether the obligation is achievement or maintenance. The element c represents the condition of the obligation, the element t the trigger, and the element d the deadline.

While the in force interval of a global obligation spans the entire duration of a trace, the in force interval of a local obligation is determined as a sub-trace where the first state of such a sub-trace satisfies the trigger, and the last state satisfies the deadline. Generally the trigger, deadline and condition of an obligation are defined as propositional formulae. Assuming the literals from \mathcal{L} contained in a state to be true, then a propositional formula can then be evaluated when that state implies it.

Evaluating the Obligations. Whether an in force obligation is fulfilled or violated is determined by the states of the trace included in the in force interval of the obligation. Moreover, whether an in force obligation is fulfilled depends on the type of an obligation, as described in Definition 6.

Definition 6 (Achievement and Maintenance Obligations).

Achievement *If this type of obligation is in force in an interval, then the fulfilment condition specified by the regulation must be satisfied by the execution in at least one point in the interval before the deadline is satisfied. If this is the case then the obligation in force is considered to be satisfied, otherwise it is violated.*

Maintenance *If this type of obligation is in force in an interval then the fulfilment condition must be satisfied continuously in all points of the interval until the deadline is satisfied. If this is the case then the obligation in force is satisfied, otherwise it is violated.*

Process Compliance. The procedure of proving whether a process is compliant with a regulatory framework can return different answers. A process is said to be *fully compliant* if every trace of the process is compliant with the regulatory framework. A process is *partially compliant* if there exists at least one trace that is compliant with the regulatory framework, and *not compliant* if there is no trace complying with the framework.

Definition 7 (Process Compliance). *Given a process (P, ann) and a regulatory framework composed by a set of obligations \mathbb{O} , the compliance of (P, ann) with respect to \mathbb{O} is determined as follows:*

- **Full compliance** $(P, \text{ann}) \models^F \mathbb{O}$ if and only if $\forall \theta \in \Theta(P, \text{ann}), \theta \vdash \mathbb{O}$.
- **Partial compliance** $(P, \text{ann}) \models^P \mathbb{O}$ if and only if $\exists \theta \in \Theta(P, \text{ann}), \theta \vdash \mathbb{O}$.
- **Not compliant** $(P, \text{ann}) \not\models \mathbb{O}$ if and only if $\neg \exists \theta \in \Theta(P, \text{ann}), \theta \vdash \mathbb{O}$.

Note that we consider a trace to be compliant with a regulatory framework if it satisfies every obligation belonging to the set composing the framework.

3 Existing Complexity Results

In this section, we introduce some of the existing computational complexity results related variants of the problem discussed in this paper. In particular, we use such results as a starting point from which we analyse the elements bringing the computational complexity of the problem into the class of **NP** problems.

3.1 Problem Acronyms

Before discussing the existing results, we first introduce a compact system to refer to different variants of the problem through acronyms representing their properties.

Definition 8 (Compact Acronyms). *The variants of the problem we refer to in this paper constantly aim to check regulatory compliance of a structured process model. The acronym system refers to the properties of the obligations being checked against the process model.*

- 1/n** *Whether the structured process is checked against a single (**1**) or a set of (**n**) obligations.*
- G/L** *Whether the in force interval of the obligations is **Global**, meaning that it spans the entirety of an execution of the model, or it is **Local**, meaning that the in force interval is determined by the trigger and deadline elements of an obligation.*
- /+** *Whether the elements of the obligation being checked on the structured process model are composed by literals (**-**) or propositional formulae (**+**).*

For instance, the variant **1G-** consists of verifying whether a structured process model is compliant with a single obligation, whose condition is expressed as a propositional literal and its in force interval spans the entire execution of the process model. Note that in the binary properties of the problems considered in this paper, the leftmost, i.e. **1** in **1/n** represents a subset of the right side. Intuitively, the case on the right side is at least as complex as the left case. For instance, a solution for a problem including a set of regulations requires also to solve the case where the set of regulations is composed of exactly 1 regulation.

3.2 Complexity Results

Table 2 outlines the existing complexity results. Note that, even if Colombo Tosatto et al. provide a reduction showing how the Hamiltonian Path problem can be reduced to **nL-**, the same reduction can also be used to show that the more expressive **nL+** is in **NP**-complete.

Table 2. Regulatory compliance complexity

Problem	Source	Compliance	Complexity Class
1G-	Colombo Tosatto et al. [23]	Full and partial	P
nL-	Colombo Tosatto et al. [22]	Partial	NP -complete
1L+	Colombo Tosatto et al. [22]	Full	co NP -complete
nL+	Colombo Tosatto et al. [22]	Partial	NP -complete

From the existing results, it can be noticed that the computational complexity of the variants of the problem jumps from **P** to **NP** as soon as the obligations become **Local**, and either the number of obligations is more than one or the elements describing them can be represented by propositional formulae.

Partial Compliance for $1L+$ is in NP-complete. Colombo Tosatto et al. [22] show that proving full compliance in $1L+$ is coNP-complete by providing a reduction from the *tautology* problem. We extend the existing result by showing that verifying whether a structured process model is *partially compliant* with a single local obligation whose elements are expressed using formulae is in NP-complete.

We show that proving partial compliance in $1L+$ is in NP-complete by exploiting the relation between *Tautology* and *Satisfiability* problems. The relation is that in a tautology problem, it is required that every interpretation satisfies the formula, while in a satisfiability problem it is required that there exists one. This relation is the same as the one between *full* and *partial* compliance, where the former checks whether every execution satisfies the regulatory framework, while the latter checks whether there is one. Thus, through the relation mentioned above, and by reusing some elements in the reduction used for the proof showing that proving full compliance of $1L+$ is in coNP-complete, we prove that checking whether $1L+$ is partial compliant is in NP-complete.

4 Further Complexity Results

Starting from the existing computational complexity results, we analyse their relations and provide some additional results and insights on the matter.

4.1 From Global to Local Obligations

From Table 2 it might seem that the feature of the problems increasing the computational complexity to the NP class is changing the regulations from being global to local, as this very feature is the common one between the variants $nL-$ and $1L+$. However, we argue that such a property is not the main cause of the computational complexity jump from P to NP.

Considering now the variant $nG-$, where each obligation contains a single literal in its condition, we can reduce it to $2G+$, which is a special case of $nG+$, where the set of regulatory requirements contain exactly 2 regulations. This is effectively the simplest case of $nG+$.

Reduction 1. Consider a set of regulations \mathbb{R} , where each regulation r_i contains a fulfilment condition c_i . Consider now the complementary subsets of \mathbb{R} : \mathbb{R}^a and \mathbb{R}^m , which respectively contain the obligations of type achievement and of type maintenance originally in \mathbb{R} .

The variant $nG-$ can be reduced to $nG+$ with an obligation set containing 2 obligations, by collapsing each obligation belonging to one of the subsets into a single one. The obligation resulting from the reduction contains a condition constructed as follows: $\bigwedge_{i=0}^n c_i$. This results in a problem containing two obligations, one achievement and one maintenance, which are in force for the entire execution of the process model and whose conditions are represented by conjunctions of literals.

Proof. The correctness of the reduction follows directly from Definition 6.

The reduction allows us to claim that from a computational complexity perspective $\mathbf{nG-} \leq \mathbf{nG+}$, when proving partial compliance of business process models. This result does not look too surprising initially as the second problem is clearly more complex than the first, due to stepping from $-$ to $+$, while seemingly maintaining the other properties. However, for the sake of precision, the reduction does reduce $\mathbf{nG-}$ to $\mathbf{2G+}$, as the reduced problem contains exactly two obligations, one for each of the possible types. In the next step of the analysis, we focus on the computational complexity of $\mathbf{1G+}$.

Partial Compliance of $\mathbf{1G+}$ is in NP-complete. We prove now that $\mathbf{1G+}$, aiming at proving whether a business process model is partially compliant with a single global obligation, whose condition is represented by a formula, is in NP-complete. We use a similar approach to the one used to prove the computational complexity class of $\mathbf{1L+}$. The difference between the problems is that the current one is limited to use a global obligation.

Despite the differences, we can still reduce *satisfiability* to $\mathbf{1G+}$, showing that it indeed belongs to the NP-complete complexity class. Considering $\mathbf{1G-}$ as a starting variant, this also shows that stepping from the simplest elements of the regulation being verified (whose elements are composed of literals) to the more complex variant (where formulae are allowed) brings it into NP. Thus, despite the first impression that moving from global to local obligations could have been the main cause of the computational complexity increase, we have shown here that the expressivity of the elements of the obligations indeed contributes to the computational complexity.

Partial Compliance of $\mathbf{nG-}$ is in NP-complete. We prove now that $\mathbf{nG-}$ (aiming at proving whether a business process model is partially compliant with a set of global obligations, whose condition is represented by a single literal) is in NP-complete.

When considering a satisfiability problem like 3-SAT, consisting of deciding whether a formula composed of a conjunction of clauses, where each clause is composed of a disjunction of at least 3 literals, is satisfiable. This problem is computationally hard, as illustrated by Karp [14]. Differently, if the number of disjoint literals is lower than 3, then the problem is solvable in polynomial time, as Krom illustrates [17].

Intuitively, when considering $\mathbf{nG-}$ and knowing that $\mathbf{1G-}$ is in P, then similarly to a satisfiability problem where multiple clauses are involved in the formula being verified mirrored in a compliance problem by the number of obligations, the computational complexity of proving partial compliance in $\mathbf{nG-}$ should also be in NP. We prove that $\mathbf{nG-}$ is in NP by reducing 3-SAT to it.

4.2 Computational Complexity Recap

Table 3 provides an overview of the computational complexity results introduced in this paper for the problem of proving partial compliance of business process

models with respect to a set of obligations with varying features. We can immediately see that the variants of the problem analysed in this paper have been proven to be **NP**-complete.

Table 3. Additional regulatory compliance complexity

Problem	Compliance	Complexity class
nG-	Partial	NP -complete
1G+	Partial	NP -complete
1L+	Partial	NP -complete

A variant having difficult features that are a superset of another, is at least as difficult as the other. For instance, the variant **nG+** is at least as difficult as both **nG-** or **1G+**. Furthermore, following the same reasoning, it can also be said that **nL+** is at least as difficult as **nG+**. Thus the following computational complexity relations are true: **nG-** ≤ **nG+** ≤ **nL+** and **1G+** ≤ **nG+** ≤ **nL+**.

Consider now Tables 2 and 3. Note that we are missing a computational complexity result for **nG+**. However, it can be seen in the computational complexity relations that **nG+** is at most as difficult as **1G+** and at most as difficult as **nL+**. Knowing that both **1G+** and **nL+** are in **NP**-complete, we can conclude that **nG+** is also in **NP**-complete.

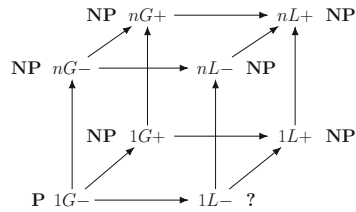


Fig. 2. Partial compliance complexity lattice.

Additionally, aggregating the existing results from Table 2 with the newly obtained results in Table 3, we obtain the computational complexity map for proving partial compliance shown in Fig. 2. Finally, note that Fig. 2 assigns a complexity class to each problem with the exception of **1L-**. We conclude this section by proposing a conjecture regarding the computational complexity of **1L-**.

Conjecture 1 (1L- is in P). We currently have no information about the computational complexity of **1L-**. That is, we cannot infer its belonging to a computational complexity class in a similar way as for **nG+**, as in this case the simpler variant (**1G-**) is in **P**.

Our conjecture is that the computational complexity of **1L-** is in **P**. We have proven that moving from - to +, or from 1 to n, definitely brings the complexity into **NP**. In general, solutions tackling such variants have to explore the entire set of possible executions in the worst case scenarios, which precludes efficient solutions. Despite moving from **G** to **L** seems to definitely increase the complexity, we strongly believe that it does not influence the computational complexity of the problem enough to move it into **NP**, and polynomial solutions are still possible.

5 Existing Approaches and Their Complexity

Compliance checking in itself is a generic term, and it is used in different ways in various approaches and the type of rules and regulations that are supported (along with the properties supported) differ for each approach. As a result, each approach may correspond to a different problem class as defined in this paper.

For example, approaches such as [3, 18] that are merely checking the control-flow of a process using temporal logic or other logic are typically $\mathbf{nG-}$. However, control-flow constraints may be conditional and apply therefore only locally, e.g. when using CTL* such as in [10]. In these cases, the enforcing part of the rules can be conditional and the problem belongs to the variant $\mathbf{nL-}$. Similarly, when variables (i.e. literals) are included in the approach in temporal logic [5], the problem can immediately be classified as the variant $\mathbf{nL-}$ as well. Approaches based on classical logic where propositional formulae are supported in addition to literals [7, 8, 11, 13, 16, 20] can be classified as $\mathbf{nL+}$.

As a result of the inherent computational complexity of such approaches, several methods describe a state-space reduction to limit state explosion in concurrent processes. However, these approaches either generate large amounts of overhead (e.g. [19]), or lose information on concurrency and local activity orders in concurrent branches due to linearisation (e.g., [4, 6, 12, 21]). The reduction approach described in [10] does reduce the state space while preserving all concurrency information, but provides only a practical optimisation and the theoretical complexity ($\mathbf{nL-}$) remains the same.

As such, each of the approaches discussed above is \mathbf{NP} -complete, hence they are bound to become infeasible, due to the current race towards digitalisation and automatisisation.

6 Conclusion

In this paper, we discussed and analysed the computational complexity of some variants of the problem of proving partial compliance of structured business process models. Despite lacking a computational complexity proof for $\mathbf{1L-}$, the results provided along with the classifications of some existing approaches tackling the problem, shows that many solutions belong to the \mathbf{NP} -complete computational complexity class.

While this was not unexpected, we claim that the current focus on optimisation is bound to lead to many complications in the future, as it is disregarding the worst case scenarios due to the limited size of current practical problems. Digitalisation and automatisisation is leading towards bigger and more complex automated processes, which current approaches would miserably fail to solve due to the, largely ignored, computational complexity of the problem. While optimising approaches tailored to current problems is still useful, dealing with the *elephant in the room* becomes more and more crucial, and will soon be impossible to ignore.

As future work to complete the computational complexity picture of the variants covered by the umbrella of $\mathbf{nL}+$, we plan to prove the computational complexity of the problem $\mathbf{1L-}$. Moreover, we also plan to switch the computational complexity analysis focus from the regulatory framework to the business process model, and studying how different variants of the business process model (i.e. including loops) influence the computational complexity of the problem.

Acknowledgments. This research is supported by the Science and Industry Endowment Fund.

A Proofs

A.1 Proving NP-completeness in General

Definition 9 (NP-complete). *A decision problem is NP-complete if it is in the set of NP problems and if every problem in NP is reducible to it in polynomial-time.*

To prove membership in **NP** of a variant of the problem of proving partial compliance, we show that a process is partially compliant with a set of obligations if and only if there is a certificate whose size is at most polynomial in terms of the length of the input (comprising the business process model and the set of obligations) with which we can check whether it fulfils the regulatory framework in polynomial time. As a certificate we use a trace of the model and we check whether it satisfies the regulatory framework.

NP Membership. The following algorithm allows to verify whether a trace of a business process model satisfies a set of obligations in polynomial time with respect to the length of the execution.

Algorithm 1 *Given a set of obligations \mathbb{O} and a trace $\theta = (\sigma_{start}, \sigma_1, \dots, \sigma_n, \sigma_{end})$ such that $\sigma_{start} = (\text{start}, L_0)$ and $\theta \in \Theta(P, \text{ann})$, the algorithm $A_1(\theta, \mathbb{O})$ is defined as follows:*

Certificate Checking Algorithm

```

Let  $\text{Ob} = \emptyset$  be the set of in force obligations
for each  $\sigma$  in  $\theta$  do
  for each  $\omega = \mathcal{O}^o(c, t, d)$  in  $\mathbb{O}$  do
    if  $\sigma \models t$  then
       $\text{Ob} = \text{Ob} \cup \omega$ 
    end if
  end for each
for each  $\omega = \mathcal{O}^o(c, t, d)$  in  $\text{Ob}$  do
  if  $\sigma = a$  then
    if  $\sigma \models c$  then
       $\text{Ob} = \text{Ob} \setminus \omega$ 
    else
      if  $\sigma \models d$  then
        return  $\theta \not\models \mathbb{O}$ 
      end if
    end if
  else
    if  $t = m$  then

```

```

        if  $\sigma \not\models c$  then
            return  $\theta \not\models \mathbb{O}$ 
        end if
        if  $\sigma \models d$  then
             $\text{Ob} = \text{Ob} \setminus \omega$ 
        end if
    end if
end if
end for each
end for each
return  $\theta \vdash \mathbb{O}$ ;
    
```

Notice that in case of **Global obligations**, whose in force interval spans the entirety of the execution, the set Ob is pre-loaded with each obligation of such type.

Algorithm 1 identifies whether a trace fulfils a set of obligations. If this is the case, then following from Definition 7 it is a sufficient condition to say that the process model is partially compliant with the regulatory framework defined by such obligations.

Complexity: the complexity of checking whether a trace is compliant with the set of obligations using Algorithm 1 is at most polynomial in time $\mathbf{O}(n \times o)$ where n is the number of tasks in the process and o is the number of obligations. Since checking whether the interpretation provided in a state satisfies a formula φ can be done in constant time, it does not affect the complexity of the algorithm. The above asymptotic time bound is at most polynomial in the length of the input (which includes the size of both the process and the set of obligations). Therefore, we can conclude that verifying Partial Compliance is indeed in **NP**.

Note that Algorithm 1 allows to check whether a trace is compliant with a regulatory framework where the class of the problem is **nL+**. With **nL+** being the most generic of the problem classes discussed in the paper, the same algorithm can be used to verify traces for every other problem. This allows to prove **NP**-completeness of various classes of the problems by just showing their **NP**-hardness, as their membership is taken care of by Algorithm 1. In the remainder of the appendix, we prove **NP** hardness of some of the problem classes by reducing existing **NP**-complete problems to the problem of proving partial regulatory compliance of a process.

A.2 NP Hardness of 1L+

We reduce the satisfiability problem to **1L+**.

Definition 10 (Satisfiability Problem). *The satisfiability problem is the problem of determining if there exists an interpretation that satisfies a given propositional formula. In other words, it asks whether there exists an interpretation of the propositions in such a way that the formula evaluates to **true**. If this is the case, the formula is satisfiable.*

Reduction. Let φ be a propositional formula for which we want to verify whether it is satisfiable or not, as described in Definition 10, and let L be the set of propositions contained in φ .

For each proposition belonging to L , we construct an XOR block containing two tasks, one labelled and containing in its annotation the positive proposition (i.e. l) and the other the negative counterpart (i.e. $\neg l$). Notice that we have not explicitly labeled the task, as they can be distinguished by their annotations. All the XOR blocks constructed from L are then included within a single AND block. This AND block is in turn followed by a task labelled “test” and containing a single literal in its annotation: l_{test} . The sequence containing the AND block and the task *test* is then enclosed within a start and an end, composing the annotated business process model (P, ann) , as graphically represented in Fig. 3.

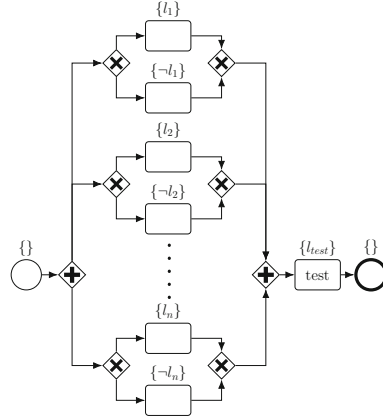


Fig. 3. Process for 1L+ reduction.

The set of obligations, to which the constructed business process has to be verified to be fully compliant with, is composed of a single obligation constructed as follows from the propositional formula φ : $\langle \mathcal{O}^a l_{test}, \varphi, \perp \rangle$. We claim that there exists a trace $\theta \in \Theta(P, \text{ann})$ such that $\theta \vdash \mathbb{O}$ if and only if φ is satisfiable.

Reduction Complexity: The process P and the obligation $\langle \mathcal{O}^a l_{test}, \varphi, \perp \rangle$ can be constructed in time proportional to $|L| + |\varphi|$ where $|\varphi|$ denotes the length of formula φ . Since $|L| \leq |\varphi|$ by construction, the time is at most polynomial in the length of the formula φ .

Correctness. Here we prove the soundness $((P, \text{ann}) \vdash^P \mathbb{O} \Rightarrow \varphi \text{ is satisfiable})$ and the completeness $(\varphi \text{ is satisfiable} \Rightarrow (P, \text{ann}) \vdash^P \mathbb{O})$ of our reduction.

Proof. Soundness: $(P, \text{ann}) \vdash^P \mathbb{O} \Rightarrow \varphi$ is satisfiable.

From the hypothesis and Definition 7, we know that there exists a trace of the business process model (P, ann) that fulfils the obligation in \mathbb{O} . Following from the construction of the reduction, we know that the only obligation belonging to \mathbb{O} is $\langle \mathcal{O}^a l_{test}, \varphi, \perp \rangle$.

From Definition 4 and the construction of the reduction we know that each trace of P contains the task l_{test} . Therefore, according to Definition 6, in order for the obligation $\langle \mathcal{O}^a l_{test}, \varphi, \perp \rangle$ to be fulfilled each trace contains a state following the one where l_{test} appears.

From Definition 4 and the construction of the reduction, in particular how (P, ann) is constructed, we have that in the only state following the one where l_{test} appears the first time, the set of literals associated to that state corresponds to an interpretation of the propositions contained in φ . Moreover, again from

the construction of the reduction, we know that in all the traces of (P, ann) , all the possible combinations of interpreting the propositions belonging to φ are considered.

Therefore, since the obligation $\langle \mathcal{O}^a l_{test}, \varphi, \perp \rangle$ is fulfilled by at least a trace and each trace corresponds to an interpretation, it follows from Definition 10 that φ is indeed satisfiable by the interpretation corresponding to the execution satisfying the obligation.

Proof. Completeness: φ is satisfiable $\Rightarrow (P, \text{ann}) \vdash^P \mathbb{O}$.

From the construction of the reduction we know that the condition of the only obligation contained in \mathbb{O} is constituted by φ . However, from the hypothesis we know that φ is satisfiable. Hence, according to Definition 6 such obligation is fulfilled by at least an interpretation of its proposition. Therefore, as by construction the process contains every possible interpretation for the formulae being analysed, and from Definition 7 it follows that if φ is satisfiable, then the compliance problem constructed using the reduction results in partial compliance, as one execution must fulfil the obligation.

A.3 NP Hardness of 1G+

We reduce the satisfiability problem to proving partial regulatory compliance for 1G+.

Reduction. Let φ be a propositional formula for which we want to verify whether it is satisfiable, and let L be the set of propositions in φ . For each proposition l belonging to L , we construct an XOR block containing two tasks, one labelled and containing in its annotation the positive proposition (i.e. l) and the other the negative counterpart (i.e. $\neg l$). All the XOR blocks constructed from L are then included within a single AND block.

The set of obligations, to which the constructed process has to be verified to be fully compliant with, is composed of a single obligation constructed as follows from the propositional formula φ : $\langle \mathcal{O}^a \varphi \rangle$. We claim that there exists a trace $\theta \in \Theta(P, \text{ann})$ such that $\theta \vdash \mathbb{O}$ if and only if φ is satisfiable.

Note that differently from the construction shown in Fig. 3, we do not need to include a *test* task to trigger the verification, as the obligation being global means that it is always in force and its condition is required to be achieved by the end of the execution.

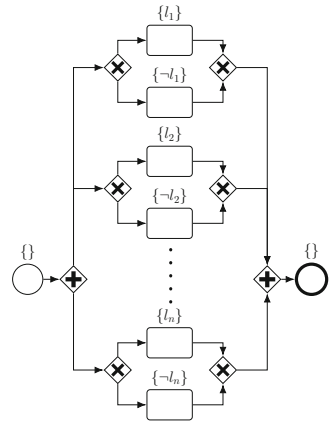


Fig. 4. Process for 1G+ reduction.

Reduction Complexity: As the reduction is similar to the one used for $\mathbf{1L+}$, the translation complexity is the same, which is in time polynomial with respect to the size of the formula φ .

Correctness. Here we prove the soundness ($(P, \text{ann}) \vdash^P \mathbb{O} \Rightarrow \varphi$ is satisfiable) and the completeness (φ is satisfiable $\Rightarrow (P, \text{ann}) \vdash^P \mathbb{O}$) of our reduction.

Proof. Soundness: $(P, \text{ann}) \vdash^P \mathbb{O} \Rightarrow \varphi$ is satisfiable.

From the hypothesis and Definition 7, we know that there exists a trace of the business process model (P, ann) that fulfils the obligation in \mathbb{O} . Following from the construction of the reduction we know that the only obligation belonging to \mathbb{O} is $\langle \mathcal{O}^a \varphi \rangle$.

From Definition 4 and the construction of the reduction we know that each trace of P contains a task related to the truth value for each literal appearing in φ . Moreover, again from the construction of the reduction, we know that in all the traces of (P, ann) all the possible combinations of interpreting the propositions belonging to φ are considered.

Therefore, since the obligation $\langle \mathcal{O}^a \varphi \rangle$ is fulfilled by at least a trace and each trace corresponds to an interpretation, it follows from Definition 10 that φ is indeed satisfiable by the interpretation corresponding to the execution satisfying the obligation.

Proof. Completeness: φ is satisfiable $\Rightarrow (P, \text{ann}) \vdash^P \mathbb{O}$.

From the construction of the reduction we know that the condition of the only obligation contained in \mathbb{O} is constituted by φ . However, from the hypothesis we know that φ is satisfiable. Hence, according to Definition 6, such obligation is fulfilled by at least an interpretation of its proposition. Therefore, as by construction the process contains every possible interpretation for the formulae being analysed, and from Definition 7, it follows that if φ is satisfiable, then the compliance problem constructed using the reduction results in partial compliance, as one execution must fulfil the obligation.

A.4 NP Hardness of nG-

We reduce the 3-SAT problem to proving partial regulatory compliance for nG-.

Definition 11 (3-SAT). *A propositional formula is in 3-CNF (Conjunctive Normal Form) if it is of the form $\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_k$ where each α_i is a disjunction of three or less literals. A propositional formula is in 3-SAT if it is in 3-CNF form and is also satisfiable.*

Reduction. Let 3-CNF : $\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_k$ be a propositional formula in 3-CNF, as shown in Fig. 5, for which we want to verify whether it is satisfiable or not, as described in Definition 11.

For each proposition l_i belonging to 3-CNF, we construct an XOR block containing two tasks, one labelled and containing in its annotation the positive proposition (i.e. a literal l_i) and the other the negative counterpart (i.e. its negation $\neg l_i$). All the XOR blocks constructed from L are then included within a single AND block. This

$$\underbrace{\{l_1 \vee l_2 \vee l_3\}}_{\alpha_1} \wedge \underbrace{\{\dots\}}_{\alpha_2} \wedge \dots$$

Fig. 5. 3-SAT formula

results in the same process model construction used in the computational complexity proof for **1G+** as graphically represented in Fig. 4.

For each clause α_i in 3-CNF, a global achievement obligation is created and its condition is set to the literal l_{α_i} , referring to the identifier of the clause. For each literal l_j in a clause α_i a *count as* rule is created as follows: $l_j \Rightarrow l_{\alpha_i}$.

Definition 12 (Count As Rule). Consider a *count as* rule: $\alpha \Rightarrow \beta$, where α and β are literals. If α is true in the process' state, then the process' state is considered to contain β .

Notice that the introduction of *count as* in such atomic version (meaning a literal to literal interpretation) does not increase the computational complexity of the problem **nG-**, as such interpretation can be computed in polynomial time while using Algorithm 1 to verify a trace in the reduced problem **nG-**.

Reduction Complexity: The construction of the process model in the reduction is the same as the one used for **1G+**. Differently, the regulatory framework is built by creating a set of obligations and count as rules. Still, constructing the regulatory framework is in time polynomial with respect to the size of 3-CNF. Therefore, the entire reduction is polynomial.

Correctness. Here we prove the soundness ($(P, \text{ann}) \vdash^P \mathbb{O} \Rightarrow$ 3-CNF is satisfiable) and the completeness (φ is satisfiable $\Rightarrow (P, \text{ann}) \vdash^P \mathbb{O}$) of our reduction.

Proof. Soundness: $(P, \text{ann}) \vdash^P \mathbb{O} \Rightarrow$ 3-CNF is satisfiable.

From the hypothesis and Definition 7 we know that there exists a trace of the business process model (P, ann) that fulfils the obligation in \mathbb{O} . Following from the construction of the reduction we know that each obligation in \mathbb{O} is in the form of $\langle O^a C_i \rangle$.

Moreover, from the construction of the reduction, we know that for each literal in a clause of the 3-CNF formula, a count as rule is created having the literal in its condition and the clause identifier in the conclusion. As each of the obligations in \mathbb{O} is satisfied and the literals used in the condition of the elements of the obligations cannot directly appear in the process' execution state, the condition of at least one of the count as rules associated to a clause must be true in a execution state of the process.

From Definition 4 and the construction of the reduction we know that each trace of P contains a task related to the truth value for each literal appearing in the 3-CNF formula. Moreover, again from the construction of the reduction,

we know that in all the traces of (P, ann) , all the possible combinations of interpreting the propositions belonging to the 3-CNF formula are considered. Additionally, we also know that: let \mathcal{I} be the final interpretation holding at the end of an execution and let \mathcal{I}_i be an intermediate interpretation, then the following is true $\mathcal{I} \models \mathcal{I}_{i+1} \models \mathcal{I}_i$ for any i .

Therefore, as the interpretation holding at \mathcal{I} is a possible interpretation of the literals in the 3-CNF formula, and the partial interpretation are not in contradiction with the final one, it follows from Definition 11 that the 3-CNF formula is indeed satisfiable by the interpretation corresponding to the execution satisfying the obligations since the obligations $\langle \mathcal{O}^a l_{\alpha_i} \rangle$ are all fulfilled by at least a trace and each trace corresponds to an interpretation, where at least one of the literals in each of the clauses is true.

Proof. Completeness: 3-CNF is satisfiable $\Rightarrow (P, \text{ann}) \vdash^P \bigcirc$

From the construction of the reduction we know that the obligations contained in \bigcirc are constituted by $\langle \mathcal{O}^a l_{\alpha_i} \rangle$. However, from the hypothesis we know that $\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_k$ is satisfiable. Hence, according to Definition 6, such obligation is fulfilled by at least an interpretation of its proposition. Moreover, from the construction of the reduction, we know that each literal in a clause forms the condition of a count as rule having the condition of one of the global obligations in its conclusion.

Therefore, as by construction the process contains every possible interpretation for the formulae being analysed, and from the hypothesis we know that $\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_k$ is satisfiable, then each clause α_i is satisfied, meaning that one of the disjunct propositions in α_i must be true in the interpretation.

From the construction, and given that there exists an interpretation satisfying the condition of at least a count as rule for each clause, then it follows that each obligation $\langle \mathcal{O}^a l_{\alpha_i} \rangle$ is satisfied. Thus, from Definition 7 it follows that if the 3-CNF is satisfiable then the compliance problem constructed using the reduction results in partial compliance, as one execution must fulfil the obligations.

References

1. Aalst, W.M.P.: Verification of workflow nets. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 407–426. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63139-9_48
2. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: partial meet contraction and revision functions. *J. Symbolic Logic* **50**(2), 510–530 (1985)
3. Awad, A., Decker, G., Weske, M.: Efficient compliance checking using BPMN-Q and temporal logic. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 326–341. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85758-7_24
4. Choi, Y., Zhao, J.L.: Decomposition-based verification of cyclic workflows. In: Peled, D.A., Tsay, Y.-K. (eds.) ATVA 2005. LNCS, vol. 3707, pp. 84–98. Springer, Heidelberg (2005). https://doi.org/10.1007/11562948_9

5. Elgammal, A., Turetken, O., van den Heuvel, W.-J., Papazoglou, M.: Formalizing and applying compliance patterns for business process compliance. *Softw. Syst. Model.* **15**(1), 119–146 (2016)
6. Feja, S., Speck, A., Pulvermüller, E.: Business process verification. In: *GI Jahrestagung*, pp. 4037–4051 (2009)
7. Ghose, A., Koliadis, G.: Auditing business process compliance. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) *ICSOC 2007*. LNCS, vol. 4749, pp. 169–180. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74974-5_14
8. Governatori, G.: The rigorous approach to process compliance. In: *2015 IEEE 19th International Enterprise Distributed Object Computing Workshop*, pp. 33–40. IEEE (2015)
9. Governatori, G., Rotolo, A.: Norm compliance in business process modeling. In: Dean, M., Hall, J., Rotolo, A., Tabet, S. (eds.) *RuleML 2010*. LNCS, vol. 6403, pp. 194–209. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16289-3_17
10. Groefsema, H., van Beest, N.R.T.P., Aiello, M.: A formal model for compliance verification of service compositions. *IEEE Trans. Serv. Comput.* **11**(3), 466–479 (2018)
11. Haarmann, S., Batoulis, K., Weske, M.: Compliance checking for decision-aware process models. In: Daniel, F., Sheng, Q.Z., Motahari, H. (eds.) *BPM 2018*. LNBP, vol. 342, pp. 494–506. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11641-5_39
12. Hoffmann, J., Weber, I., Governatori, G.: On compliance checking for clausal constraints in annotated process models. *Inf. Syst. Frontiers* **14**(2), 155–177 (2012)
13. Indiono, C., Fdhila, W., Rinderle-Ma, S.: Evolution of instance-spanning constraints in process aware information systems. In: Panetto, H., Debruyne, C., Proper, H., Ardagna, C., Roman, D., Meersman, R. (eds.) *OTM 2018*. LNCS, vol. 11229. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_17
14. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) *Complexity of Computer Computations*, pp. 85–103. Plenum Press, New York (1972)
15. Kiepuszewski, B., ter Hofstede, A.H.M., Bussler, C.J.: On structured workflow modelling. In: Wangler, B., Bergman, L. (eds.) *CAiSE 2000*. LNCS, vol. 1789, pp. 431–445. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45140-4_29
16. Knuplesch, D., Ly, L.T., Rinderle-Ma, S., Pfeifer, H., Dadam, P.: On enabling data-aware compliance checking of business process models. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) *ER 2010*. LNCS, vol. 6412, pp. 332–346. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16373-9_24
17. Krom, M.R.: The decision problem for a class of first-order formulas in which all disjunctions are binary. *Math. Logic Q.* **13**(1–2), 15–20 (1967)
18. Lu, R., Sadiq, S., Governatori, G.: Compliance aware business process design. In: ter Hofstede, A., Benatallah, B., Paik, H.-Y. (eds.) *BPM 2007*. LNCS, vol. 4928, pp. 120–131. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78238-4_14
19. Nakajima, S.: Verification of web service flows with model-checking techniques. In: *Proceedings of International Symposium on Cyber Worlds*, pp. 378–385 (2002)
20. Sadiq, S., Governatori, G., Namiri, K.: Modeling control objectives for business process compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 149–164. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_12

21. Sadiq, S., Orlowska, M.E., Sadiq, W.: Specification and validation of process constraints for flexible workflows. *Inf. Syst.* **30**(5), 349–378 (2005)
22. Colombo Tosatto, S., Governatori, G., Kelsen, P.: Business process regulatory compliance is hard. *IEEE Trans. Serv. Comput.* **8**(6), 958–970 (2015)
23. Colombo Tosatto, S., Kelsen, P., Ma, Q., El Kharbili, M., Governatori, G., van der Torre, L.W.N.: Algorithms for tractable compliance problems. *Frontiers Comput. Sci.* **9**(1), 55–74 (2015)
24. van der Aalst, W.M.P.: The application of petri nets to workflow management. *J. Circuits Syst. Comput.* **8**(1), 21–66 (1998)



Modeling and Reasoning over Declarative Data-Aware Processes with Object-Centric Behavioral Constraints

Alessandro Artale¹, Alisa Kovtunova¹, Marco Montali¹(✉),
and Wil M. P. van der Aalst²

¹ Free University of Bozen-Bolzano, Bolzano, Italy
{artale,kovtunova,montali}@inf.unibz.it

² Process and Data Science, RWTH Aachen University, Aachen, Germany
wvdaalst@pads.rwth-aachen.de

Abstract. Existing process modeling notations ranging from Petri nets to BPMN have difficulties capturing the data manipulated by processes. Process models often focus on the control flow, lacking an explicit, conceptually well-founded integration with real data models, such as ER diagrams or UML class diagrams. To overcome this limitation, *Object-Centric Behavioral Constraints* (OCBC) models were recently proposed as a new notation that combines full-fledged data models with control-flow constraints inspired by declarative process modeling notations such as DECLARE and DCR Graphs. We propose a formalization of the OCBC model using temporal description logics. The obtained formalization allows us to lift all reasoning services defined for constraint-based process modeling notations without data, to the much more sophisticated scenario of OCBC. Furthermore, we show how reasoning over OCBC models can be reformulated into decidable, standard reasoning tasks over the corresponding temporal description logic knowledge base.

1 Introduction

Despite the plethora of notations available to model business processes, process modelers struggle to capture real-life processes using mainstream notations such as Business Process Model and Notation (BPMN), Event-driven Process Chains (EPC), and UML activity diagrams. All such notations require the simplifying assumption that each process model focuses on a single, explicitly defined *case* notion (also referred to as *process instance*). The discrepancy between the single case view and reality becomes evident when using process mining techniques to reconstruct processes based on the available data [2]. Process mining starts from the available data and, unless one is using a Business Process Management (BPM) or Workflow Management (WFM) system for process execution, explicit case information is typically missing. Process-centric diagrams using BPMN, EPCs, or UML describe the life-cycle of individual cases. When formal languages like Petri nets, automata, and process algebras are used to describe business

processes, they tend to model cases in isolation, and the data perspective is secondary or missing completely. Languages like BPMN allow modelers to attach data to processes, but without the possibility to express complex constraints over such data (e.g., cardinality constraints, is-a links, disjointness, covering, etc. as in ER/UML/ORM data models). Mainstream business process modeling notations describe the *lifecycle of one type of process instance* at a time missing the opportunity to capture the co-evolution of multiple, interacting instances. In particular, complex constraints over data attached to processes *must* influence the behavior of the process itself—e.g., consider the management of different orders, where the evolution of one order impacts on the possible evolutions of the related orders.

Object-Centric Behavioral Constraint (OCBC) [3,21,22] models have been proposed as a modeling language that combines ideas from declarative, constraint-based languages like *DECLARE* [1], and from data modeling languages. OCBC allows to: (i) describe the temporal interaction between activities in a given process and to attach (structured) data to processes in a *unified framework*; (ii) model the *interactions between multiple process instances*, specifically when there is a *one-to-many* or *many-to-many* relationship between them. Figure 1 illustrates the way in which OCBC models tackle the above two issues. *Register Email* and *Send Invite* are two activities related to object classes *Person* and *Meeting*, respectively. A meeting is organized by many persons, each of which can in turn organize many meetings. The double-headed arrow connecting *Register Email* and *Send Invite* expresses the constraint that an invitation for a meeting can be sent only if at least one organizer of *that* meeting has previously registered her e-mail. Assuming that the object targeted by each activity is indeed a case for that activity, this simple example already contains two distinct case notions (*Person* and *Meeting*) that are intertwined. In conventional notations, this can only be modeled from the viewpoint of one of the two instances: the registration process of a person or the invitation process for a meeting. Taking the latter viewpoint using conventional notations such as BPMN would require to explicitly introduce a loop to handle the registration of one or more persons organizing a meeting. However, this is incorrect because one registration may be followed by many meetings. One-to-many and many-to-many relationships lead to convergence and divergence problems that cannot be handled in notations describing isolated cases.

OCBC models are related to artifact- and data-centric approaches [12,16,19] aiming to integrate data and processes. However, this is not done in a single diagram representing different types of process instances and their interactions. In addition, these approaches usually assume complete knowledge over the data, and require to fully spell out data updates when specifying the activities [14,26]. The few proposals dealing with artifact-centric models with incomplete knowledge [10] do not come with a fully

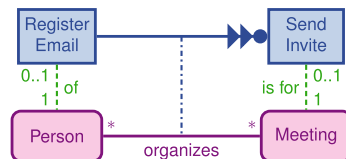


Fig. 1. An OCBC constraint

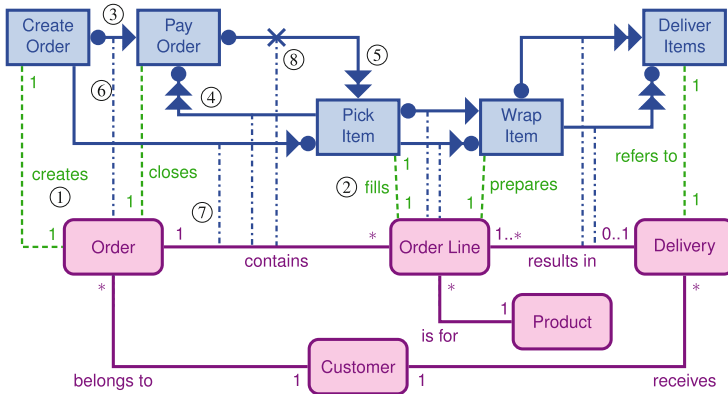


Fig. 2. Example of an OCBC model

integrated, declarative semantics as done here, but follow instead the Levesque functional approach [20] to separate the evolution of the system from the inspection of (incomplete) knowledge in each state.

This paper provides a complete characterization of the formal semantics of the OCBC approach, unambiguously defining the logical *meaning* of OCBC constraints. We provide a visual and textual syntax for OCBC, then defining the semantics of the different modeling constructs in terms of *temporal description logics*, i.e., a temporal extension of (fragments of) the well-known OWL language. The obtained formalization, in turn, allows us to lift all reasoning services defined for constraint-based process modeling notations without data, to the much more sophisticated setting of OCBC. In particular, we show how reasoning over OCBC models can be reformulated into decidable, standard reasoning tasks over the corresponding temporal description logic knowledge base, giving solid foundations to the boundaries of decidability and complexity of reasoning over processes and their manipulated data.

The paper is organized as follows. We present a running example in Sect. 2. Section 3 briefly illustrates the temporal DL that will be used to encode and reason over OCBC models. Section 4 shows the syntax for OCBC models and their semantics via the temporal DL encoding. Reasoning and verification tasks for OCBC models are tackled in Sect. 5. We present our remarks and future work in Sect. 6.

2 Running Example

The driving assumption underlying our proposal is that processes are modeled as a mirror of their manipulated data. Such data is structured according to complex data modeling constraints (see the lower part of Fig. 2). Data can be attached to activities (see the dotted lines of Fig. 2) and ad-hoc *co-reference* constraints can be expressed on those manipulated data (see the dash-dotted lines of Fig. 2) describing how activities can share/reuse the same data objects.

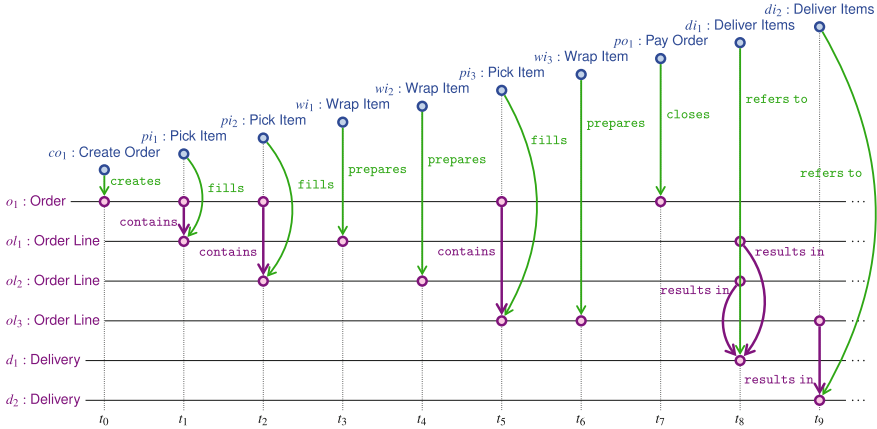


Fig. 3. Trace fragment for the OCBC model in Fig. 2

Example 1. Figure 2 shows an OCBC model for a process composed by five activities (CreateOrder, PickItem, WrapItem, PayOrder and DeliverItems) and five object classes in the data model (Order, OrderLine, Delivery, Product and Customer). The top part describes the temporal ordering of activities and the bottom part how objects relevant for the process execution are structured (read the lower part as a standard UML class diagram). The middle layer (dotted lines) relates activities and data. We now informally describe the constructs highlighted in Fig. 2. ① There is a one-to-one correspondence between a CreateOrder activity and an Order, i.e., the execution of a CreateOrder activity creates a unique Order and, vice-versa, due to the 1 on the CreateOrder side, each Order has been generated by a single execution of a CreateOrder activity. ② Every execution of the PickItem activity refers to a unique OrderLine and each OrderLine has been generated by an execution of a PickItem activity (and not by a WrapItem activity). ③ Each CreateOrder activity is followed by exactly one (single arrow) PayOrder activity related to the same order. ④ Each PayOrder activity is preceded by possibly many (double arrow) PickItem activities. ⑤ Whenever we execute PayOrder we will never execute PickItem on the same paid order. ⑥ The dash-dotted line denotes a co-reference constraint over an object class, imposes that when the CreateOrder creates an order instance, that order instance will eventually be paid by executing a PayOrder activity. ⑦ The dash-dotted line is, in this case, a co-reference constraint now over a relationship which imposes that when we fill an order line it must have been contained in exactly one order created by executing a CreateOrder activity. Since an order line instance could not exist at the same time we create an order instance and relationships are instantiated by co-existing objects, the UML model correctly specifies that, at each point in time, each order participates zero or more times in the contains relation. On the other hand, the co-reference constraint together with the mandatory cardinalities constraints and

the temporal constraints between `CreateOrder`, `PayOrder` and `PickItem` imply the *eventual* existence of *at least one* order line contained in any given order. ⑧ The dash-dotted line starting with a \times denotes a *negative co-reference constraint* that forbids filling with further order lines an order that has been closed by a `PayOrder` activity.

A possible execution of an OCBC process, called in the following *trace fragment*, records at once events, with their execution time, and the objects they operate on. In addition, it also captures facts that are known to hold over such objects in a given timestamp, in particular, the classes to which objects belong to at that time, as well as how objects are related to each other. In addition, the trace fragment captures, as customary in a standard first-order logic setting, *incomplete knowledge* about a process execution, and OCBC constraints are hence interpreted under the *open-world semantics*. This means that a trace fragment conforms to an OCBC model if it can be extended towards a full trace that satisfies all the constraints contained therein. A trace fragment conforming to the OCBC model of Fig. 2 is depicted in Fig. 3 and shown in the following first-order logic notation (but also as a DL ABox after a small transformation). We abbreviate activity names with their initials. Instances of activities, classes and relationships are timestamped denoting the execution time of the activity, and the time point when the described fact holds (timestamps respect the time ordering starting from t_0).

```
CO( $co_1, t_0$ ), PI( $pi_1, t_1$ ), PI( $pi_2, t_2$ ), WI( $wi_1, t_3$ ), WI( $wi_2, t_4$ ), PI( $pi_3, t_5$ ), WI( $wi_3, t_6$ ), PO( $po_1, t_7$ ),
DI( $di_1, t_8$ ), DI( $di_2, t_9$ ), creates( $co_1, o_1, t_0$ ), fills( $pi_1, ol_1, t_1$ ), contains( $o_1, ol_1, t_1$ ), fills( $pi_2, ol_2, t_2$ ),
contains( $o_1, ol_2, t_2$ ), prepares( $wi_1, ol_1, t_3$ ), prepares( $wi_2, ol_2, t_4$ ), fills( $pi_3, ol_3, t_5$ ),
contains( $o_1, ol_3, t_5$ ), prepares( $wi_3, ol_3, t_6$ ), closes( $po_1, o_1, t_7$ ), refers to( $di_1, d_1, t_8$ ),
results in( $ol_1, d_1, t_8$ ), results in( $ol_2, d_1, t_8$ ), refers to( $di_2, d_2, t_9$ ), results in( $ol_3, d_2, t_9$ ),
```

The process described in the example cannot be modeled using conventional process modeling languages, because (a) three different types of instances (of activities, classes and also relationships instances) are intertwined in a uniform framework so that no further coding or annotations are needed, and (b) cardinality and structural constraints in the object class model influence the allowed behavior of activities, and vice-versa. Take, e.g., the fact that in the example we have three different `OrderLine` instances (ol_1, ol_2, ol_3), then, together with the co-reference constraints on `OrderLine`, we implicitly enforce the occurrence of three different `PickItem` and `WrapItem` activities.

3 A Gentle Introduction to Temporal DLs

Since description logics (DLs) are able to capture data models [4, 11, 17] and are the logical formalism underpinning ontologies expressed in the standard Web Ontology Language OWL (www.w3.org/2007/OWL), while the linear temporal logic (LTL) is able to formalize the temporal interweaving of the activities in a process [1], we propose here to use temporal description logics based on $T_{US}ALCCQI$ and its fragments [8, 18, 27] to formally describe the semantics of

OCBC models and to capture in a uniform formalism both the processes and their attached data.

$T_{\mathcal{US}}\mathcal{ALCQI}$ is one of the most expressive and still decidable temporal description logics. The language alphabet contains *object names* a_0, a_1, \dots , *concept names* A_0, A_1, \dots and *role names* P_0, P_1, \dots . Then, *roles* R and *concepts* C are given by the following grammar:

$$R ::= P_i \mid R^- \quad C ::= \top \mid A_i \mid (\geq q R C) \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \mathcal{U} C_2 \mid C_1 \mathcal{S} C_2$$

where R^- denotes the *inverse* of the role R (obtained by reversing the relation R) and q is a positive integer. We use the standard abbreviations: $C_1 \sqcup C_2 = \neg(\neg C_1 \sqcap \neg C_2)$, $\perp = \neg\top$, $\exists R = (\geq 1 R \top)$, $\exists R.C = (\geq 1 R C)$, $(\leq q R C) = \neg(\geq (q+1) R C)$. Furthermore, all the temporal operators used in LTL can be expressed via \mathcal{S} ‘since’ and \mathcal{U} ‘until’ [18]. Operators \diamond_F and \diamond_P (‘sometime in the future/past’) can be expressed as $\diamond_F C = \top \mathcal{U} C$ and $\diamond_P C = \top \mathcal{S} C$; operators \square_F (‘always in the future’) and \square_P (‘always in the past’) are defined as dual to \diamond_F and \diamond_P , i.e., $\square_F C = \neg\diamond_F\neg C$ and $\square_P C = \neg\diamond_P\neg C$. The non-strict operators (including the current evaluation time), denoted as \diamond_P^+ and \diamond_F^+ , can be captured as $\diamond_P^+ C = C \sqcap \diamond_P C$ and $\diamond_F^+ C = C \sqcap \diamond_F C$ (similarly, \square_P^+ and \square_F^+ are defined as the dual operators of \diamond_P^+ and \diamond_F^+ , respectively). The ‘always’ operator \boxtimes can be expressed as $\boxtimes C = \square_F \square_P C$, while the dual ‘sometime’ is defined as $\diamond C = \neg\boxtimes\neg C$. Finally, the temporal operators \circ_F (‘next time’) and \circ_P (‘previous time’) can be defined as $\circ_F C = \perp \mathcal{U} C$ and $\circ_P C = \perp \mathcal{S} C$.

A $T_{\mathcal{US}}\mathcal{ALCQI}$ TBox \mathcal{T} is a finite set of *concept* and *role inclusion* axioms of the form $C_1 \sqsubseteq C_2$ and $R_1 \sqsubseteq R_2$, respectively. An ABox, \mathcal{A} , consists of assertions of the form $\circ^n A_k(a_i)$, $\circ^n P_k(a_i, a_j)$, where A_k is a concept name, P_k a role name, a_i, a_j object names and, for $n \in \mathbb{Z}$,

$$\circ^n = \underbrace{\circ_F \cdots \circ_F}_{n \text{ times}}, \text{ if } n \geq 0, \text{ and } \circ^n = \underbrace{\circ_P \cdots \circ_P}_{-n \text{ times}}, \text{ if } n < 0.$$

Taken together, the TBox \mathcal{T} and ABox \mathcal{A} form the *knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. In this paper, OCBC models will be encoded using TBoxes (see Sect. 4.4), while single process executions (i.e., trace fragments as shown in Example 1) are encoded as ABoxes (e.g., $\mathbf{CO}(co_1, t_0)$ is encoded as $\circ^{t_0} \mathbf{CO}(co_1)$).

A *temporal interpretation* is a structure of the form $\mathcal{I} = ((\mathbb{Z}, <), \Delta^{\mathcal{I}}, \{\cdot^{\mathcal{I}} \mid n \in \mathbb{Z}\})$, where $(\mathbb{Z}, <)$ is the linear model of time, $\Delta^{\mathcal{I}}$ is a non-empty interpretation domain and $\mathcal{I}(n)$ gives a standard DL interpretation for each time instant $n \in \mathbb{Z}$: $\mathcal{I}(n) = (\Delta^{\mathcal{I}}, a_0^{\mathcal{I}(n)}, A_0^{\mathcal{I}(n)}, \dots, P_0^{\mathcal{I}(n)}, \dots)$, assigning to each concept name A_i a unary predicate $A_i^{\mathcal{I}(n)} \subseteq \Delta^{\mathcal{I}}$ and to each role name P_i a binary relation $P_i^{\mathcal{I}(n)} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. We assume that the domain $\Delta^{\mathcal{I}}$ and the interpretations $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ of object names are the same for all $n \in \mathbb{Z}$, i.e., we adopt the *constant domain assumption* and *rigid designators* (consult [18] for more details on these assumptions). At each time instant $n \in \mathbb{Z}$, role and concept constructs are interpreted as follows

$$\begin{aligned}
 (R^-)^{\mathcal{I}(n)} &= \{(y, x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}(n)}\}, \\
 (\geq q R C)^{\mathcal{I}(n)} &= \{x \in \Delta^{\mathcal{I}} \mid \#\{y \in C^{\mathcal{I}(n)} \mid (x, y) \in R^{\mathcal{I}(n)}\} \geq q\}, \\
 (\neg C)^{\mathcal{I}(n)} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}(n)}, \quad \top^{\mathcal{I}} = \Delta^{\mathcal{I}}, \quad (C_1 \sqcap C_2)^{\mathcal{I}(n)} = C_1^{\mathcal{I}(n)} \cap C_2^{\mathcal{I}(n)}, \\
 (C_1 \mathcal{U} C_2)^{\mathcal{I}(n)} &= \bigcup_{k>n} (C_2^{\mathcal{I}(k)} \cap \bigcap_{n<m<k} C_1^{\mathcal{I}(m)}), \\
 (C_1 \mathcal{S} C_2)^{\mathcal{I}(n)} &= \bigcup_{k<n} (C_2^{\mathcal{I}(k)} \cap \bigcap_{n>m>k} C_1^{\mathcal{I}(m)}),
 \end{aligned}$$

where $\#X$ denotes the cardinality of X . Thus, for example, $x \in (C_1 \mathcal{U} C_2)^{\mathcal{I}(n)}$ iff there is a moment $k > n$ such that $x \in C_2^{\mathcal{I}(k)}$ and $x \in C_1^{\mathcal{I}(m)}$, for all moments m between n and k . Note that the operators \mathcal{S} and \mathcal{U} are ‘strict’ in the sense that their semantics does not include the current moment of time.

Concept and role inclusion axioms (TBox) are interpreted in \mathcal{I} *globally*:

$$\begin{aligned}
 \mathcal{I} \models C_1 \sqsubseteq C_2 &\text{ iff } C_1^{\mathcal{I}(n)} \subseteq C_2^{\mathcal{I}(n)} \text{ for all } n \in \mathbb{Z}, \\
 \mathcal{I} \models R_1 \sqsubseteq R_2 &\text{ iff } R_1^{\mathcal{I}(n)} \subseteq R_2^{\mathcal{I}(n)} \text{ for all } n \in \mathbb{Z}.
 \end{aligned}$$

ABox assertions are interpreted relatively to the *initial moment*, 0:

$$\begin{aligned}
 \mathcal{I} \models \bigcirc^n A_k(a_i) &\text{ iff } a_i^{\mathcal{I}} \in A_k^{\mathcal{I}(n)}, \\
 \mathcal{I} \models \bigcirc^n P_k(a_i, a_j) &\text{ iff } (a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in P_k^{\mathcal{I}(n)}.
 \end{aligned}$$

We call \mathcal{I} a *model* of a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and write $\mathcal{I} \models \mathcal{K}$ if \mathcal{I} satisfies all inclusions in \mathcal{T} and all assertions in \mathcal{A} . A KB \mathcal{K} is *satisfiable* if it has a model. A concept C (role R) is *satisfiable* with respect to \mathcal{K} if there are a model \mathcal{I} of \mathcal{K} and $n \in \mathbb{Z}$ such that $C^{\mathcal{I}(n)} \neq \emptyset$ (respectively, $R^{\mathcal{I}(n)} \neq \emptyset$). It is readily seen that the concept and role satisfiability problems are equivalent to KB satisfiability.

Reasoning in $T_{US}ALCQI$ w.r.t. to a KB is a problem which has been proven to be ExpTime-complete [18, 27]. To achieve better complexity results fragments of $ALCQI$ must be considered. Nice results have been gained when temporalizing *DL-Lite* logics [6, 13]—see, e.g., the temporal *DL-Lite* called $T_{US}DL-Lite_{bool}^{\mathcal{H}\mathcal{N}}$ where reasoning has the same complexity of LTL reasoning, i.e., PSpace-complete [8].

4 The OCBC Model

We now present the syntax and graphical appearance of OCBC models, together with their formal semantics. The original proposal of the OCBC model is the way activities and data are related. In particular, an OCBC model captures, at once: (i) *Data dependencies*, represented using standard data modeling constructs, i.e., *classes*, *relationships* and *constraints* between them; (ii) *Activities*, accounting for units of work within a process; (iii) *Mutual relationships between activities and classes*, linking the execution of activities in a given process with the data

objects they manipulate; (iv) *Temporal constraints* between activities; (v) *Co-reference constraints* that enforce the application of temporal constraints, and in particular limit their application to those activities that indirectly co-refer thanks to the objects and relationships they point to.

4.1 The Data Model – ClaM

Data used by the activities of an OCBC model is structured according to a standard modeling language, i.e., ER/UML/ORM. While \mathcal{ALCQI} is able to fully capture the semantics of such data models (see [4, 11, 17] and references therein) in the following, just for the sake of simplicity and lack of space, we present only a subset of the complete set of modeling constructs allowed in those standard data modeling languages and denote such set of modeling constructs as the ClaM data model (which stands for *CLAss data Model*). In particular, the following syntax limits ClaM to capture object classes that can be organized along ISA hierarchies (with possibly disjoint sub-classes and covering constraints), binary relationships between object classes and cardinalities expressing participation constraints of object classes in relationships.

Definition 1 (ClaM Syntax). *A conceptual schema Σ in the Class Model, ClaM, is a tuple $\Sigma = (\mathcal{U}_C, \mathcal{U}_R, \tau, \#_{dom}, \#_{ran}, \text{ISA}, \text{DISJ}, \text{COV})$, where:*

- \mathcal{U}_C is the universe of object classes. We denote object classes as O_1, O_2, \dots ;
- \mathcal{U}_R is the universe of binary relationships among object classes. We denote relationships as R_1, R_2, \dots ;
- $\tau : \mathcal{U}_R \rightarrow \mathcal{U}_C \times \mathcal{U}_C$ is a total function associating a signature to each binary relationship. If $\tau(R) = (O_1, O_2)$ then O_1 is the range and O_2 the domain of the relationship;
- $\#_{dom} : \mathcal{U}_R \times \mathcal{U}_C \rightarrow \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$ is a partial function defining cardinality constraints on the domain of a relationship. $\#_{dom}(R, O)$ is defined only if $\tau(R) = (O, O_1)$;
- $\#_{ran} : \mathcal{U}_R \times \mathcal{U}_C \rightarrow \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$ is a partial function defining cardinality constraints on the range of a relationship. $\#_{ran}(R, O)$ is defined only if $\tau(R) = (O_1, O)$;
- $\text{ISA} \subseteq \mathcal{U}_C \times \mathcal{U}_C$ is a binary relation defining the super-class and sub-class hierarchy on object classes. If $\text{ISA}(C_1, C_2)$ then C_1 is said to be a sub-class of C_2 while C_2 is said to be a super-class of C_1 ;
- $\text{DISJ} \subseteq 2^{\mathcal{U}_C} \times \mathcal{U}_C$ is a binary relation defining the set of disjoint sub-classes in an ISA hierarchy;
- $\text{COV} \subseteq 2^{\mathcal{U}_C} \times \mathcal{U}_C$ is a binary relation defining the set of sub-classes covering the super-class in an ISA hierarchy.

As for the full-fledged syntax of ER/UML/ORM, their formal set-theoretic semantics, and their translation as \mathcal{ALCQI} KBs we refer to [4, 11, 17]. Concerning the semantics of the ClaM constructs, cardinality constraints are interpreted as the number of times each instance of the involved class participates in the given relationship, ISA is interpreted as sub-setting, DISJ and COV are interpreted

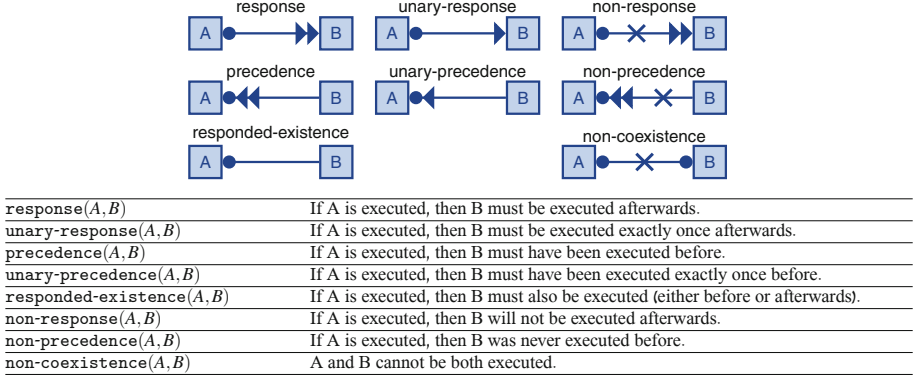


Fig. 4. Types of temporal constraints between activities and their intuitive semantics

in the obvious way using disjointness/union between classes, relationships are interpreted as binary predicates, while the relationship signature acts as a typing for its arguments.

Example 2. The lower part of the OCBC model shown in Fig. 2 captures the data model as a ClaM diagram with:

$$\mathcal{U}_C = \{\text{Order}, \text{OrderLine}, \text{Product}, \text{Customer}, \text{Delivery}\};$$

$$\mathcal{U}_R = \{\text{contains}, \text{belongs to}, \text{is for}, \text{results in}, \text{receives}\};$$

$$\tau(\text{contains}) = (\text{Order}, \text{OrderLine}), \dots$$

$$\#_{\text{dom}}(\text{contains}, \text{Order}) = (0, \infty); \#_{\text{ran}}(\text{contains}, \text{OrderLine}) = (1, 1); \dots$$

Cardinalities are shown in the diagram following the UML reading.

4.2 Temporal Constraints over Activities

Taking inspiration from the DECLARE patterns [1], we present here the temporal constraints between (pairs of) activities that can be expressed in OCBC. Figure 4 graphically renders such constraints together with their intuitive meaning. In the following we present their syntax.

Definition 2 (Temporal constraints). *Let*

- \mathcal{U}_A be the universe of activities, denoted with capital letters A_1, A_2, \dots ;
- \mathcal{U}_{TC} be the universe of temporal constraints, i.e., $\mathcal{U}_{TC} = \{\text{response}, \text{unary-response}, \text{precedence}, \text{unary-precedence}, \text{responded-existence}, \text{non-response}, \text{non-precedence}, \text{non-coexistence}\}$, where each $tc \in \mathcal{U}_{TC}$ is a binary relation over activities, i.e., $tc \subseteq \mathcal{U}_A \times \mathcal{U}_A$.

The set of temporal constraints in a given OCBC model is denoted as Σ_{TC} and is conceived as a set of elements of the form $tc(A_1, A_2)$, where $tc \in \mathcal{U}_{TC}$ and $A_1, A_2 \in \mathcal{U}_A$.

Remark 1. We observe that the *non-precedence* constraint is syntactic sugar, as it can be emulated using *non-response*: $\text{non-precedence}(A, B) \equiv \text{non-response}(B, A)$. Thus, in the following we will not consider it anymore. When defining later on the OCBC model we will consider the set Σ_{TC}^+ of *positive* constraints containing *response*, *unary-response*, *precedence*, *unary-precedence*, and *responded-existence*, and the set Σ_{TC}^- of *negative* constraints containing *non-response* and *non-coexistence*.

4.3 Syntax of OCBC Models

We are now ready to define the OCBC model starting from data models and temporal constraints as respectively defined in Sects. 4.1 and 4.2.

Definition 3 (OCBC syntax). *An OCBC model, \mathcal{M} , is a tuple:*

($\text{ClaM}, \Sigma_{TC}, \mathcal{U}_A, \mathcal{U}_{RAC}, \tau_{RAC}, \#_{act}, \#_{obj}, \text{cref}, \text{neg-cref}$), where:

- *ClaM is a data model as in Definition 1, and Σ_{TC} a set of temporal constraints as in Definition 2;*
- *\mathcal{U}_A is the universe of activities;*
- *\mathcal{U}_{RAC} is the universe of activity-object relationships being a set of binary relationships;*
- *$\tau_{RAC}: \mathcal{U}_{RAC} \rightarrow \mathcal{U}_A \times \mathcal{U}_C$ is a total function associating a signature to each activity-object relationship. If $\tau_{RAC}(R) = (A, O)$ then $A \in \mathcal{U}_A$ and $O \in \mathcal{U}_C$;*
- *$\#_{act}: \mathcal{U}_{RAC} \times \mathcal{U}_A \rightarrow \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$ is a partial function defining cardinality constraints on the participation of activities in activity-object relationships. $\#_{act}(R, A)$ is defined only if $\tau_{RAC}(R) = (A, O)$;*
- *$\#_{obj}: \mathcal{U}_{RAC} \times \mathcal{U}_C \rightarrow \{1\}$ is a partial function denoting the activity that generated a given object in O . $\#_{obj}(R, O)$ is defined only if $\tau_{RAC}(R) = (A, O)$;*
- *cref is the partial function of co-reference constraints s.t.*

$$\text{cref}: \Sigma_{TC}^+ \times \mathcal{U}_{RAC} \times \mathcal{U}_{RAC} \rightarrow \mathcal{U}_C \cup \mathcal{U}_R;$$
- *neg-cref is the partial function of negative co-reference constraints s.t.*

$$\text{neg-cref}: \Sigma_{TC}^- \times \mathcal{U}_{RAC} \times \mathcal{U}_{RAC} \rightarrow \mathcal{U}_C \cup \mathcal{U}_R.$$

Inverses of activity-object relationships are assumed to be functional capturing the intuition that a single occurrence of an activity can manipulate an object at a given point in time. To clarify the syntax of the OCBC modeling language we illustrate the scenario provided in Example 1.

Example 3. We consider the OCBC model in Fig. 2 where the activities are depicted in the upper part of the figure while the lower part shows the ClaM data model for the data manipulated by the activities of the process. The set \mathcal{U}_{RAC} of the *activity-object relationships* is: $\mathcal{U}_{RAC} = \{\text{create}, \text{closes}, \text{fills}, \text{prepares}, \text{refers to}\}$ connecting an activity with the manipulated objects as an effect of executing the activity itself. For example, the activity `CreateOrder` *creates* an instance of the object class `Order` when it is executed. Cardinality constraints can be added to activity-object relationships to specify participation constraints either on the activity side or

on the object class side. For example, each execution of `PickItem` fills one and only one `OrderLine`, i.e., $\#_{act}(\text{fills}, \text{PickItem}) = (1, 1)$. On the other hand, any `OrderLine` must be necessarily filled by executing a `PickItem` activity, i.e., $\#_{obj}(\text{fills}, \text{OrderLine}) = 1$. The *co-reference constraints* involving object classes specify constraints on how objects connected to different activities can be shared. For example, the `OrderLine` instance filled by a `PickItem` is the same as the one prepared by the corresponding `WrapItem`. These co-reference constraints can be expressed using the following OCBC syntax:

$$\begin{aligned} \text{cref}(\text{unary-response}(\text{PickItem}, \text{WrapItem}), \text{fills}, \text{prepares}) &= \text{OrderLine}, \\ \text{cref}(\text{unary-precedence}(\text{WrapItem}, \text{PickItem}), \text{prepares}, \text{fills}) &= \text{OrderLine}. \end{aligned}$$

The co-reference constraint (7), and the negative co-reference constraint (8) are expressed as, respectively:

$$\begin{aligned} \text{cref}(\text{unary-precedence}(\text{PickItem}, \text{CreateOrder}), \text{fills}, \text{creates}) &= \text{contains}; \\ \text{neg-cref}(\text{non-response}(\text{PayOrder}, \text{PickItem}), \text{closes}, \text{fills}) &= \text{contains}. \end{aligned}$$

4.4 Semantics of OCBC Models

We now focus on the semantics of OCBC models. As pointed out in Sect. 2, OCBC models are interpreted using traces that capture the occurrence of events, the relationships between events and objects, and the evolution of objects and relationships over time. Here, we base the OCBC semantics on *infinite* traces (cf. Sect. 6 for a remark on finite traces). The information recorded in an actual execution trace is interpreted under incomplete knowledge, i.e., as a *trace fragment* containing explicit factual knowledge that is known to certainly hold but, in general, only partially capturing what actually occurred. Thus, the notion of trace as used in event log formats such as the XES IEEE standard has to be interpreted, in our setting, as a trace fragment.

Our effort is to reconcile the process flow semantics with the data model semantics. We thus resort to a knowledge base expressed in the temporal DL $T_{US}ALCQI$. In particular, we map both activities and object classes to $T_{US}ALCQI$ concepts, while activity-object relationships and relationships of the data model are mapped to $T_{US}ALCQI$ roles. Such an encoding of OCBC models using KBs in the temporal DL $T_{US}ALCQI$ interprets constraints of an OCBC model over infinite traces, while the ABox, that encodes the explicit factual knowledge, i.e., the *trace fragment* at hand, is interpreted as a finite portion of such infinite traces. Here we detail the encoding.

Concerning the semantics of the *ClAM data model*, we interpret it via a mapping to $ALCQI$ as already discussed in Sect. 4.1. Furthermore, we can add to the data model temporal constraints captured in $T_{US}ALCQI$ as shown in [5, 7].

As for *activity-object relationships*, let $R \in \mathcal{U}_{RAC}$ so that $\tau_{RAC}(R) = (A, O)$. The following $T_{US}ALCQI$ axioms captures *inverse functionality*, and *domain* and *range* restrictions for R :

$$(\geq 2 R^- \top) \sqsubseteq \perp, \quad \exists R \sqsubseteq A, \quad \exists R^- \sqsubseteq O. \quad (1)$$

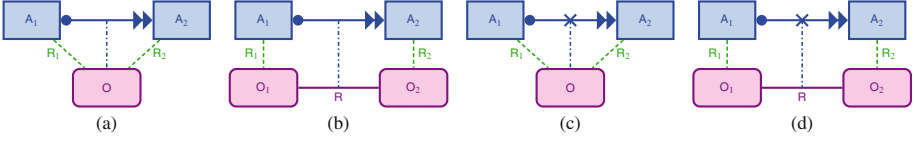


Fig. 5. Co-reference (response) constraints over (a) object classes and (b) relationships, with their negated versions (c-d)

A cardinality constraint of the form $\#_{obj}(R, O) = 1$, denoting the activity that generated an object of class O , is captured as:

$$O \sqsubseteq \diamond_P^+(O \sqcap \exists R^-).$$

Cardinality constraints for the participation of activities in activity-object relationships ($\#_{act}$) are instead captured as classical cardinalities in data models (see [5, 7, 11]).

Semantics of Co-reference Constraints. Having fixed the semantics for the ClaM data model and the one for the activity-object relationships we are left with the most tricky aspect of OCBC, namely the semantics of *co-reference constraints*. In the following, we consider the different kinds of co-reference constraints which, according to Definition 3, can be either positive or negative, and can range either over object classes (as illustrated in Fig. 5a and c) or over relationships (as illustrated in Fig. 5b and d). Let $R_1, R_2 \in \mathcal{U}_{R_{AC}}$, $A_1, A_2 \in \mathcal{U}_A$ and $O \in \mathcal{U}_C$ s.t. $tc(A_1, A_2) \in \Sigma_{TC}^+$, $\tau_{R_{AC}}(R_1) = (A_1, O)$, $\tau_{R_{AC}}(R_2) = (A_2, O)$ and $cref$ be a *co-reference constraint over object classes* of the form: $cref(tc(A_1, A_2), R_1, R_2) = O$ (as in Fig. 5a). Then, *co-reference over object classes* when tc is the **response** temporal constraint is captured by the axiom:

$$\exists R_1^- \sqsubseteq \diamond_F \exists R_2^- \quad (2)$$

This expresses that “*whenever an object is in the range of R_1 then sometime in the future it must be also in the range of R_2* ”. This semantics enforces a temporal constraint over the activities via the co-referenced object, i.e., when the activity A_1 is linked via R_1 to an object in O then it must be followed by an execution of A_2 referencing the *same* object via R_2 . Formally, the following logical implication holds:

$$\{(1), (2), A_1 \sqsubseteq \exists R_1\} \models A_1 \sqsubseteq \exists R_1 \cdot \diamond_F \exists R_2^- \cdot A_2 \quad (3)$$

When tc is the **unary-response** temporal constraint we need to add to formula (2) another formula that guarantees a *unique* occurrence of A_2 over the co-referenced object:

$$\exists R_2^- \sqcap \diamond_F \exists R_1^- \sqsubseteq \square_F \neg \exists R_2^- \quad (4)$$

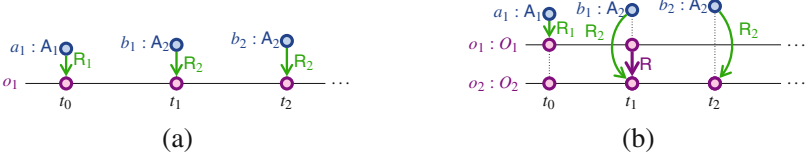


Fig. 6. (a) Trace fragment for (2) but not (4); (b) trace fragment for (8) but not (10)

Figure 6a shows a possible instantiation of the OCBC model in Fig. 5a which, in turn, is not a valid fragment in case the temporal constraint is changed to **unary-response**.

Similar formulas hold when tc is a temporal constraint over the past, i.e., either **precedence** (formula (5)), **unary-precedence** (formulas (5) and (6)) or **responded-existence** (formula (7)).

$$\exists R_1^- \sqsubseteq \diamond_P \exists R_2^- \quad (5)$$

$$\exists R_2^- \sqcap \diamond_F \exists R_1^- \sqsubseteq \square_P \neg \exists R_2^- \quad (6)$$

$$\exists R_1^- \sqsubseteq \diamond \exists R_2^- \quad (7)$$

We now consider *co-reference constraints over relationships*. As in Fig. 5b, let $O_1, O_2 \in \mathcal{U}_C$, $R \in \mathcal{U}_R$, with $\tau(R) = (O_1, O_2)$, $\tau_{RAC}(R_1) = (A_1, O_1)$, $\tau_{RAC}(R_2) = (A_2, O_2)$ and $cref$ be a co-reference of the form: $cref(tc(A_1, A_2), R_1, R_2) = R$. Then, the semantics of *co-reference over relationships* when tc is the **response** constraint is captured by:

$$\exists R_1^- \sqsubseteq \diamond_F \exists R. \exists R_2^- \quad (8)$$

Expressing that “every object in the range of R_1 sometime in the future should be connected via R to an object in the range of R_2 .” A logical implication similar to (3) holds:

$$\{(1), (8), A_1 \sqsubseteq \exists R_1\} \models A_1 \sqsubseteq \exists R_1. \diamond_F \exists R. \exists R_2^- . A_2 \quad (9)$$

When tc is **unary-response** we should add to formula (8) another formula that guarantees that activity A_1 is followed by a single occurrence of A_2 via R . The following axiom expresses that “whenever an object is in the range of R_2 (thus under the occurrence of A_2) and is connected via R^- to an object that before was in the range of R_1 (due to the occurrence of the activity A_1) then, it will never be in the range of R_2 .”

$$\exists R_2^- \sqcap \exists R^- . \diamond_P \exists R_1^- \sqsubseteq \square_P \neg \exists R_2^- \quad (10)$$

Figure 6b shows an instantiation of the OCBC model in Fig. 5b that, in turn, is not anymore a valid fragment in case the temporal constraint is changed to

unary-response (because o_2 is pointed to by two different instances— b_1, b_2 —of the activity A_2).

Similar formulas hold when tc is **precedence** (axiom (11)), **unary-precedence** (axioms (11) and (12)) and **responded-existence** (axiom (13))

$$\exists R_1^- \sqsubseteq \exists R. \diamond_F \exists R_2^- \quad (11)$$

$$\exists R_2^- \sqcap \diamond_F \exists R^- . \exists R_1^- \sqsubseteq \square_P \neg \exists R_2^- \quad (12)$$

$$\exists R_1^- \sqsubseteq \diamond \exists R. \diamond \exists R_2^- \quad (13)$$

Note that axiom (13) allows for **responded-existence** to be symmetric—as for axiom (7)—i.e., $\{(13)\} \models \exists R_2^- \sqsubseteq \diamond \exists R^- . \diamond \exists R_1^-$.

We now consider co-references in the presence of *negative behavioral constraints* (see Fig. 5c-d). We start with co-reference over object classes. In case tc is **non-response** (as in Fig. 5c) then the following axiom expresses that “*whenever an object is in the range of R_1 then never in the future it could be in the range of R_2* ”:

$$\exists R_1^- \sqsubseteq \square_F \neg \exists R_2^- . \quad (14)$$

As a consequence of this axiom, and of the fact that the domains of R_1 and R_2 are activities A_1 and A_2 , while they both range over the same class O , we can also read this negative co-reference as “*every instance of activity A_1 can never be followed by instances of A_2 sharing the same object in O* ”. The right-hand side of the axiom is the negation of the right-hand side of axiom (2). When tc is **non-coexistence**, we have

$$\exists R_1^- \sqsubseteq \square^* \neg \exists R_2^- \quad (15)$$

Again, the right-hand side is the negation of the right-hand side of axiom (7).

When negative co-references involve a relationship and tc is **non-response** (as in Fig. 5d) the following axiom expresses that “*whenever an object is in the range of R_1 then never in the future it could be connected via R to an object in the range of R_2 (thus under the occurrence of A_2)*”:

$$\exists R_1^- \sqsubseteq \square_F \neg \exists R. \exists R_2^- \quad (16)$$

implying that “*every instance of activity A_1 can never be followed by instances of A_2 sharing the same pair of objects in R* ”. Notice again that the right-hand side of the above axiom is the negation of the right-hand side of axiom (8). Finally, by negating the right-hand side of axiom (13) we capture the case when tc is **non-coexistence**

$$\exists R_1^- \sqsubseteq \square^* \neg \exists R. \diamond \exists R_2^- \quad (17)$$

Similar to **responded-existence, non-coexistence** over both object classes (15) and relationships (17) is obviously symmetric. Formally, considering the co-reference over a relationship, $\{(17)\} \models \exists R_2^- \sqsubseteq \boxtimes \neg \exists R^- . \diamond \exists R_1^-$.

Altogether, an OCBC model can be captured via a TBox in $T_{US}ALCQI$, and its trace fragments using corresponding ABoxes. Overall, a $T_{US}ALCQI$ KB is thus able to provide a uniform representation for OCBC, on which we can apply ad hoc reasoning services as described in the following section.

5 Verification and Reasoning over OCBC Models

The main motivation to provide a mapping from OCBC models to a DL Knowledge Base is the possibility of carrying out automated reasoning over them. We discuss how the typical services for verifying declarative, constraint-based process models can be lifted to the more sophisticated setting of OCBC. To do so, we build on the services defined for the well-established DECLARE language [24, 25]. In the following, we show how such services can be reformulated as standard reasoning tasks over $T_{US}ALCQI$ knowledge bases, in turn inheriting their decidability and worst-case complexity.

Let \mathcal{M} be an OCBC model of interest, and ρ a trace fragment over \mathcal{M} . We denote by $\mathcal{T}_{\mathcal{M}}$ and \mathcal{A}_{ρ} the TBox and ABox obtained by encoding \mathcal{M} and ρ in $T_{US}ALCQI$, and by $\mathcal{K}_{\mathcal{M},\rho}$ the resulting $T_{US}ALCQI$ KB, i.e., $\mathcal{K}_{\mathcal{M},\rho} = (\mathcal{T}_{\mathcal{M}}, \mathcal{A}_{\rho})$.

Model Consistency. The most fundamental service is to check whether \mathcal{M} is consistent, that is, supports the empty trace fragment (in turn witnessing that it supports at least one full trace). This directly reduces to check whether $\mathcal{T}_{\mathcal{M}}$ is satisfiable.

Activity Executability. An OCBC model may be consistent, but including so-called *dead activities* [25], i.e., activities that cannot be executed at all. We can show whether an activity A in \mathcal{M} can be executed by verifying whether such an activity is not logically implied to be empty in the corresponding TBox, i.e., $\mathcal{T}_{\mathcal{M}} \not\models A \sqsubseteq \perp$.

Implied Properties. Let α be a model property expressible in $T_{US}ALCQI$. We can check whether $\mathcal{M} \models \alpha$ by checking whether $\mathcal{K}_{\mathcal{M},\rho} \models \alpha$. E.g., (3) is a property implied by \mathcal{M} . The presented encoding of OCBC into $T_{US}ALCQI$ allows us to use its reasoning capabilities to detect so-called *hidden constraints* [24], i.e., constraints that are implicitly present in \mathcal{M} even though they are not shown graphically.

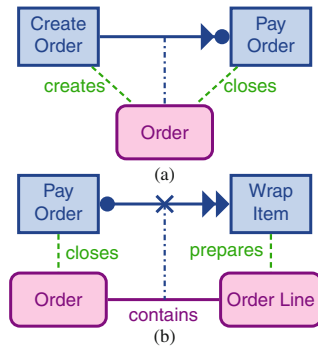


Fig. 7. Implied (a) and non-implied (b) constraints by the OCBC model of Fig. 2

Example 4. Consider again the OCBC model of Fig. 2 and the two constraints in Fig. 7 where Fig. 7a captures that an order can be paid only if it has been created before, and Fig. 7b that no order line of an order can be wrapped after that order is paid. It is easy to verify that the former constraint is indeed implied, while the latter constraint it is not. While it is true that once an order is paid no further items can be picked for it, already picked order lines may still need to be wrapped.

Execution Trace Compliance. This amounts to check whether a trace fragment ρ satisfies the constraints in \mathcal{M} . Since ρ is a trace fragment, we require that no explicit violation is contained in ρ and that ρ can be ‘completed’ into a fully specified, infinite trace that satisfies \mathcal{M} . This corresponds to the notion of *conditional compliance* recently introduced in [15]. In our setting, this amounts to check whether the ABox \mathcal{A}_ρ encoding ρ is satisfiable w.r.t. the TBox $\mathcal{T}_\mathcal{M}$, i.e., whether the KB $\mathcal{K}_{\mathcal{M},\rho}$ is satisfiable.

Complexity Considerations. Notice that, KB satisfiability and logical implication are mutually reducible in *ALCQI* [6] (and thus in $T_{US}ALCQI$) and these reasoning problems over $T_{US}ALCQI$ are ExpTime-complete [18, 27], which establishes an ExpTime upper bound for verifying properties of OCBC models. The need to use *ALCQI* as the base DL is due to co-reference constraints over relationships, which requires the power of qualified existential ($\exists R.C$) and its dual. If we renounce such constraints (i.e., only consider OCBC constraints co-referring on classes), we could use a temporalized version of a *DL-Lite* dialect. In particular, the temporal *DL-Lite* fragment $T_{US}DL-Lite_{bool}^{(\mathcal{LN})}$, showed to be PSpace-complete in [8], is able to capture OCBC models with the exception of co-reference constraints over relationships while, at the level of the data model, $T_{US}DL-Lite_{bool}^{(\mathcal{LN})}$ captures the main constructs of UML—with the exception of ISA between relationships and n-ary relationships (cf. [4, 7] for details).

6 Conclusions

We presented the first, complete formalization of *object-centric behavioral constraints* (OCBC): a new approach to business process modeling where data models and declarative constraints over activities are seamlessly integrated. Our approach comes with a logic-based semantics for OCBC in terms of an encoding into the temporal DL $T_{US}ALCQI$. This unambiguously defines the meaning of OCBC models, and lays the foundations for reasoning over them, allowing us to understand the (decidability and) complexity boundaries of reasoning tasks over OCBC models. $T_{US}ALCQI$ interprets time as a linear, infinite structure, which contrasts with the finite-trace semantics adopted in other declarative process modeling languages such as Declare. The study of temporal description logics with finite-time semantics is rather novel [9], and may constitute the basis for reasoning over OCBC models on finite traces.

We have considered here standard data models to capture the structural aspects of OCBC. Variants of OCBC with non-conventional temporalized cardinality constraints over relationships have been used [21, 22]. We intend to study whether such constraints may impact on the decidability and complexity of reasoning over OCBC models.

In our research agenda, we are interested not only in design-time reasoning of OCBC models, but also in enactment, monitoring, and runtime verification. This poses two major challenges. On the one hand, a monitored trace has to be considered under a “partially closed” semantics, that is, by interpreting it as a complete record of what happened so far, while missing information about the future. On the other hand, a more fine-grained analysis, in the style of [23], regarding if and how a monitored trace conforms to an OCBC model is needed. We intend to attack this problem by combining finite and infinite reasoning over a partially closed knowledge base.

Acknowledgments. This research has been partially supported by the UNIBZ CRC projects PWORM and REKAP.

References

1. van der Aalst, W., Pesic, M., Schonenberg, H.: Declarative workflows: balancing between flexibility and support. *Comput. Sci.-Res. Dev.* **23**(2), 99–113 (2009)
2. van der Aalst, W.M.P.: *Process Mining: Data Science in Action*. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
3. van der Aalst, W.M.P., Li, G., Montali, M.: Object-centric behavioral constraints. CoRR Technical report, CoRR (2017). <http://arxiv.org/abs/1703.05740>
4. Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., Zakharyashev, M.: Reasoning over extended ER models. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) *ER 2007*. LNCS, vol. 4801, pp. 277–292. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75563-0_20
5. Artale, A., Parent, C., Spaccapietra, S.: Evolving objects in temporal information systems. *Ann. Math. Artif. Intell.* **50**(1–2), 5–38 (2007)
6. Artale, A., Calvanese, D., Kontchakov, R., Zakharyashev, M.: The DL-Lite family and relations. *JAIR* **36**, 1–69 (2009)
7. Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyashev, M.: Complexity of reasoning over temporal data models. In: *Proceedings of the 29th International Conference on Conceptual Modeling (ER)*. LNCS, vol. 4801, pp. 277–292. Springer, Heidelberg (2010)
8. Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyashev, M.: A cookbook for temporal conceptual data modeling with description logics. *ACM Transactivity Comput. Logic (TOCL)* **15**(3), 25 (2014)
9. Artale, A., Mazzullo, A., Ozaki, A.: Do you need infinite time? In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)* (2019, to appear)
10. Bagheri Hariri, B., Calvanese, D., Montali, M., De Giacomo, G., De Masellis, R., Felli, P.: Description logic knowledge and action bases. *JAIR* **46**, 651–686 (2013)
11. Berardi, D., Calvanese, D., De Giacomo, G.: Reasoning on UML class diagrams. *Artif. Intell. J.* **168**(1–2), 70–118 (2005)

12. Bhattacharya, K., Gerede, C., Hull, R., Liu, R., Su, J.: Towards formal analysis of artifact-centric business process models. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 288–304. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_21
13. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: the DL-Lite family. *J. Autom. Reasoning* **39**(3), 385–429 (2007)
14. Calvanese, D., De Giacomo, G., Montali, M.: Foundations of data-aware process analysis: a database theory perspective. In: Proceedings of 32nd PODS. ACM (2013)
15. Chesani, F., et al.: Compliance in business processes with incomplete information and time constraints: a general framework based on abductive reasoning. *Fundamenta Informaticae* **159**(3), 1–37 (2018)
16. Cohn, D., Hull, R.: Business artifacts: a data-centric approach to modeling business operations and processes. *IEEE Data Eng. Bull.* **32**(3), 3–9 (2009)
17. Franconi, E., Mosca, A., Solomakhin, D.: ORM2: formalisation and encoding in OWL2. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM 2012. LNCS, vol. 7567, pp. 368–378. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33618-8_51
18. Gabbay, D., Kurucz, A., Wolter, F., Zakharyashev, M.: Many-Dimensional Modal Logics: Theory and Applications. Studies in Logic. Elsevier, Amsterdam (2003)
19. Gonzalez, P., Griesmayer, A., Lomuscio, A.: Verification of GSM-based artifact-centric systems by predicate abstraction. In: Barros, A., Grigori, D., Narendra, N.C., Dam, H.K. (eds.) ICSOC 2015. LNCS, vol. 9435, pp. 253–268. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48616-0_16
20. Levesque, H.J.: Foundations of a functional approach to knowledge representation. *Artif. Intell. J.* **23**, 155–212 (1984)
21. 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
22. 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
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., Pesic, M., van der Aalst, W.M.P., Chesani, F., Mello, P., Storari, S.: Declarative specification and verification of service choreographies. *ACM Trans. TWEB* **4**(1), 3 (2010)
25. Pesic, M., Schonenberg, H., van der Aalst, W.M.: DECLARE: Full support for loosely-structured processes. In: Proceedings of the Eleventh IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), pp. 287–298. IEEE Computer Society (2007)
26. Vianu, V.: Automatic verification of database-driven systems: a new frontier. In: Proceedings of the 12th International Conference on Database Theory (ICDT), pp. 1–13 (2009)
27. Wolter, F., Zakharyashev, M.: Temporalizing description logics. In: *Frontiers of Combining Systems*, pp. 379–401. Research Studies Press-Wiley (2000)



Formal Modeling and SMT-Based Parameterized Verification of Data-Aware BPMN

Diego Calvanese¹, Silvio Ghilardi², Alessandro Gianola¹(✉), Marco Montali¹,
and Andrey Rivkin¹

¹ Faculty of Computer Science, Free University of Bozen-Bolzano, Bolzano, Italy
gianola@inf.unibz.it

² Dipartimento di Matematica, Università degli Studi di Milano, Milan, Italy

Abstract. We propose DAB – a data-aware extension of BPMN where the process operates over case and persistent data (partitioned into a read-only database called catalog and a read-write database called repository). The model trades off between expressiveness and the possibility of supporting parameterized verification of safety properties on top of it. Specifically, taking inspiration from the literature on verification of artifact systems, we study verification problems where safety properties are checked irrespectively of the content of the read-only catalog, and accepting the potential presence of unboundedly many tuples in the catalog and repository. We tackle such problems using an array-based backward reachability procedure fully implemented in MCMT – a state-of-the-art array-based SMT model checker. Notably, we prove that the procedure is sound and complete for checking safety of DABs, and single out additional conditions that guarantee its termination and, in turn, show decidability of checking safety.

1 Introduction

In recent years, increasing attention has been given to multi-perspective models of business processes that strive to capture the interplay between the process and data dimensions [21]. Conventional finite-state verification techniques only work in this setting if data are abstractly represented, e.g., as finite state machines [20] or process annotations [23]. If data are instead tackled in their full generality, verifying whether a process meets desired temporal properties (e.g., is safe) becomes highly undecidable, and cannot be directly attacked using conventional finite-state model checking techniques [1]. This triggered a flourishing research on the formalization and the boundaries of verifiability of data-aware processes, focusing mainly on data- and artifact-centric models [1, 10]. Recent results in this stream of research [4, 11] come with two strong advantages. First, they consider the relevant setting where the running process evolves a set of relations (henceforth called a data *repository*) containing data objects that may have been injected from the external environment (e.g., due to user interaction),

or borrowed from a read-only relational database with constraints (henceforth called *catalog*). The repository acts as a working memory and a log for the process. Notably, it may accumulate unboundedly many tuples resulting from complex constructs in the process, such as while loops whose repeated activities insert new tuples in the repository (e.g., the applications sent by candidates in response to a job offer). The catalog stores background, contextual facts that do not change during the process execution, such as the catalog of product types, the usernames and passwords of registered customers in an order-to-cash process. In this setting, verification is studied parametrically to the catalog, so as to ensure that the process works as desired irrespectively of the specific read-only data stored therein. This is crucial to verify the process under robust conditions, also considering that actual data may not yet be available at modeling time. The second advantage of these techniques is that they tame the infinity of the state space to be verified with a symbolic approach, paving the way for the development of feasible implementations [17] or the usage of mature symbolic model checkers for infinite-state systems [4, 15].

In a parallel research line more conventional, activity-centric approaches, such as the de-facto standard BPMN, have been extended towards data support, mainly focusing on modeling and enactment [6, 7, 18], but not on verification. At the same time, several formalisms have been brought forward to capture multi-perspective processes based on Petri nets enriched with various forms of data: from data items carried by tokens [16, 22], to case data with different datatypes [8], and persistent relational data manipulated with the full power of FOL/SQL [9, 19]. While these formalisms qualify well to directly capture data-aware extensions of BPMN (e.g., [7, 18]), they suffer of two main limitations. On the foundational side, they require to specify the data present in the read-only storage, and only allow boundedly many tuples (with an a-priori known bound) to be stored in the read-write ones. On the applied side, they have not yet led to the development of actual verifiers.

This leads us to the main question tackled by this paper: *how to extend BPMN towards data support, guaranteeing the applicability of the existing parameterized verification techniques and the corresponding actual verifiers, so far studied only in the artifact-centric setting?* We answer this question by considering the framework of [4] and the verification of safety properties (i.e., properties that must hold in every state of the analyzed system). Our *first contribution* is a data-aware extension of BPMN called DAB, which supports case data, as well as persistent relational data partitioned into a read-only catalog and a read-write repository. Case and persistent data are used to express conditions in the process as well as task preconditions; tasks, in turn, change the values of the case variables and insert/update/delete tuples into/from the repository. The resulting framework is similar, in spirit, to the BAUML approach [12], which relies on UML and OCL instead of BPMN as we do here. While [12] approaches verification via a translation to first-order logic with time, we follow a different route, by encoding DABs into the array-based artifact system framework from [4]. Thanks to this encoding, we can effectively verify safety properties of

DABs using the MCMT (*Model Checker Modulo Theories*) model checker [13, 14]. MCMT implements a backward reachability procedure that relies on state-of-the-art Satisfiability Modulo Theories (SMT) solvers, and that has been widely used to verify infinite-state *array-based systems*.

Using the encoding above, we provide our *second contribution*: we show that this backward reachability procedure is sound and complete when it comes to checking safety of DABs. In this context, soundness means that whenever the procedure terminates the returned answer is correct, whereas completeness means that if the process is unsafe then the procedure will always discover it.

The fact that the procedure is sound and complete does not guarantee that it will always terminate. This brings us to the *third and last contribution* of this paper: we introduce further conditions that, by carefully controlling the interplay between the process and data components, guarantee the termination of the procedure. Such conditions are expressed as syntactic restrictions over the DAB under study, thus providing a concrete, BPMN-grounded counterpart of the conditions imposed in [4, 17]. By exploiting the encoding from DABs to array-based artifact systems, and the soundness and completeness of backward reachability, we derive that checking safety for the class of DABs satisfying these conditions is decidable.

To show that our approach goes end-to-end from theory to actual verification, we finally report some preliminary experiments demonstrating how MCMT checks safety of DABs. An extended version of this paper is available in [2]. Full proofs of our technical results and the files of the experiments with MCMT can be found in [3].

2 Data-Aware BPMN

We start by describing our formal model of data-aware BPMN processes (DABs). We focus here on private, single-pool processes, analyzed considering a single case, similarly to soundness analysis in workflow nets [24].¹ Incoming messages are therefore handled as pure nondeterministic events. The model combines a wide range of (block-structured) BPMN control-flow constructs with task, event-reaction, and condition logic that inspect and modify persistent as well as case data.

First, some preliminary notation. We consider a set $\mathcal{S} = \mathcal{S}_v \uplus \mathcal{S}_{id}$ of (semantic) *types*, consisting of *primitive types* \mathcal{S}_v accounting for data objects, and *id types* \mathcal{S}_{id} accounting for identifiers. We assume that each type $S \in \mathcal{S}$ comes with a (possibly infinite) domain \mathbb{D}_S , a special constant $\mathbf{undef}_S \in \mathbb{D}_S$ to denote an undefined value in that domain, and a type-wise equality operator $=_S$. We omit the type and simply write \mathbf{undef} and $=$ when clear from the context. We do not consider here additional type-specific predicates (such as comparison and arithmetic operators for numerical primitive types); these will be added in future

¹ The interplay among multiple cases is also crucial. The technical report [3] already contains an extension of the framework presented here, in which multiple cases are modeled and verified.

work. In the following, we simply use *typed* as a shortcut for \mathcal{S} -typed. We also denote by \mathbb{D} the overall domain of objects and identifiers (i.e., the union of all domains in \mathcal{S}). We consider a countably infinite set \mathcal{V} of typed variables. Given a variable or object x , we may explicitly indicate that x has type S by writing $x : S$. We omit types whenever clear. We indicate a possibly empty tuple $\langle x_1, \dots, x_n \rangle$ of variables as \vec{x} , and write $\vec{x} \subseteq \vec{y}$ if all variables in \vec{x} also appear in \vec{y} .

2.1 The Data Schema

Consistently with BPMN, we consider two main forms of data: *case data*², instantiated and manipulated on a per-case basis; *persistent data* (cf. *data store references* in BPMN), accounting for global data that are accessed by all cases. For simplicity, case data are defined at the whole process level, and are directly visible by all tasks and subprocesses (without requiring the specification of input-output bindings and the like).

To account for persistent data, we consider relational databases. We describe relation schemas by using the *named perspective*, i.e., by assigning a dedicated typed attribute to each component (i.e., column) of a relation schema. Also for an attribute, we use the notation $a : S$ to explicitly indicate its type.

Definition 1. A relation schema is a pair $R = \langle N, A \rangle$, where: (i) $N = R.name$ is the relation name; (ii) $A = R.attrs$ is a nonempty tuple of attributes. \triangleleft

We call $|A|$ the *arity* of R . We assume that distinct relation schemas use distinct names, blurring the distinction between the two notions (i.e., $R.name = R$). We also use the predicate notation $R(A)$ to represent a relation schema $\langle R, A \rangle$. A sample relation schema is $User(Uid:Int, Name:String)$, where the first component represents the id-number of a user, and the second component is the string of her name.

Data Schema. First of all, we define the *catalog*, i.e., a read-only, persistent storage of data that is not modified during the execution of the process. Examples of the cat-relations are product types and registered customers in an order-to-cash scenario.

Definition 2. A catalog *Cat* is a set of relation schemas equipped with single-column primary key and foreign key constraints. We assume that the primary key of relation schema R is always its first attribute, and denote it by $R.id$. The type of $R.id$ is a dedicated id type from \mathcal{S}_{id} (i.e., no two relation schemas from *Cat* have the same id type). If another attribute a of R has as type an id-type $S \in \mathcal{S}_{id}$, then a is a foreign key referencing the relation schema whose primary key has type S . \triangleleft

Example 1. Consider a simplified example of a job hiring process. To store background information related to the process we use the catalog with relation schemas:

² These are called *data objects* in BPMN, but we prefer to use the term *case data* to avoid name clashes with the formal notions.

- *JobCategory*(*Jcid*:jobcatID) - storing the (ids of the) job categories available in the company (e.g., programmer, analyst);
- *User*(*Uid*:userID, *Name*:StringName, *Age*:NumAge) - storing data about users registered to the company website, who may apply to positions offered by the company.

Each case of the process is about a job. Jobs are identified by the type jobcatID.

◁

The full data schema of a BPMN process combines a catalog with: (i) a persistent data *repository*, consisting of updatable relation schemas possibly referring to the catalog; (ii) a set of *case variables*, constituting local data carried by each process case.

Definition 3. A data schema \mathcal{D} is a tuple $\langle \text{Cat}, \text{Repo}, X \rangle$, where (i) $\text{Cat} = \mathcal{D}.\text{cat}$ is a catalog, (ii) $\text{Repo} = \mathcal{D}.\text{repo}$ is a set of relation schemas called repository, and (iii) $X = \mathcal{D}.\text{cvars} \subset \mathcal{V}$ is a finite set of typed variables called case variables.

We use bold-face to distinguish case and normal variables. We call repo-relation (resp., cat-relation) a relation whose schema is in the repository (resp., catalog).

Relation schemas in the repository are not equipped with an explicit primary key, and thus they cannot reference each other, but may contain foreign keys pointing to the catalog or the case identifiers. In particular, similarly to foreign keys in the catalog, every attribute in *Repo* and case variable in *X* whose type is an id-type $S \in \mathcal{S}_{id}$ references a corresponding cat-relation whose primary key has type *S*. It will be clear how tuples can be inserted and removed from the repository once we introduce updates.

Example 2. To manage key information about the applications submitted for the job hiring, the company employs a repository that consists of one relation schema:

Application(*Jcid*:jobcatID, *Uid*:userID, *Score*:NumScore, *Eligible*:Bool)

NumScore is a finite-domain type containing 100 scores in the range [1, 100]. For readability, we use the usual comparison predicates for variables of type NumScore: this is syntactic sugar and does not require to introduce datatype predicates in our framework. Since each posted job is created using a dedicated portal, its corresponding data do not have to be stored persistently and thus can be maintained just for a given case. At the same time, some specific values have to be moved from a specific case to the repository and vice-versa. This is done by resorting to the following case variables $\mathcal{D}.\text{cvars}$: (i) **jc**id : jobcatID references a job type from the catalog, matching the type of job associated to the case; (ii) **u**id : userID references the identifier of a user who is applying for the job associated to the case; (iii) **r**esult : Bool indicates whether the user identified by **u**id is eligible for winning the position or not; (iv) **q**ualif : Bool indicates whether the user identified by **u**id qualifies for directly getting the job

(without the need of carrying out a comparative evaluation of all applicants);
 (v) **winner** : `userID` contains the identifier of the applicant winning the position.

◁

At runtime, a *data snapshot* of a data schema consists of three components:

- An immutable *catalog instance*, i.e., a fixed set of tuples for each relation schema contained therein, so that the primary and foreign keys are satisfied.
- An assignment mapping case variables to corresponding data objects.
- A *repository instance*, i.e., a set of tuples forming a relation for each schema contained therein, so that the foreign key constraints pointing to the catalog are satisfied. Each tuple is associated to a distinct primary key that is not explicitly accessible.

Querying the Data Schema. To inspect the data contained in a snapshot, we need suitable query languages operating over the data schema of that snapshot. We start by considering boolean *conditions* over (case) variables, to express choices in the process.

Definition 4. A condition is a formula of the form $\varphi ::= (x = y) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2$, where x and y are variables from \mathcal{V} or constant objects from \mathbb{D} . ◁

We make use of the standard abbreviation $\varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$.

We now extend conditions to also access the data stored in the catalog and repository, and to ask for data objects subject to constraints. We consider the well-known language of unions of conjunctive queries with atomic negation, which correspond to unions of select-project-join SQL queries with table filters.

Definition 5. A conjunctive query with filters over a data component \mathcal{D} is a formula of the form $Q ::= \varphi \mid R(x_1, \dots, x_n) \mid \neg R(x_1, \dots, x_n) \mid Q_1 \wedge Q_2$, where φ is a condition with only atomic negation, $R \in \mathcal{D}.cat \cup \mathcal{D}.repo$ is a relation schema of arity n , and x_1, \dots, x_n are variables from \mathcal{V} (including $\mathcal{D}.cvars$) or constant objects from \mathbb{D} . We denote by $free(Q)$ the set of variables occurring in Q that are not case variables in $\mathcal{D}.cvars$. ◁

For example, a conjunctive query $JobCategory(c) \wedge c \neq HR$ lists all the job categories available in the company, apart from HR.

Definition 6. A guard G over a data component \mathcal{D} is an expression of the form $q(\vec{x}) \leftarrow \bigvee_{i=1}^n Q_i$, where: (i) $q(\vec{x})$ is the head of the guard with answer variables \vec{x} ; (ii) each Q_i is a conjunctive query with filters over \mathcal{D} ; (iii) for some $i \in \{1, \dots, n\}$, $\vec{x} \subseteq free(Q_i)$. We denote by $casevars(G) \subseteq \mathcal{D}.cvars$ the set of case variables used in G , and by $normvars(G) = \bigcup_{i \in \{1, \dots, n\}} free(Q_i)$ the other variables used in G . ◁

To distinguish guard heads from relations, we write the former in camel case, while the latter shall always begin with capital letters.

Definition 7. A guard G over a data component \mathcal{D} is *repo-free* if none of its atoms queries a relation schema from $\mathcal{D}.repo$. ◁

As anticipated before, this language can be seen as a query language to retrieve data from a snapshot, or as a mechanism to constrain the combinations of data objects that can be injected into the process. E.g., guard $\text{input}(y:\text{string}, z:\text{string}) \rightarrow y \neq z$ returns all pairs of strings that are different from each other. Picking an answer in this (infinite) set of pairs can be seen as a (constrained) user input for y and z .

Going beyond this guard query language (e.g., by introducing universal quantification) would hamper the soundness and completeness of SMT-based verification over the resulting DABs. We will come back to this important aspect in the conclusion.

2.2 Tasks, Events, and Impact on Data

We now formalize how the process can access and update the data component when executing a task or reacting to the trigger of an external event.

The Update Logic. We start by discussing how data maintained in a snapshot can be subject to change while executing the process.

Definition 8. *Given a data schema \mathcal{D} , an update specification α is a pair $\langle G, E \rangle$, where: (i) $G = \alpha.\text{pre}$ is a guard over \mathcal{D} of the form $q(\vec{x}) \leftarrow Q$, called precondition; (ii) $E = \alpha.\text{eff}$ is an effect rule that changes the snapshot of \mathcal{D} , as described next. Each effect rule has one of the following forms:*

(Insert&Set) *INSERT \vec{u} INTO R AND SET $\mathbf{x}_1 = v_1, \dots, \mathbf{x}_n = v_n$, where: (i) \vec{u}, \vec{v} are variables in \vec{x} or constant objects from \mathbb{D} ; (ii) $\vec{x} \in \mathcal{D}.\text{cvars}$ are distinct case variables; (iii) R is a relation schema from $\mathcal{D}.\text{repo}$ whose arity (and types) match \vec{u} . Either the INSERT or SET parts may be omitted, obtaining a pure Insert rule or Set rule.*

(Delete&Set) *DEL \vec{u} FROM R AND SET $\mathbf{x}_1 = v_1, \dots, \mathbf{x}_n = v_n$, where: (i) \vec{u}, \vec{v} are variables in \vec{x} or constant objects from \mathbb{D} ; (ii) $\vec{x} \in \mathcal{D}.\text{cvars}$; (iii) R is a relation schema from $\mathcal{D}.\text{repo}$ whose arity (and types) match \vec{u} . As in the previous rule type, the AND SET part may be omitted, obtaining a pure (repository) Delete rule.*

(Conditional update) *UPDATE $R(\vec{v})$ IF $\psi(\vec{u}, \vec{v})$ THEN η_1 ELSE η_2 , where: (i) \vec{u} is a tuple containing variables in \vec{x} or constant objects from \mathbb{D} ; (ii) ψ is a repo-free guard (called filter); (iii) R is a repo-relation schema; (iv) \vec{v} is a tuple of new variables, i.e., such that $\vec{v} \cap (\vec{u} \cup \mathcal{D}.\text{cvars}) = \emptyset$; (v) η_i is either an atomic formula of the form $R(\vec{u}')$ with \vec{u}' a tuple of elements from $\vec{x} \cup \mathbb{D} \cup \vec{v}$, or a nested IF... THEN... ELSE. \triangleleft*

We now comment on the semantics of update specifications. An update specification α is executable in a given data snapshot if there is at least one answer to $\alpha.\text{pre}$ in that snapshot. If so, the process executor(s) nondeterministically pick an answer, *binding* the answer variables of $\alpha.\text{pre}$ to corresponding data objects in \mathbb{D} . This confirms the interpretation of $\alpha.\text{pre}$ as a *constrained user input* when multiple bindings are available. Once a binding for the answer variables is picked,

the effect rule $\alpha.\text{eff}$ is instantiated with that binding and issued. How this affects the current data snapshot depends on which effect rule is adopted.

If $\alpha.\text{eff}$ is an insert&set rule, the binding is used to *simultaneously* insert a tuple in one of the repo-relations, and update some of the case variables – with the implicit assumption that those not explicitly mentioned in the SET part maintain their current values. Since repo-relations do not have an explicit primary key, upon insertion of a tuple \vec{u} in the instance of a repo-relation R , a fresh primary key is generated and attached to \vec{u} . Two insertion semantics can then be used to characterize the insertion. Under the first semantics of *multiset insertion*, the update always succeeds in inserting \vec{u} into R . Under the *set insertion* semantics, instead, R comes not only with its implicit primary key, but also with a key constraint defined over a subset $K \subseteq R.\text{attrs}$ of its attributes (by default, coinciding with $R.\text{attrs}$ itself); the update is then committed only if such a key constraint is satisfied. In the case of $K = R.\text{attrs}$, this implies that the update succeeds only if \vec{u} is not already present in R , thus treating R as a proper set.

Example 3. We continue the job hiring example, by considering update specifications of type insert&set. When a new case is created, the update specification `InsJobCat` indicates what is the job category associated to the case. Specifically, `InsJobCat.pre` selects a job category from the corresponding `cat`-relation, while `InsJobCat.eff` assigns the selected job category to the case variable `jcId`:

$$\text{InsJobCat.pre} \triangleq \text{getJobType}(c) \leftarrow \text{JobCategory}(c) \quad \text{InsJobCat.eff} \triangleq \text{SET } \mathbf{jcId} = c$$

When the case receives an application, the user id is picked from the corresponding `User` via the update specification `InsUser`, where:

$$\text{InsUser.pre} \triangleq \text{getUser}(u) \leftarrow \text{User}(u, n, a) \quad \text{InsUser.eff} \triangleq \text{SET } \mathbf{uid} = u$$

A different usage of precondition, resembling a pure external choice, is the update specification `CheckQual` to handle a quick evaluation of the candidate and check whether she has such a high profile qualifying her to directly get an offer:

$$\text{CheckQual.pre} \triangleq \text{isQualified}(q : \text{Bool}) \leftarrow \text{true} \quad \text{CheckQual.eff} \triangleq \text{SET } \mathbf{qualif} = q$$

As an example of insertion rule, we consider the situation where the candidate whose id is currently stored in the case variable `uid` has not been directly judged as qualified. She is then subject to a more fine-grained evaluation via the `EvalApp` specification, resulting in a score that is then registered in the repository.

$$\begin{aligned} \text{EvalApp.pre} &\triangleq \text{getScore}(s : \text{NumScore}) \leftarrow 1 \leq s \wedge s \leq 100 \\ \text{EvalApp.eff} &\triangleq \text{INSERT } \langle \mathbf{jcId}, \mathbf{uid}, s, \text{undef} \rangle \text{ INTO } \textit{Application} \end{aligned}$$

Here, the insertion indicates an `undef` eligibility, since it will be assessed in a consequent step of the process. Notice that with the *multiset insertion semantics*, the same user may apply multiple times for the same job (resulting multiple times as applicant). With the *set insertion semantics*, we could instead enforce the uniqueness of the application by declaring the second component (i.e., the user id) of `Application` as a key. \triangleleft

If $\alpha.\text{eff}$ is a `delete&set` rule, then the executability of the update is subject to the fact that the tuple \vec{u} selected by the binding and to be removed from R , is actually present in the current instance of R . If so, the binding is used to *simultaneously* delete \vec{u} from R and update some of the case variables – with the implicit assumption that those not explicitly mentioned in the `SET` part maintain their current values.

Finally, a conditional update rule applies, tuple by tuple, a bulk operation over the content of R . Each tuple in R is reviewed. If the tuple passes the filter associated to the rule, it is updated according to the `THEN` part, otherwise according to the `ELSE` part.

Example 4. Continuing with our running example, we now consider the update specification `MarkE` handling the situation where no candidate has been directly considered as qualified, and so the eligibility of all received (and evaluated) applications has to be assessed. Here we consider that each application is eligible if and only if its evaluation resulted in a score greater than 80. Technically, `MarkE.pre` is a true precondition, and:

$$\begin{aligned} \text{MarkE.eff} &\triangleq \text{UPDATE } \textit{Application}(c, u, s, e) \\ &\quad \text{IF } s > 80 \text{ THEN } \textit{Application}(c, u, s, \text{true}) \text{ ELSE } \textit{Application}(c, u, s, \text{false}) \end{aligned}$$

If there is at least one eligible candidate, she can be selected as a winner using the `SelWinner` update specification, which deletes the selected winner tuple from `Application`, and transfers its content to the corresponding case variables (also ensuring that the `winner` case variable is set to the applicant id). Technically:

$$\begin{aligned} \text{SelWinner.pre} &\triangleq \textit{getWinner}(c, u, s, e) \leftarrow \textit{Application}(c, u, s, e) \wedge e = \text{true} \\ \text{SelWinner.eff} &\triangleq \text{DEL } \langle c, u, s, e \rangle \text{ FROM } \textit{Application} \\ &\quad \text{AND SET } \textit{jcoid} = c, \textit{uid} = u, \textit{winner} = u, \textit{result} = e, \textit{qualif} = \text{false} \end{aligned}$$

Deleting the tuple is useful when the selected winner may refuse the job, and hence should not be considered again if a new winner selection is done. To keep such tuple in the repository, one would just need to remove the `DEL` part from `SelWinner.eff`. ◀

The Task/Event Logic. We now substantiate how the update logic is used to specify the task/event logic within a DAB process. The first important observation, not related to our specific approach, but inherently present whenever the process control flow is enriched with relational data, is that update effects manipulating the repository must be executed in an atomic, non-interruptible way. This is essential to ensure that insertions/deletions into/from the repository are applied on the same data snapshot where the precondition is checked. Breaking simultaneity would lead to nondeterministic interleaving with other update specifications potentially operating over the same portion of the repository. This is why we consider two types of task: *atomic* and *nonatomic*.

Each atomic task/catching event is associated to a corresponding update specification. In the case of tasks, the specification precondition indicates under which circumstances the task can be enacted, and the specification effect how enacting the task impacts on the underlying data snapshot. In the case of events,

the specification precondition constrains the data payload that comes with the event (possibly depending on the data snapshot, which is global and therefore accessible also by an external event trigger). The effect dictates how reacting to a triggered event impacts on the data snapshot.

This is realized according to the following lifecycle. The task/event is initially **idle**, i.e., quiescent. When the progression of a case reaches an **idle** task/event, such a task/event becomes **enabled**. An **enabled** task/event may nondeterministically fire depending on the choice of the process executor(s). Upon firing, a binding satisfying the precondition of the update specification associated to the task/event is selected, consequently grounding and applying the corresponding effect. At the same time, the lifecycle moves from **enabled** to **compl**. Finally, a **compl** task/event triggers the progression of its case depending on the process-control flow, simultaneously bringing the task/event back to the **idle** state (which would then make it possible for the task to be executed again later, if the process control-flow dictates so).

The lifecycle of a nonatomic task diverges in two crucial respects. First, upon firing it moves from **enabled** to **active**, and later on nondeterministically from **active** to **compl** (thus having a duration). The precondition of its update specification is checked and bound to one of the available answers when the task becomes **active**, while the corresponding effect is applied when the task becomes **compl**. Since these two transitions occur asynchronously, to avoid the aforementioned transactional issues we assume that the task effect operates only on case variables (and not on the repository).

2.3 Process Schema

A process schema consists of a block-structured BPMN diagram, enriched with conditions and update effects expressed over a given data schema, according to what described in the previous sections. As for the control flow, we consider a wide range of block-structured patterns compliant with the standard. We focus on private BPMN processes, thereby handling incoming messages in a pure nondeterministic way. So we do for timer events, nondeterministically accounting for their expiration without entering into their metric temporal semantics. Focusing on block-structured components helps us in obtaining a direct, execution semantics, and a consequent modular and clean translation of various BPMN constructs (including boundary events and exception handling). However, it is important to stress that our approach would seamlessly work also for non-structured processes where each case introduces boundedly many tokens.

As usual, blocks are recursively decomposed into sub-blocks, the leaves being task or empty blocks. Depending on its type, a block may come with one or more nested blocks, and be associated with other elements, such as conditions, types of the involved events, and the like. We consider a wide range of blocks, covering the basic, flow, and exception handling patterns in [2]. Figure 1 gives an idea about what is covered by our approach. With these blocks at hand, we finally obtain the full definition of a DAB.

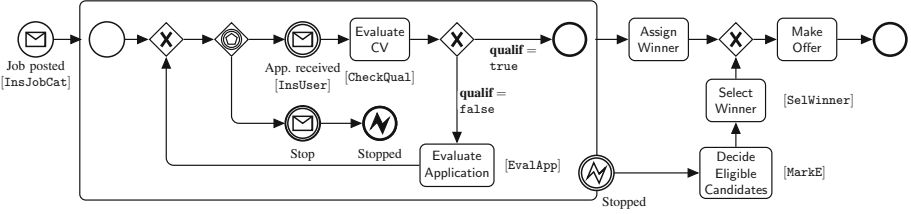


Fig. 1. The job hiring process. Elements in squared brackets attach the update specifications in Examples 3 and 4 to corresponding tasks/events.

Definition 9. A DAB \mathcal{M} is a pair $\langle \mathcal{D}, \mathcal{P} \rangle$ where \mathcal{D} is a data schema, and \mathcal{P} is a root process block such that all conditions and update effects attached to \mathcal{P} and its descendant blocks are expressed over \mathcal{D} . \triangleleft

Example 5. The full hiring job process is shown in Fig. 1, using the update effects described in Examples 3 and 4. Intuitively, the process works as follows. A case is created when a job is posted, and enters into a looping subprocess where it expects candidates to apply. Specifically, the case waits for an incoming application, or for an external message signalling that the hiring has to be stopped (e.g., because too much time has passed from the posting). Whenever an application is received, the CV of the candidate is evaluated, with two possible outcomes. The first outcome indicates that the candidate directly qualifies for the position, hence no further applications should be considered. In this case, the process continues by declaring the candidate as winner, and making an offer to her. The second outcome of the CV evaluation is instead that the candidate does not directly qualify. A more detailed evaluation is then carried out, assigning a score to the application and storing the outcome into the process repository, then waiting for additional applications to come. When the application management subprocess is stopped (which we model through an error so as to test various types of blocks in the experiments reported in Sect. 3.3), the applications present in the repository are all processed in parallel, declaring which candidates are eligible and which not depending on their scores. Among the eligible ones, a winner is then selected, making an offer to her. We implicitly assume here that at least one applicant is eligible, but we can easily extend the DAB to account also for the case where no application is eligible. \triangleleft

As customary, each block has a lifecycle indicating its current state, and how the state may evolve depending on the specific semantics of the block, and the evolution of its inner blocks. In Sect. 2.2 we have already characterized the lifecycle of tasks and catch events. For the other blocks, we continue to use the standard states *idle*, *enabled*, *active*, and *compl*. We use the very same rules of execution described in the BPMN standard to regulate the progression of blocks through such states, taking advantage from the fact that, being the process block-structured, only one instance of a block can be enabled/active at a

given time for a given case. As an example, we describe the lifecycle of a sequence block S with nested blocks B_1 and B_2 (considering that the transitions of S from `idle` to `enabled` and from `compl` back to `idle` are inductively regulated by its parent block): (i) if S is `enabled`, then it becomes `active`, inducing a transition of B_1 from `idle` to `enabled`; (ii) if B_1 is `compl`, then it becomes `idle`, inducing a transition of B_2 from `idle` to `enabled`; (iii) if B_2 is `compl`, then it becomes `idle`, inducing S to move from `active` to `compl`. Analogously for other block types.

2.4 Execution Semantics

We intuitively describe the execution semantics of a case over DAB $\mathcal{M} = \langle \mathcal{D}, \mathcal{P} \rangle$, using the update/task logic and progression rules of blocks as a basis. Upon execution, each state of \mathcal{M} is characterized by an \mathcal{M} -snapshot, which consists of a data snapshot of \mathcal{D} (cf. Sect. 2.1) and an assignment mapping each block in \mathcal{P} to its current lifecycle state. Initially, the data snapshot fixes the immutable content of the catalog $\mathcal{D}.cat$, while the repository instance is empty, the case assignment is initialized to all `undef`, and the control assignment assigns to all blocks in \mathcal{P} the `idle` state, with the exception of \mathcal{P} itself, which is `enabled`. At each moment in time, the \mathcal{M} -snapshot is then evolved by nondeterministically evolving the case through one of the executable steps in the process, depending on the current \mathcal{M} -snapshot. If the execution step is about the progression of the case inside the process control-flow, then the control assignment is updated. If instead the execution step is about the application of some update effect, the new \mathcal{M} -snapshot is obtained according to Sect. 2.2.

3 Parameterized Verification of Safety Properties

We now focus on parameterized verification of DABs using the framework of array-based artifact systems of [4], which bridges the gap between SMT-based model checking of array-based systems [13,14], and data- and artifact-centric processes [10,11].

3.1 Array-Based Artifact Systems and Safety Checking

In general terms, an array-based system describes the evolution of array data structures of unbounded size. The logical representation of an array relies on a theory with two types of sorts: one for the array indexes, and the other for the elements stored in the array cells. Since the content of an array changes

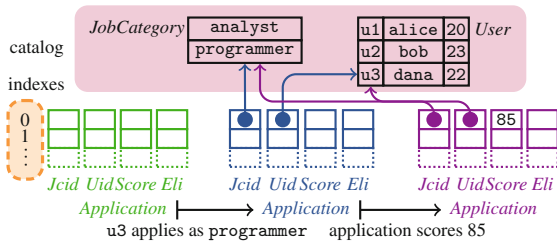


Fig. 2. Array-based representation of the *Application* repo-relation of Example 2, and manipulation of a job application. Empty cells contain `undef`.

over time, it is represented by a *function* variable, called *array state variable*, which defines for each index what is the value stored in the corresponding cell. Its interpretation changes when moving from one state to another, reflecting the intended manipulation of the array. Hence, an array-based system working over array a is defined through: (i) a state formula $I(a)$ describing the *initial configuration(s)* of a ; (ii) a formula $\tau(a, a')$ describing the *transitions* that transforms the content of the array from a to a' . By suitably using logical operators, τ can express in a single formula a set of different updates over a . One of the most studied verification problems is that of *unsafety verification*: it checks whether the evolution induced by τ over a starting from a configuration in $I(a)$ eventually *reaches* an *unsafe* configuration described by a state formula $K(a)$. Several mature model checkers exist to ascertain (un)safety of these type of systems. In our work, we rely on MCMT [15].

In [4], we have extended array-based systems towards an array-based version of the artifact-centric approach, considering in particular the sophisticated model in [17]. In the resulting formalism, called RAS, a *relational artifact system* accesses a read-only database with keys and foreign keys (cf. our DAB catalog). In addition, the RAS operates over a set of relations possibly containing unboundedly many updatable entries (cf. our DAB repository). Figure 2 gives an intuitive idea of how this type of system looks like, using the catalog and repository from Example 2. The catalog is treated as a rich, background theory, which can be considered as a more sophisticated version of the element sort in basic array systems. Each repo-relation is treated as a set of arrays, where each array accounts for one component of the corresponding repo-relation. A tuple in the relation is reconstructed by accessing all such arrays with the same index.

Verification of RAS is also tamed in [4], checking whether there exists an instance of the read-only database so that the RAS can reach an unsafe configuration. This is approached by extending the original symbolic *backward reachability* procedure for unsafety verification of array-based systems to the case of RAS. The procedure starts from the undesired states captured by $K(a)$, and iteratively computes so-called *preimages*, i.e., logical formulae symbolically describing those states that, through consecutive applications of τ , directly or indirectly reach configurations satisfying $K(a)$. A preimage formula may contain existentially quantified variables referring to data objects in the catalog. MCMT employs novel quantifier elimination techniques [4, 5] to suitably remove such variables, and obtain a state formula that describes the predecessor states. A fixpoint check is delegated to a state-of-the-art SMT solver, so as to check whether the computed predecessors all coincide with already iteratively computed states. If no new state has been produced, the procedure stops by emitting *safe*. Otherwise, the preimage formula is conjoined with $I(a)$ and sent to the SMT solver to check for satisfiability: if so, then the computed preimage states intersect the initial ones, and the procedure stops by emitting *unsafe* as a verdict; if not, new preimages are iteratively computed and the steps above are repeated. This procedure is shown to be sound and complete for checking unsafety of RAS in [4], where we also single out subclasses of RAS with decidable unsafety, and for which MCMT

is ensured to terminate. One class is that of RAS operating over arrays whose maximum size is bounded a-priori. This type of RAS is called SAS (for *simple artifact system*). All in all, this framework provides a natural foundational and practical basis to formally analyze DABs, which we tackle next.

3.2 Verification Problems for DABs

First, we need a language to express unsafety properties over a DAB $\mathcal{M} = \langle \mathcal{D}, \mathcal{P} \rangle$. Properties are expressed in a fragment of the *guard* language of Definition 6 that queries repo-relations and case variables as well as the cat-relations that tuples from repo-relations or case variables refer to. Properties also query the control state of \mathcal{P} . This is done by implicitly extending \mathcal{D} with additional, special case *control variables* that refer to the lifecycle states of the blocks in \mathcal{P} (where a block named B gets variable **Blifecycle**). Given a snapshot, each such variable is assigned to the lifecycle state of the corresponding block (i.e., **idle**, **enabled**, and the like).

Definition 10. A property over $\mathcal{M} = \langle \mathcal{D}, \mathcal{P} \rangle$ is a guard G over \mathcal{D} and the control variables of \mathcal{P} , such that every non-case variable in G also appears in a relational atom $R(y_1, \dots, y_n)$, where either R is a repo-relation, or R is a cat-relation and $y_1 \in \mathcal{D}.cvars$.

Example 6. By naming HP the root process block of Fig. 1, the property (**HLifecycle = completed**) checks whether some case of the process can terminate. This property is *unsafe* for our hiring process, since there is at least one way to evolve the process from the start to the end. Since DAB processes are block-structured, this is enough to ascertain that the hiring process is *sound*. Property **EvalAppLifecycle = completed** \wedge $Application(j, u, s, \text{true}) \wedge s > 100$ (the **5th safe** in Sect. 3.3) describes instead the undesired situation where, after the evaluation of an application, there exists an applicant with score greater than 100. The hiring process is *safe* w.r.t. this property. \triangleleft

We study unsafety of these properties by considering the general case, and also the one where the repository can store only boundedly many tuples, with a fixed bound. In the latter case, we call the DAB *repo-bounded*.

Translating DABs into Array-Based Artifact Systems. Given an unsafety verification problem over a DAB $\mathcal{M} = \langle \mathcal{D}, \mathcal{P} \rangle$, we encode it as a corresponding unsafety verification problem over a RAS that reconstructs the execution semantics of \mathcal{M} . We only provide here the main intuitions behind the translation, which is fully addressed in [3]. In the translation, $\mathcal{D}.cat$ and $\mathcal{D}.cvars$ are mapped into their corresponding abstractions in RAS (namely, the RAS read-only database and artifact variables, respectively). $\mathcal{D}.repo$ is instead encoded using the intuition of Fig. 2: for each $R \in \mathcal{D}.repo$ and each attribute $a \in R.attrs$, a dedicated array is introduced. Array indexes represent (implicit) identifiers of tuples in R , in line with our repository model. To retrieve a tuple from R , one just needs to access the arrays corresponding to the different attributes of R with the same

index. Finally, case variables are represented using (bounded) arrays of size 1. On top of these data structures, \mathcal{P} is translated into a RAS transition formula that exactly reconstructs the execution semantics of the blocks in \mathcal{P} .

With this translation in place, we define **BackReach** as the backward reachability procedure that: (1) takes as input (i) a DAB \mathcal{M} , (ii) a property φ to be verified, (iii) a boolean indicating whether \mathcal{M} is repo-bounded or not (in the first case, also providing the value of the bound), and (iv) a boolean indicating whether the semantics for insertion is set or multiset; (2) translates \mathcal{M} into a corresponding RAS \mathcal{M}' , and φ into a corresponding property φ' over \mathcal{M}' (Definition 10 ensures that φ' is indeed a RAS state formula); (3) returns the result produced by the mCMT backward reachability procedure (cf. Sect. 3.1) on \mathcal{M}' and φ' .

3.3 Verification Results

Using the DAB-to-RAS translation and the results in [4], we provide now our main technical contributions. First: DABs can be correctly verified using **BackReach**.

Theorem 1. *BackReach is sound and complete for checking unsafety of DABs that use the multiset or set insertion semantics.* \triangleleft

Soundness tells us that when **BackReach** terminates, it produces a correct answer, while completeness guarantees that whenever a DAB is unsafe with respect to a property, then **BackReach** detects this. Hence, **BackReach** is a semi-decision procedure for unsafety.

We study additional conditions on the input DAB to guarantee termination of **BackReach**, then becoming a full decision procedure for unsafety. The first, unavoidable condition, in line with [4, 17], is that the catalog must be *acyclic*: its foreign keys cannot form referential cycles (where a table directly or indirectly refers to itself).

Theorem 2. *BackReach terminates when verifying properties over repo-bounded and acyclic DABs using the multiset or set insertion semantics.* \triangleleft

If the input DAB is not repo-bounded, acyclicity of the catalog is not enough: termination requires to carefully control the interplay between the different components of the DAB. While the conditions required by the technical proofs are quite difficult to grasp at the syntactic level, they can be intuitively understood using the following *locality principle*: whenever the progression of the DAB depends on the repository, it does so only via a single entry in one of its relations. Hence, direct/indirect comparisons and joins of distinct tuples within the same or different repo-relations cannot be used. To avoid indirect comparisons/joins, queries cannot mix case variables and repo-relations.

Thus, set insertions cannot be supported, since by definition they require to compare tuples in the same relation. The next definition is instrumental to enforce locality.

Definition 11. A guard $G \triangleq q(\vec{x}) \leftarrow \bigvee_{i=1}^n Q_i$ over data component \mathcal{D} is separated if $\text{normvars}(Q_i) \cap \text{normvars}(Q_j) = \emptyset$ for every $i \neq j$, and each Q_i is of the form $\chi \wedge R(\vec{y}) \wedge \xi$ (with χ , $R(\vec{y})$, and ξ optional), where: (i) χ is a conjunctive query with filters only over $\mathcal{D}.\text{cat}$, and that can employ case variables; (ii) $R \in \mathcal{D}.\text{repo}$ is a repo-relation schema; (iii) \vec{y} is a tuple of variables and/or constant objects in \mathbb{D} , such that $\vec{y} \cap \mathcal{D}.\text{cvars} = \emptyset$, and $\text{normvars}(\chi) \cap \vec{y} = \emptyset$; (iv) ξ is a conjunctive query with filters over $\mathcal{D}.\text{cat}$ only, that possibly mentions variables in \vec{y} but does not include any case variable, and such that $\text{normvars}(\chi) \cap \text{normvars}(\xi) = \emptyset$. A property is separated if it is so as a guard. \triangleleft

A separated guard is made of two isolated parts: a part χ inspecting case variables and their relationship with the catalog, and a part $R(\vec{y}) \wedge \xi$ retrieving a single tuple \vec{y} from some repo-relation R , possibly filtering it through inspection of the catalog via ξ .

Example 7. Consider the refinement $\text{EvalApp.pre} \triangleq \text{GetScore}(s : \text{NumScore}) \leftarrow \xi \wedge \chi$ of the guard EvalApp.pre from Example 3, where $\chi := \text{User}(\mathbf{uid}, \text{name}, \text{age})$ checks if the variables $\langle \mathbf{uid}, \text{name}, \text{age} \rangle$ form a tuple in User , and $\xi := 1 \leq s \wedge s \leq 100$. This guard is separated since χ and ξ match the requirements of the previous definition. \triangleleft

Theorem 3. Let \mathcal{M} be an acyclic DAB that uses the multiset insertion semantics, and is such that for each update specification \mathbf{u} of \mathcal{M} , the following holds: 1. If $\mathbf{u}.\text{eff}$ is an insert&set rule (with explicit *INSERT* part), $\mathbf{u}.\text{pre}$ is repo-free; 2. If $\mathbf{u}.\text{eff}$ is a set rule (with no *INSERT* part), then either (i) $\mathbf{u}.\text{pre}$ is repo-free, or (ii) $\mathbf{u}.\text{pre}$ is separated and all case variables appear in the *SET* part of $\mathbf{u}.\text{eff}$; 3. If $\mathbf{u}.\text{eff}$ is a delete&set rule, then $\mathbf{u}.\text{pre}$ is separated and all case variables appear in the *SET* part of $\mathbf{u}.\text{eff}$; 4. If $\mathbf{u}.\text{eff}$ is a conditional update rule, then $\mathbf{u}.\text{pre}$ is repo-free and boolean, so that $\mathbf{u}.\text{eff}$ only makes use of the new variables introduced in its *UPDATE* part (as well as constant objects in \mathbb{D}). Then, *BackReach* terminates when verifying separated properties over \mathcal{M} .

The conditions of Theorem 3 represent a concrete, BPMN-like counterpart of the abstract ones used in [17] and [4] towards decidability.

Specifically, Theorem 3 employs: (i) repo-freedom, and (ii) separation with the manipulation of *all* case variables at once. We intuitively explain how these conditions substantiate the locality principle. Overall, the main difficulty is that case variables may be loaded with data objects extracted from the repository. Hence, the usage of a case variable may mask an underlying reference to a tuple component stored in some repo-relation. Given this, locality demands that no two case variables can simultaneously hold data objects coming from different tuples in the repository. At the beginning, this is trivially true, since all case variables are undefined. A safe snapshot guaranteeing this condition continues to stay so after an insertion of the form mentioned in point 1 of Theorem 3: a repo-free precondition ensures that the repository is not queried at all, and hence trivially preserves locality. Locality may be easily destroyed by arbitrary set or

delete&set rules whose precondition accesses the repository. Three aspects have to be considered to avoid this. First, we have to guarantee that the precondition does not mix case variables and repo-relations: Theorem 3 does so thanks to separation. Second, we have to avoid that when the precondition retrieves objects from the repository, it extracts them from different tuples therein: this is again guaranteed by separation, since only one tuple is extracted. A third, subtle situation that would destroy locality is the one in which the objects retrieved from (the same tuple in) the repository are only used to assign *a proper subset* of the case variables: the other case variables could in fact still hold objects previously retrieved from a *different* tuple in the repository. Theorem 3 guarantees that this never happens by imposing that, upon a set or delete&set operation, *all* case variables are involved in the assignment. Those case variables that get objects extracted from the repository are then guaranteed to all implicitly point to the same repository tuple retrieved by the separated precondition.

Example 8. The hiring DAB obeys to all conditions in Theorem 3, ensuring termination of BackReach. E.g., EvalApp in Example 3 matches point 1: its precondition is repo-free. SelWinner from the same example matches point 3: SelWinner.pre is trivially separated and *all* case variables appear in the SET part of SelWinner.eff. ◀

First mcmt Experiments. We have encoded the job hiring DAB in MCMT, systematically applying the translation rules recalled in Sect. 3.2, and fully spelled out in [3] when proving the main theorems of Sect. 3.3. We have then checked the DAB for process termination (which took 0.43s), and against five safe and five unsafe properties. E.g., the 1st **unsafe** property describes the desired situation in which, after having evaluated an application (i.e., EvalApp is completed), there exists at least an applicant with a score greater than 0. Formally: $\mathbf{EvalApplifecycle} = \mathbf{completed} \wedge \mathit{Application}(j, u, s, e) \wedge s > 0$. The 4th **safe** property represents the situation where a winner has been selected after the deadline (i.e., SelWin is completed), but the case variable **result** indicates that the winner is not eligible. Formally: $\mathbf{SelWinlifecycle} = \mathbf{completed} \wedge \mathbf{result} = \mathbf{false}$. MCMT returns **SAFE**, witnessing that this configuration is not reachable from the initial one. The table on the right summarizes the obtained, encouraging results (time in seconds). The MCMT specifications with all the checked properties (and their intuitive interpretation) are available in [3]. All tests are directly reproducible.

	<i>prop.</i>	<i>time(s)</i>
safe	1	0.20
	2	5.85
	3	3.56
	4	0.03
	5	0.27
unsafe	1	0.18
	2	1.17
	3	4.45
	4	1.43
	5	1.14

4 Conclusion

We have introduced a data-aware extension of BPMN, called DAB, balancing between expressiveness and verifiability. We have shown that parameterized safety problems over DABs can be tackled by array-based SMT techniques, and

in particular the backward reachability procedure of the MCMT model checker. We have then identified classes of DABs with conditions that control how the process operates over data and ensure termination of backward reachability. Methodologically, such conditions can be seen as modeling principles for data-aware process designers who aim at making their processes verifiable. Whether these conditions apply to real-life processes is an open question that calls for novel research in their empirical validation on real-world scenarios, and in the definition of guidelines to refactor arbitrary DABs into verifiable ones.

From the foundational perspective we want to equip DABs with datatypes and arithmetic operators, widely supported by SMT solvers. We also want to attack the main limitation of our approach, namely that guards and conditions are existential formulae, and the only (restricted) form of universal quantification in the update language is that of conditional updates. From the experimental point of view, the initial results obtained in this paper and [4] indicate that the approach is promising. We intend to fully automate the translation from DABs to array-based systems, and benchmark the performance of verifiers for data-aware processes, starting from the examples in [17]: they are inspired by reference BPMN processes, and consequently should be easily encoded as DABs.

References

1. Calvanese, D., De Giacomo, G., Montali, M.: Foundations of data aware process analysis: a database theory perspective. In: Proceedings of the PODS, pp. 1–12 (2013)
2. Calvanese, D., Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: Formal modeling and SMT-based parameterized verification of data-aware BPMN (extended version). Technical report [arXiv:1906.07811](https://arxiv.org/abs/1906.07811) (2019)
3. Calvanese, D., Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: Formal modeling and SMT-based parameterized verification of multi-case data-aware BPMN. Technical report [arXiv:1905.12991](https://arxiv.org/abs/1905.12991) (2019)
4. Calvanese, D., Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: From model completeness to verification of data aware processes. In: Lutz, C., Sattler, U., Tinelli, C., Turhan, A.Y., Wolter, F. (eds.) Description Logic, Theory Combination, and All That. LNCS, vol. 11560, pp. 212–239. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22102-7_10
5. Calvanese, D., Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: Model completeness, covers and superposition. In: Automated Deduction - CADE 27, LNCS (LNAI), vol. 11716. Springer, Cham (2019)
6. Combi, C., Oliboni, B., Weske, M., Zerbato, F.: Conceptual modeling of processes and data: connecting different perspectives. In: Trujillo, J., et al. (eds.) ER 2018. LNCS, vol. 11157, pp. 236–250. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_18
7. De Giacomo, G., Oriol, X., Estañol, M., Teniente, E.: Linking data and BPMN processes to achieve executable models. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 612–628. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_38

8. de Leoni, M., Felli, P., Montali, M.: A holistic approach for soundness verification of decision-aware process models. In: Trujillo, J., et al. (eds.) ER 2018. LNCS, vol. 11157, pp. 219–235. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_17
9. De Masellis, R., Di Francescomarino, C., Ghidini, C., Montali, M., Tessaris, S.: Add data into business process verification: bridging the gap between theory and practice. In: Proceedings of AAAI, pp. 1091–1099. AAAI Press (2017)
10. Deutsch, A., Hull, R., Li, Y., Vianu, V.: Automatic verification of database-centric systems. SIGLOG News **5**(2), 37–56 (2018)
11. Deutsch, A., Li, Y., Vianu, V.: Verification of hierarchical artifact systems. In: Proceedings of the PODS, pp. 179–194 (2016)
12. 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
13. Ghilardi, S., Nicolini, E., Ranise, S., Zucchelli, D.: Towards SMT model checking of array-based systems. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 67–82. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-71070-7_6
14. Ghilardi, S., Ranise, S.: Backward reachability of array-based systems by SMT solving: termination and invariant synthesis. Log. Methods Comput. Sci. **6**(4), 1–48 (2010)
15. Ghilardi, S., Ranise, S.: MCMT: a model checker modulo theories. In: Giesl, J., Hähnle, R. (eds.) IJCAR 2010. LNCS (LNAI), vol. 6173, pp. 22–29. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14203-1_3
16. Lasota, S.: Decidability border for Petri nets with data: WQO dichotomy conjecture. In: Kordon, F., Moldt, D. (eds.) PETRI NETS 2016. LNCS, vol. 9698, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39086-4_3
17. Li, Y., Deutsch, A., Vianu, V.: VERIFAS: a practical verifier for artifact systems. PVLDB **11**(3), 283–296 (2017)
18. Meyer, A., Pufahl, L., Fahland, D., Weske, M.: Modeling and enacting complex data dependencies in business processes. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 171–186. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_14
19. Montali, M., Rivkin, A.: DB-Nets: on the marriage of colored Petri Nets and relational databases. ToPNoC **28**(4), 91–118 (2017)
20. Müller, D., Reichert, M., Herbst, J.: Data-driven modeling and coordination of large process structures. In: Meersman, R., Tari, Z. (eds.) OTM 2007. LNCS, vol. 4803, pp. 131–149. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76848-7_10
21. Reichert, M.: Process and data: two sides of the same coin? In: Meersman, R., et al. (eds.) OTM 2012. LNCS, vol. 7565, pp. 2–19. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33606-5_2
22. Rosa-Velardo, F., de Frutos-Escrig, D.: Decidability and complexity of Petri nets with unordered data. Theor. Comput. Sci. **412**(34), 4439–4451 (2011)
23. Sidorova, N., Stahl, C., Trcka, N.: Soundness verification for conceptual workflow nets with data: early detection of errors with the most precision possible. Inf. Syst. **36**(7), 1026–1043 (2011)
24. Aalst, W.M.P.: Verification of workflow nets. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 407–426. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63139-9_48

Engineering



Estimating Process Conformance by Trace Sampling and Result Approximation

Martin Bauer, Han van der Aa^(✉), and Matthias Weidlich

Department of Computer Science, Humboldt-Universität zu Berlin, Berlin, Germany
{martin.bauer,han.van.der.aa,matthias.weidlich}@hu-berlin.de

Abstract. The increasing volume of event data that is recorded by information systems during the execution of business processes creates manifold opportunities for process analytics. Specifically, conformance checking compares the behaviour as recorded by an information system to a model of desired behaviour. Unfortunately, state-of-the-art conformance checking algorithms scale exponentially in the size of both the event data and the model used as input. At the same time, event data used for analysis typically relates only to a certain interval of process execution, not the entire history. Given this inherent data incompleteness, we argue that an understanding of the overall conformance of process execution may be obtained by considering only a small fraction of a log. In this paper, we therefore present a statistical approach to ground conformance checking in trace sampling and conformance approximation. This approach reduces the runtime significantly, while still providing guarantees on the accuracy of the estimated conformance result. Comprehensive experiments with real-world and synthetic datasets illustrate that our approach speeds up state-of-the-art conformance checking algorithms by up to three orders of magnitude, while largely maintaining the analysis accuracy.

1 Introduction

Process-oriented information systems coordinate the execution of a set of actions to reach a business goal [15]. The behaviour of such systems is commonly described by process models that define a set of activities along with execution dependencies. However, once event data is recorded during runtime, the question of conformance emerges [9]: how do the modelled behaviour of a system and its recorded behaviour relate to each other? Answering this question is required to detect, interpret, and compensate deviations between a model of a process-oriented information system and its actual execution.

Driven by trends such as process automation, data sensing, and large-scale instrumentation of process-related resources, the volume of event data and the frequency at which it is generated is increasing in today's world: Event logs comprise up to billions of events [2]. Also, information systems are subject to

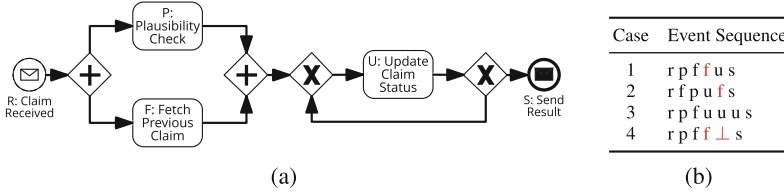


Fig. 1. (a) Model of a claim handling process; (b) events of four process executions.

frequent changes [34], so that analysis is often a continuous process, repeated when new event data becomes available.

Acknowledging the resulting need for efficient analysis, various angles have been followed to improve the runtime performance of state-of-the-art, alignment-based conformance checking algorithms [4], which suffer from an exponential worst-case complexity. Efficiency improvements have been obtained through the use of search-based methods [13, 25], planning algorithms [21], and distributed computing [16, 22]. Furthermore, several authors suggested to compromise correctness and approximate conformance results to gain efficiency, e.g., by employing approximate alignments [29] or applying divide-and-conquer schemes in the computation of conformance results [3, 12, 20, 23]. However, these approaches primarily target the applied algorithms. Fundamentally, they still require the consideration of *all*, possibly billions, of recorded events.

In practice, an event log is recorded for a certain interval of process execution, not the entire history. Given this inherent incompleteness of event data, which is widely acknowledged [1, 17, 26], analysis often strives for a general understanding of the conformance of process execution. This may relate, e.g., to the overall fitness of recorded and modelled behaviour [4] or the activities that denote hotspots of non-conformance [3].

In this paper, we argue that for a general understanding of the overall conformance, it is sufficient to compute conformance results for only a small fraction of an event log. Since the latter per se provides an incomplete view, minor differences in the conformance results obtained for the whole log and a partial log may be attributed to the inherent uncertainty of the conformance checking setting. We illustrate this idea with a claim handling process in Fig. 1. Here, the events recorded for the first case indicate non-conformance, as a previous claim is fetched twice (f). Considering also the second case, the average amount of non-conformance (one deviation) and the set of non-conforming activities ($\{F\}$) are the same, though, despite the different sequence of events. Considering also the third and fourth case, new information on the overall conformance of process execution is obtained. Yet, the fourth case resembles the first one. Hence, its conformance (two deviations w.r.t. the model) may be approximated based on the result of the first case (one deviation) and the difference between both event sequences (one event differs).

To realise the above ideas, we follow two complimentary angles to avoid computation of conformance results for all available data. First, we contribute

an incremental approach based on trace sampling, which, for each trace, assesses whether it yields new information on the overall conformance. Assuming the view of a series of binomial experiments, we establish bounds on the error of the conformance result derived from a partial event log. Second, we show how trace sampling is combined with result approximation that, instead of computing a conformance result for a trace at hand, relies on a worst-case approximation of its implications on the overall conformance. This way, we further reduce the amount of data for which conformance results are actually computed. We instantiate this framework for two types of conformance results as mentioned above, a numerical fitness measure and a distribution of conformance issues over all activities.

In the remainder, we first give preliminaries in Sect. 2. We then introduce our approach to sample-based (Sect. 3) and approximation-based (Sect. 4) conformance checking. Experimental results using real-world and synthetic event logs are presented in Sect. 5. We review related work in Sect. 6, before we conclude in Sect. 7.

2 Preliminaries

Events and Event Logs. We adopt an event model that builds upon a set of activities \mathcal{A} . An event recorded by an information system is assumed to be related to the execution of one of these activities. By \mathcal{E} , we denote the universe of all events. A single execution of a process, called a *trace*, is modelled as a sequence of events $\xi \in \mathcal{E}^*$, such that no event can occur in more than one trace. An event log is a set of traces, $L \subseteq 2^{\mathcal{E}^*}$. Our example in Fig. 1b defines four traces. While each event is unique, we represent them with small letters $\{r, p, f, u, s\}$, that indicate for which activity of the process model, denoted by respective capital letters $\{R, P, F, U, S\}$, the execution is signalled. Two distinct traces that indicate the same sequence of activity executions are of the same *trace variant*.

Process Models. A process model defines the execution dependencies between the activities of a process. For our purposes, it is sufficient to abstract from specific process modelling languages and focus on the behaviour defined by a model. That is, a process model defines a set of execution sequences, $M \subseteq \mathcal{A}^*$, that capture sequences of activity executions that lead the process to its final state. For instance, the model in Fig. 1a defines the execution sequences $\langle R, P, F, U, S \rangle$ and $\langle R, F, P, U, S \rangle$, potentially including additional repetitions of U . We write \mathcal{M} for the set of all process models.

Alignments. State-of-the-art techniques for conformance checking construct *alignments* between traces and execution sequences of a model to detect deviations [4, 23]. An alignment between a trace ξ and a model M , denoted by $\sigma(\xi, M)$ in the remainder, is a sequence of steps, each step comprising a pair of an event and an activity, or a *skip* symbol \perp , if an event or activity is without counterpart. For instance, for the non-conforming trace ξ_1 (case 1 from Fig. 1b), an alignment

is constructed as follows:

$$\frac{\text{Trace } \xi_1 \quad r \ p \ f \ f \ u \ s}{\text{Execution sequence } R \ P \ F \ \perp \ U \ S}$$

Assigning costs to skip steps, a cost-optimal alignment (not necessarily unique) is constructed for a trace in relation to all execution sequences of a model [4]. An optimal alignment then enables the quantification of non-conformance. Specifically, the *fitness* of a log with respect to a given model is computed as follows:

$$\text{fitness}(L, M) = 1 - \frac{\sum_{\xi \in L} c(\xi, M)}{\sum_{\xi \in L} c(\xi, \emptyset) + |L| \times \min_{x \in M} c(\langle \rangle, \{x\})} \quad (1)$$

Here, $c(\xi, M)$ is the aggregated cost of an optimal alignment $\sigma(\xi, M)$. The denominator captures the maximum possible cost per trace, i.e., the sum of the costs of aligning a trace with an empty model, $c(\xi, \emptyset)$, and the minimal costs of aligning an empty trace with the model, $\min_{x \in M} c(\langle \rangle, \{x\})$. Using a standard cost function (all skip steps have equal costs), the fitness value of the example log $\{\xi_1, \xi_2, \xi_3, \xi_4\}$ in Fig. 1b is 0.9. Alignments further enable the detection of hotspots of non-conformance. To this end, the conformance result can be defined in terms of a *deviation distribution* that captures the relative frequency with which an activity (not to be confused with a task of a process model) is part of a conformance violation. For a log L and a model M , this distribution follows from skip steps in the optimal alignments of all traces. It is formalised based on a bag of activities, $dev(L, M) : \mathcal{A} \rightarrow \mathbb{N}_0$ (note that multiple skip steps may relate to a single activity even in the alignment of one trace). The relative deviation frequency of an activity $a \in \mathcal{A}$ is then obtained by dividing the number of occurrences of a in the bag of deviations by the total number of deviations, i.e., $f_{dev(L, M)}(a) = dev(L, M)(a) / |dev(L, M)|$.

For our example in Fig. 1, assuming that skip steps relate to the highlighted trace positions, it holds that $dev(\{\xi_1, \xi_2, \xi_3, \xi_4\}, M) = [F^3, U]$ and $f_{dev(\{\xi_1, \dots, \xi_4\}, M)}(F) = 3/4$, so that the fetching of a previous claim (F) is identified as a hotspot of non-conformance.

3 Sample-Based Conformance Checking

This section describes how trace sampling can be used to improve the efficiency of conformance checking. The general idea is that it often suffices to only compute alignments for a subset of all trace variants to gain insights into the overall conformance of a log to a model. However, we randomly sample an event log trace-by-trace, not by trace variant, which avoids to load the entire log and step-wise reveals the distribution of traces among the variants. At some point, though, the sampled traces then do not provide new information on the overall conformance of the process.

In the remainder of this section, we first describe a general framework for sample-based conformance checking (Sect. 3.1), which we then instantiate for two types of conformance results: fitness (Sect. 3.2) and deviation distribution (Sect. 3.3).

3.1 Statistical Sampling Framework

To operationalise sample-based conformance checking, we regard it as a series of binomial experiments. In this, we follow a log sampling technique introduced in the context of process discovery [5, 7] and lift it to the setting of conformance checking.

Information Novelty. When parsing a log trace-by-trace, some traces may turn out to provide information on the conformance of a log to a model that is similar or equivalent to the information provided by previously encountered traces. To assess whether this is the case, we capture the conformance information associated with a log by a *conformance function* $\psi : 2^{\mathcal{E}^*} \times \mathcal{M} \rightarrow \mathcal{X}$. That is, $\psi(L, M)$ is the conformance result (of some domain \mathcal{X}) between a log L and a model M . If we are interested in the distribution of deviations, ψ provides information on the model activities for which deviations are observed, whereas for fitness, it would return the fitness value.

Based thereon, we define a random Boolean predicate $\gamma(L', \xi, M)$ that captures whether a trace $\xi \in \mathcal{E}^*$ provides new information on the conformance with model M , i.e., whether it changes the result obtained already for a set of previously observed traces $L' \subseteq 2^{\mathcal{E}^*}$. Assuming that the distance between conformance results can be quantified by a function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_0^+$, we define a *new information predicate* as:

$$\gamma(L', \xi, M) \Leftrightarrow d(\psi(L', M), \psi(\{\xi\} \cup L', M)) > \epsilon. \quad (2)$$

Here, $\epsilon \in \mathbb{R}_0^+$ is a relaxation parameter. If incorporating trace ξ changes the conformance result by more than ϵ , then it adds new information over L' .

Framework. We exploit the notion of information novelty for hypothesis testing when sampling traces from an event log L . We determine when *enough* sampled traces have been included in a log $L' \subseteq L$ to derive an understanding of its overall conformance to a model M . Following the interpretation of log sampling as a series of binomial experiments [5], L' is regarded as sufficient if the algorithm consecutively draws a certain number of traces that did not contain new information. Specifically, with δ as a measure that bounds the probability of a newly sampled trace to provide new information over L' , at a significance level α , a minimum sample size N is computed. Based on the normal approximation to the binomial distribution, the latter is given as $N \geq 1/2\delta (-2\delta^2 + z^2 + \sqrt{z})$, where z corresponds to the realisation of a standardised normal random variable for $1 - \alpha$ (one-sided hypothesis test). As such, N is calculated given values for δ and α for the desired levels of similarity and significance, respectively.

Consider $\alpha = 0.01$ and $\delta = 0.05$, so that $N \geq 128$. Hence, after observing 128 traces without new information, sampling can be stopped knowing with 0.99 confidence that the probability of finding new information in the remaining log is less than 0.05.

Using the above formulation, our framework for sample-based conformance checking is presented in Algorithm 1. The algorithm takes as input an event log L , a process model M , the number of trials that need to fail N , a predicate γ

to determine whether a trace provides new information, and a conformance function ψ . Going through L trace-by-trace (lines 3–12), the algorithm conducts a series of binomial experiments that check, if a newly sampled trace provides new information according to the predicate γ (line 5). Once N consecutive traces without new information have been selected, the procedure stops and the conformance result is derived based on the sampled log L' .

Result Re-use. Note that the algorithm provides a conceptual view, in the sense that checking the new information predicate $\gamma(L', \xi, M)$ in line 5 according to Eq. 2, requires the computation of $\psi(L', M)$ and $\psi(\{\xi\} \cup L', M)$ in each iteration. A technical realisation of this algorithm, of course, shall exploit that most types of conformance results can be computed incrementally. For instance, considering fitness and the deviation distribution, an alignment is computed only once per *trace variant*, i.e., per unique sequence of activity executions, and reused in the iterations of the algorithm. Also, the value of $\psi(L', M)$ for $\gamma(L', \xi, M)$ in line 5 is always known from the previous iteration, while the conformance result in line 13 is not actually computed at this stage, as the respective result is known from the last evaluation of $\gamma(L', \xi, M)$.

In the next sections, we discuss how to define γ when the conformance function assesses the fitness of a log to a model, or the observed deviation distribution.

3.2 Sample-Based Fitness

The overall conformance of a log to a model may be assessed by considering the log fitness (see Sect. 2) as a conformance function, $\psi_{fit}(L, M) = \text{fitness}(L, M)$. Then, determining whether a trace ξ provides new information over a log sample L' requires us to assess, if incorporating ξ leads to a difference in the overall fitness for the sampled log. Following Eq. 2, we capture this by computing the absolute difference between the fitness value for traces in the sample L' and the value of the sample plus the new trace:

$$d_{fit}(\psi_{fit}(L', M), \psi_{fit}(\{\xi\} \cup L', M)) = |\text{fitness}(L', M) - \text{fitness}(\{\xi\} \cup L', M)|. \quad (3)$$

If this distance is smaller than the relaxation parameter ϵ , the change in the overall fitness value induced by trace ξ is considered to be negligible.

To illustrate this, consider a scenario with $\epsilon = 0.03$ and a sample consisting of the traces ξ_1 and ξ_3 of our running example (Fig. 1). Then, the log fitness for $\{\xi_1, \xi_3\}$ is 0.95. In this situation, if the next sampled trace is ξ_2 , the distance function yields $|\text{fitness}(\{\xi_1, \xi_3\}, M) - \text{fitness}(\{\xi_1, \xi_2, \xi_3\}, M)| = 0.95 - 0.93 = 0.02$. In this case, since the distance is smaller than ϵ , we would conclude that the additional consideration of ξ_2 does not provide new information. By contrast, considering trace ξ_4 would yield a distance of $0.95 - 0.89 = 0.06$. This indicates that trace ξ_4 would imply a considerable change in the overall fitness value, i.e., it provides new information.

Algorithm 1. Framework for Sample-Based Conformance Checking

```

input :  $L$ , an event log;  $M$ , a process model;  $N$ , a number of failed trials to observe;
          $\gamma$ , a predicate that holds true, if a trace provides new information;
          $\psi$ , a conformance function.
output:  $\psi(L')$ , the conformance results for sampled traces.

1  $L', \hat{L} \leftarrow \emptyset$ ; /* The sampled logs, overall and for current experiment series */
2  $i \leftarrow 0$ ; /* The number of current iterations without new information */
3 repeat /* Repeat until  $N$  traces without new information have been seen */
4    $\xi \leftarrow \text{select}(L \setminus \hat{L})$ ; /* Sample a single trace */
   /* Check if  $\xi$  provides new information. Re-uses alignments of traces of
   the same variant and the previous conformance result  $\psi(L', M)$ . */
5   if  $\gamma(L', \xi, M)$  then
6      $i \leftarrow 0$ ; /* Reset the counter */
7      $\hat{L} \leftarrow \emptyset$ ; /* Reset log for current experiment series */
8   else
9      $i \leftarrow i + 1$ ; /* Increment the counter */
10     $\hat{L} \leftarrow \hat{L} \cup \{\xi\}$ ; /* Add trace to log for current experiment series */
11     $L' \leftarrow L' \cup \{\xi\}$ ; /* Add trace to overall sampled log */
12 until  $i \geq N \vee L' = L$ ;
13 return  $\psi(L')$ ; /* Return results based on the overall sampled log */
    
```

3.3 Sample-Based Deviation Distributions

Next, we instantiate the above framework for conformance checking based on the deviation distribution. As detailed in Sect. 2, this distribution captures the relative frequency with which activities are related to conformance issues.

To decide whether a trace ξ provides new information over a log sample L' , we assess if the deviations obtained for ξ lead to a considerable difference in the overall deviation distribution. As such, the distance function for the predicate γ needs to quantify the difference between two discrete frequency distributions. This suggests to employ the L1-distance, also known as the *Manhattan distance*, as a measure:

$$d_{dev}(\psi_{dev}(L', M), \psi_{dev}(\{\xi\} \cup L', M)) = \sum_{a \in \mathcal{A}} |f_{dev}(L', M)(a) - f_{dev}(\{\xi\} \cup L', M)(a)| \quad (4)$$

Taking up our example from Fig. 1, processing only the trace ξ_1 , all deviations are related to the activity of fetching an earlier claim, i.e., $f_{dev}(\{\xi_1\}, M)(F) = 1$. Notably, this does not change when incorporating traces ξ_2 and ξ_3 , i.e., $f_{dev}(\{\xi_1, \xi_2, \xi_3\}, M)(F) = 1$, as they do not provide new information in terms of the deviation distribution. If, after processing trace ξ_1 , however, we sample ξ_4 , we do observe such a difference: based on $f_{dev}(\{\xi_1, \xi_4\}, M)(F) = 2/3$ and $f_{dev}(\{\xi_1, \xi_4\}, M)(U) = 1/3$, we compute a Manhattan distance of $2/3$. With a relaxation parameter ϵ that is smaller than this value, we conclude that ξ_4 provides novel information.

Given the distance functions based on trace fitness and deviation distribution, it is interesting to note that these behave differently, as illustrated in our

example: If the log is $\{\xi_1, \xi_2\}$ and trace ξ_3 is sampled next, the overall fitness changes. Yet, since ξ_3 is a conforming trace, it does not provide new information on the distribution of deviations.

4 Approximation-Based Conformance Checking

This section shows how conformance results can be approximated, further avoiding the need to compute a conformance result for certain traces. Our idea is to derive a worst-case approximation for traces that are similar to variants for which results have previously been derived. Approximation complements the sampling method of Sect. 3: Even when a trace of a yet unseen variant is sampled, we decide whether to compute an actual conformance result or whether to approximate it. As such, the decision on whether a trace provides new information may be taken either based on a computed or an approximated result.

Against this background, our technique for approximation-based conformance checking, formalised in Algorithm 2, extends our procedure given in Algorithm 1. In fact, it primarily provides a realisation of checking the new information predicate $\gamma(L', \xi, M)$, as done in line 5 of Algorithm 1. That is, whether the sampled trace ξ , of an unseen variant, provides new information is potentially decided based on the approximated, rather than computed impact of it on the overall conformance result. At the same time, however, the algorithm also needs to keep track of all sampled traces $L'' \subseteq L'$, for which the approximated results shall be used whenever a conformance result is computed. This leads to an adaptation of the conformance function ψ , i.e., we consider a *partially approximating conformance* function $\hat{\psi} : 2^{\mathcal{E}^*} \times 2^{\mathcal{E}^*} \times \mathcal{M} \rightarrow \mathcal{X}$. Given a log L' and a subset $L'' \subseteq L'$, this function approximates the conformance result $\psi(L', M)$ by computing solely $\psi(L' \setminus L'', M)$, i.e., the impact of traces L'' is not precisely computed. In the same way, to use the approximation technique as part of sample-based conformance checking, the use of the conformance function ψ in Algorithm 1 also has to be adapted accordingly.

Turning to the details of Algorithm 2, its input includes a log sample L' , a sampled trace $\xi \notin L'$, and a process model M , i.e., the arguments of γ in line 5 of Algorithm 1, as well as a distance function d and relaxation parameter ϵ from the definition of γ (Eq. 2). Moreover, there is a similarity threshold k to determine which traces may be used for approximation. Finally, the aforementioned set of traces L'' for which results shall be approximated and the respective adapted conformance function $\hat{\psi}$ are given as input.

From the sampled traces for which approximation is not applied (i.e., $L' \setminus L''$), the algorithm first selects the trace that is most similar to ξ , referred to as the reference trace ξ_r (line 1). Then, we assess whether this similarity is above the threshold k (line 2). If not, we check the trace for new information as done before, just using the adapted conformance function (lines 9–10). If ξ_r is sufficiently similar, however, we perform a worst-case approximation of the impact of ξ on the overall conformance result based on ξ_r (line 3). As part of that, we may obtain several different approximations Φ , each of which is checked whether it

Algorithm 2. Framework for Approximation-Based Conformance Checking

```

input :  $L'$ , a log sample;  $M$ , a process model;  $\xi$ , a trace sampled of a yet unseen
          variant with  $\xi \notin L'$ ;  $d$ , a distance function;  $\epsilon$ , a relaxation parameter;
           $k$ , a similarity threshold;  $\hat{\psi}$ , a partially approximating conformance function;
           $L'' \subseteq L'$ , traces for which approximated results are used.
output:  $(v, L'')$ , where  $v \in \{true, false\}$  indicates whether  $\xi$  provides new information;
           $L''$  is the updated set of traces, for which approximated results are used.

1  $\xi_r \leftarrow \operatorname{argmin}_{\xi' \in L' \setminus L''} \operatorname{sim}(\xi, \xi')$ ;           /* Select most similar trace */
2 if  $\operatorname{sim}(\xi, \xi_r) \leq k$  then                                   /* Check if  $\xi$  and  $\xi_r$  are  $k$ -similar */
3    $\Phi \leftarrow \operatorname{approx}(\xi, \xi_r, L', M)$ ;                 /* Derive all possible approximations */
4   if  $\exists \phi \in \Phi : d(\hat{\psi}(L', L'', M), \phi) > \epsilon$  then
5     /* Approx. indicates new information, use actual result */
6     return  $(true, L'')$ ;
7   else /* Approx. does not indicate new information, use approx. result */
8     return  $(false, L'' \cup \{\xi\})$ ;
9   else /* No  $k$ -similar trace available, check for new information */
10   $v \leftarrow d(\hat{\psi}(L', L'', M), \hat{\psi}(\{\xi\} \cup L', L'', M)) > \epsilon$ ;
    return  $(v, L'')$ ;
    
```

indicates new information over the current sample L' (line 4). Only if this is not the case, we conclude that ξ indeed does not provide new information and, by adding it to L'' make sure that its impact on the overall conformance will always only be approximated, but never precisely computed (line 7).

Next, we give details on the assessment of trace similarity (function *sim*, Sect. 4.1) and the conformance result approximation (function *approx*, Sect. 4.2).

4.1 Trace Similarity

Given a trace ξ and the part of the sample log for which approximation did not apply ($L' \setminus L''$), Algorithm 2 requires us to identify a reference trace ξ_r that is most similar according to some function $\operatorname{sim} : \mathcal{E}^* \times \mathcal{E}^* \rightarrow [0, 1]$. As we consider conformance results that are based on alignments, we define this similarity function based on the alignment cost of two traces. To this end, we consider a function c_t , which, in the spirit of function c discussed in Sect. 2, is the sum of the costs assigned to skip steps in an optimal alignment of two traces. To obtain a similarity measure, we normalise this aggregated cost by a maximal cost, which is obtained by aligning each trace with an empty trace. This normalisation resembles the one discussed for the fitness measure in Sect. 2. We define the similarity function for traces as $\operatorname{sim}(\xi, \xi') = 1 - c_t(\xi, \xi') / (c_t(\xi, \langle \rangle) + c_t(\xi', \langle \rangle))$.

Considering trace $\xi_4 = \langle r, p, f, f, s \rangle$ of our running example, the most similar trace (assuming equal costs for all skip steps) is $\xi_1 = \langle r, p, f, f, u, s \rangle$, with $c_t(\xi_1, \xi_4) = 1$ and, thus, $\operatorname{sim}(\xi_1, \xi_4) = 10/11$.

4.2 Conformance Result Approximation

In the approximation step of Algorithm 2, we derive a set of worst-case approximations of the impact of the trace ξ on the overall conformance result, using the reference trace ξ_r (which is at least k -similar). Based thereon, it is decided whether ξ provides new information. The approximation, however, depends on the type of conformance result.

Fitness Approximation. To approximate the impact of trace ξ on the overall fitness, we compute a single value, i.e., $approx(\xi, \xi_r, L', M)$ in line 3 of Algorithm 2 yields a singleton set. This value is derived by reformulating Eq. 3, which captures the change in fitness induced by a sample trace. That is, we assess the difference between the current fitness, $fitness(L', M)$, and an approximation of the fitness when incorporating ξ , i.e., $fitness(\{\xi\} \cup L', M)$. This approximation, denoted by $\widehat{fit}(\xi, \xi_r, L', M)$, is derived from (i) the change in fitness induced by the reference trace ξ_r , and (ii) the differences between ξ and ξ_r . The former is assessed using the aggregated alignment cost $c(\xi_r, M)$, whereas the latter leverages the aggregated cost of aligning the traces, $c_t(\xi, \xi_r)$. Normalising these costs, function $approx_{\widehat{fit}}(\xi, \xi_r, L', M)$ yields a worst-case approximation for the change in overall fitness imposed by ξ , as follows:

$$approx_{\widehat{fit}}(\xi, \xi_r, L', M) = \left\{ \left| fitness(L', M) - \widehat{fit}(\xi, \xi_r, L', M) \right| \right\} \quad (5)$$

$$\widehat{fit}(\xi, \xi_r, L', M) = 1 - \frac{\sum_{\xi' \in L'} c(\xi', M) + c(\xi_r, M) + c_t(\xi, \xi_r)}{\sum_{\xi' \in L'} c(\xi', \emptyset) + \max_{\xi' \in \{\xi, \xi_r\}} c(\xi', \emptyset) + (|L'| + 1) \min_{x \in M} c(\langle \cdot \rangle, \{x\})}$$

Turning to our running example, assume that we have sampled $\{\xi_1, \xi_2\}$ and computed the precise fitness value based on both traces, which is $1 - 2/(12 + 10) \approx 0.909$ using a standard cost function. If trace ξ_4 is sampled next, we approximate its impact using the similar trace ξ_1 . To this end, we consider $c(\xi_1, M) = 1$ and $c_t(\xi_1, \xi_4) = 1$, which yields an approximated fitness value of $1 - (2 + 1 + 1)/(12 + 6 + 15) = 1 - 4/33 \approx 0.879$. This is close to the actual fitness value for $\{\xi_1, \xi_2, \xi_4\}$, which is $1 - 4/(17 + 15) \approx 0.875$. The minor difference stems from ξ_1 being slightly longer than ξ_4 .

Deviation Distribution Approximation. To approximate the impact of trace ξ on the deviation distribution, we follow a similar approach as for fitness approximation. However, we note that the approximation function here yields a set of possible values, as there are multiple different distributions to be considered when measuring the Manhattan distance to the current distribution. The reason being that the difference between ξ and the reference trace ξ_r induces a set of possible changes of the distribution.

Specifically, we denote by $ed(\xi, \xi_r)$ the edit distance of the two traces, i.e., the pure number of skip steps in their alignment. This number gives an upper bound for the number of conformance issues that need to be incorporated in addition to those stemming from the alignment of the reference trace and the model, i.e., $dev(\{\xi_r\}, M)$. Yet, the exact activities are not known, so that we need to consider all bags of activities of size $ed(\xi, \xi_r)$, the set of which is denoted by $[A]^{ed(\xi, \xi_r)}$.

Each of these bags leads to a different approximation $\widehat{f_{dev}}(\beta, \xi_r, L', M)$ of the distribution $f_{dev}(\{\xi\} \cup L', M)$ that we are actually interested in. We compute those as follows:

$$\begin{aligned} \text{approx}_{dev}(\xi, \xi_r, L', M) &= \bigcup_{\beta \in [\mathcal{A}]^{ed(\xi, \xi_r)}} \left\{ \sum_{a \in \mathcal{A}} \left| f_{dev}(L', M)(a) - \widehat{f_{dev}}(\beta, \xi_r, L', M)(a) \right| \right\} \\ \widehat{f_{dev}}(\beta, \xi_r, L', M)(a) &= \frac{dev(L', M)(a) + dev(\{\xi_r\}, M)(a) + \beta(a)}{|dev(L', M)| + |dev(\{\xi_r\}, M)| + |\beta|} \end{aligned} \quad (6)$$

Consider our example again: Based on $\{\xi_1, \xi_2\}$, we determine that $dev(\{\xi_1, \xi_2\}, M) = [F^2]$ and $f_{dev}(\{\xi_1, \xi_2\}, M)(F) = 1$. If ξ_4 is then sampled, we obtain an approximation based on $dev(\{\xi_1\}, M) = [F]$ and $ed(\xi_4, \xi_1) = 1$. We therefore consider the change in the distribution incurred by approximating the deviations of ξ_4 as $[F] \uplus \beta$ with $\beta \in \{[R], [P], [F], [U], [S]\}$. For instance, $\widehat{f_{dev}}([R], \xi_4, \{\xi_1, \xi_2\}, M)$ yields a distribution assigning relative frequencies of $3/4$ and $1/4$ to activities F and R , respectively.

The above approximation may be tuned heuristically by narrowing the set of activities that are considered for β , i.e., the possible deviations incurred by the difference between ξ and ξ_r . While this means that $\widehat{f_{dev}}$ is no longer a worst-case approximation, it may steer the approximation in practice, hinting at which activities shall be considered for possible deviations. Such an approach is also beneficial for performance reasons: Since β may be any bag built of the respective activities, the exponential blow-up limits the applicability of the approximation to traces that are rather similar, i.e., for which $ed(\xi, \xi_r)$ is small.

Here, we describe one specific heuristic. First, we determine the overlap between ξ and ξ_r in terms of their maximal shared prefix and suffix of activities, for which the execution is signalled by their events. Next, we determine the events that are *not* part of the shared prefix and suffix, and derive the activities referenced by these events. Only these activities are then considered for the construction of β .

In our example, traces ξ_1 and ξ_4 share the prefix $\langle r, p, f, f \rangle$ and suffix $\langle s \rangle$. Thus, ξ_1 contains one event between the shared prefix and suffix, u , while there is none for ξ_4 . Hence, we consider a single bag of deviations, $\beta = [U]$, and $\widehat{f_{dev}}([U], \xi_4, \{\xi_1, \xi_2\}, M)$ is the only distribution considered in the approximation.

5 Evaluation

This section reports on an experimental evaluation of the proposed techniques for sample-based and approximation-based conformance checking. Section 5.1 describes the three real-world and seven synthetic event logs used in the experimental setup, described in Sect. 5.2. The evaluation results demonstrate that our techniques achieve considerable efficiency gains, while still providing highly accurate conformance results (Sect. 5.3).

5.1 Datasets

We conducted our experiments based on three real-world and seven synthetic event logs, which are all publicly available.

Real-World Data. The three real-world event logs differ considerably in terms of the number of unique traces they contain, as well as their average trace lengths, which represent key characteristics for our approach.

- *BPI-12* [31] is a log of a process for loan or overdraft applications at a Dutch financial institute that was part of the Business Process Intelligence (BPI) Challenge. The log contains 13,087 traces (4,366 variants), with 20.0 events per case (avg.).
- *BPI-14* [32] is the log of an ICT incident management process used in the BPI Challenge. For the experiments, we employed the event log of incidence activities, containing 46,616 traces (31,725 variants), with 7.3 events per case (avg.).
- *Traffic Fines* [11] is a log of an information system managing road traffic fines. The log contains 150,370 traces (231 variants), with 3.7 events per case (avg.).

We obtained accompanying process models for these logs using the inductive miner infrequent [19] with its default parameter settings (i.e., 20% noise filtering).

Synthetic Data. To analyse the scalability of our techniques, we considered a synthetic dataset designed to stress-test conformance checking techniques [24]. It consists of seven process models and accompanying event logs. The models are considerably large and complex, as characterised in Table 1, which impacts the computation of alignments. Furthermore, the included event logs consist of a high number of variants (compared to the number of traces), which may affect the effectiveness of log sampling.

5.2 Experimental Setup

We employed the following measures and experimental setup to conduct the evaluation.

Measures. We measure the *efficiency* of our techniques by the fraction of traces from a log required to obtain our conformance results. This fraction indicates for how many traces the conformance computation was not needed due to the trace not being sampled, or the result being approximated. Simultaneously, we consider the fraction of the total trace variants for which our techniques actually had to establish alignments. As these fractions provide us with analytical measures of efficiency, we also assess the *runtime* of our techniques, based on a prototypical implementation. Again, this is compared to the runtime of the conformance checking over the complete log. Finally, we assess the impact of sampling and

Table 1. Characteristics of the synthetic model-log pairs

Characteristic	PrA	PrB	PrC	PrD	PrE	PrF	PrG
Activities	363	317	317	429	275	299	335
Traces	1,200	1,200	500	1,200	1,200	1,200	1,200
Variants	1,049	1,126	500	1,200	1,200	1,200	1,200
Events per trace (avg.)	31.6	41.5	42.9	248.6	98.8	240.8	143.1

approximation on the *accuracy* of conformance results. We determine the accuracy by comparing the results, i.e., the fitness value or the deviation distribution, obtained using sampling and approximation, to the results for the total log.

All presented results are determined based on 20 experimental runs (i.e., replications) of which we report on the mean value, along with the 10th and 90th percentiles.

Environment. Our approach has been implemented as a plugin in ProM [33], which is publicly available.¹ For the computation of alignments, we rely on the ProM implementation of the search-based technique recently proposed in [13]. Runtime measurements have been obtained on a PC (Dual-Core, 2.5 GHz, 8 GB RAM) running Oracle Java 1.8.

5.3 Evaluation Results

This section first considers the overall efficiency and accuracy of our approach on the real-world event logs, using default parameter values ($\delta = 0.01$, $\alpha = 0.99$, $\epsilon = 0.01$, and $k = 1/3$), before conducting a sensitivity analysis in which these values are varied. Lastly, we demonstrate the scalability of our approaches by showing that their performance also applies to complex, synthetic datasets.

Efficiency. We first explore the efficiency of our approach in terms of sample size and runtime for four configurations: conformance in terms of fitness, without (f) and with approximation (fa), as well as for the deviation distribution without (d) and with approximation (da). Figure 2 reveals that all configurations only need to consider a tiny fraction of the complete log. For instance, for BPI-14, the sample-based fitness computation (f) requires only 685 traces (on average) out of the total of 46,616 traces (i.e., 1.5% of the log). This sample included traces from 144 out of the total 31,725 variants (less than 0.5%), which means that the approach established just 144 alignments. As expected, these gains are propagated to the runtime of our approach, as shown in Fig. 3.

When looking at the overall efficiency results, we observe that the additional use of approximation generally does not lead to considerable improvements in comparison to just sampling. However, this is notably different for the deviation distribution of the Traffic Fines dataset. Here, without approximation, the

¹ https://github.com/Martin-Bauer/Conformance_Sketching.

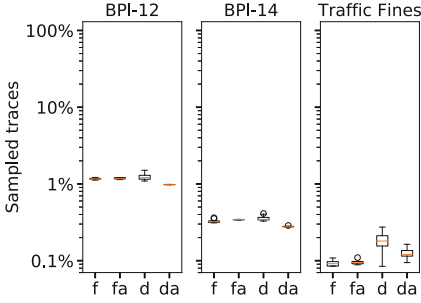


Fig. 2. Sample size efficiency

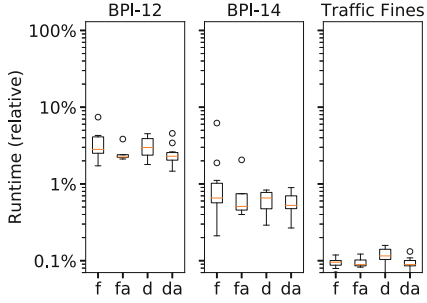


Fig. 3. Runtime efficiency

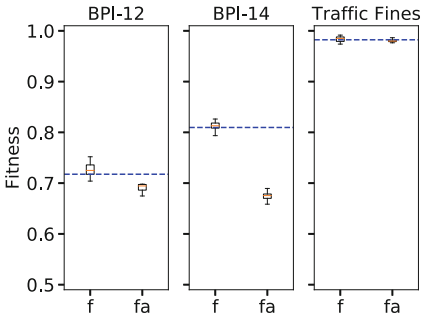


Fig. 4. Accuracy (fitness)

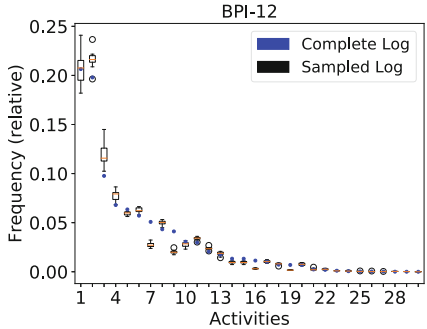


Fig. 5. Accuracy (deviation distribution)

sample size is 1323 on average (0.88%), whereas the sample size drops to 713 traces (0.47%) with approximation (da). Hence, approximation appears to be more important if sampling alone is not as effective.

Accuracy. The drastic gains in efficiency are obtained while maintaining highly accurate conformance results. According to Fig. 4, the fitness computed using sample-based conformance checking differs by less than 0.1% from the original fitness (indicated by the dashed line). Since the accuracy in terms of deviation distribution is harder to capture in a single value, we use Fig. 5 to demonstrate that the deviation distribution obtained by our sample-based technique closely follows the distribution for the complete log. In decreasing order, Fig. 5 depicts the activities with their numbers of deviations observed in the complete log and in the sampled log. As shown, our technique clearly identifies which activities are most often affected by deviations, i.e., our technique correctly identifies the main hotspots of non-conformance. Although, for clarity, not depicted, our approach including approximation achieved comparable results.

Parameter Sensitivity. We performed a parameter sensitivity analysis using sample-based conformance checking on the BPI-12 dataset. We explored how parameters δ (probability bound), α (significance value), and ϵ (relaxedness

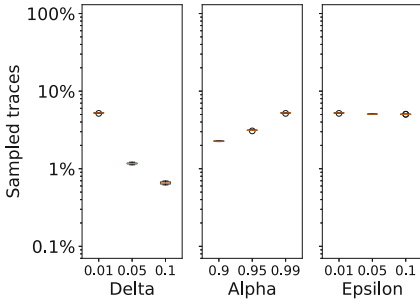


Fig. 6. Sample size sensitivity

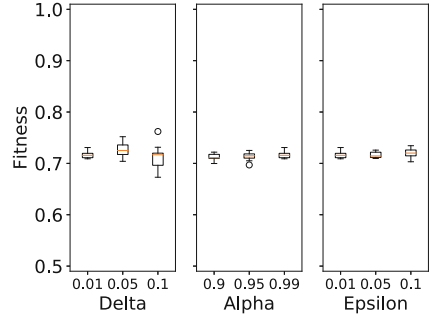


Fig. 7. Fitness sensitivity

value), affect the performance of our approach in terms of efficiency (number of traces) and accuracy (fitness).² Figure 6 shows that selection of δ and α have a considerable impact on the sample required for conformance checking. For instance, for $\delta = 0.01$ we require an average sample size of 684.9, whereas when we relax the bound to $\delta = 0.10$, the average size is reduced to 85.8 traces. For α , i.e., the confidence, we observe a range from 296.5 to 684.9 traces. By contrast, varying epsilon is shown to result in only marginal differences (ranging between 659.0 and 684.9). Still, for all these results, it should be considered that even the largest sample sizes represent only 5% of the traces in the original log.

Notably, as shown in Fig. 7, the average accuracy of our approach remains highly stable throughout this sensitivity analysis, ranging from 0.711 to 0.726. However, we do observe that the variance across replications differs for the parameter settings using smaller sample sizes, specifically for $\delta = 0.1$. Here, the obtained fitness values range between 0.67 and 0.76. This indicates that for such sample sizes, the selection of the particular sample may impact the obtained conformance result in some replications.

Scalability. The results obtained for the synthetic datasets confirm that our approaches are able to provide highly accurate conformance checking results in a small fraction of the runtime. Here, we reflect on experiments performed using our fitness-based sampling approach with $\delta = 0.05$, $\alpha = 0.99$, and $\epsilon = 0.01$. Figure 8 shows that, for six out of seven cases, runtime is reduced to 21.2% to 25.5% of the time needed for the total log (sample sizes range from 10.7% to 12.2%). At the same time, for all cases, the obtained fitness results are virtually equivalent to those of the total log, see Fig. 9, where the fitness values of the total logs are given by the crosses. When comparing these results to those of the real-world datasets, it should be noted that the synthetic logs hardly have any re-occurring trace variants, which makes it harder to generalize over the sampled results. This is particularly pronounced for process *PrC*: There is virtually no difference between the fitness obtained for the total log and the samples. Yet,

² While keeping the other parameter values stable at their respective defaults.

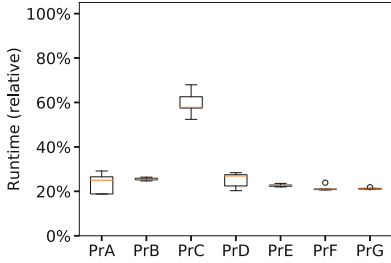


Fig. 8. Runtime efficiency, sample-based fitness computation (synthetic data)

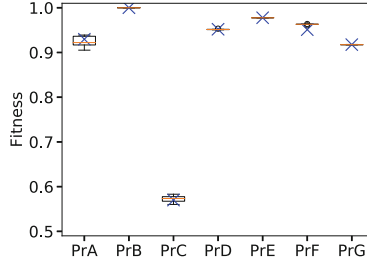


Fig. 9. Fitness (synthetic data). Crosses denote fitness of the total log.

the relatively low fitness value of 0.57 along with a comparatively small number of traces in the log (500 vs. 1,200) lead to a runtime of 59% with sampling. Still, overall, the results on the synthetic data demonstrate that our approach is beneficial in highly complex scenarios.

6 Related Work

Conformance checking can be grounded in various notions. Non-conformance may be detected based on a comparison of sets of binary relations defined over events of a log and activities of a model, respectively [35]. Other work suggests to ‘replay’ traces in a process model, thereby identifying whether events denote valid activity executions [27]. However, both of these streams have limitations with respect to completeness [9]. Therefore, alignment-based conformance checking techniques [4, 23], on which we focus in this work, are widely recognized as the state-of-the-art.

Acknowledging the complexity associated with the establishment of alignments, various approaches [13, 16, 25, 28], discussed in Sect. 1, have been developed to improve runtime efficiency. Other work also aims to achieve efficiency gains by approximating alignments, cf., [14, 30]. While these approaches can lead to efficiency gains, all of them fundamentally depend on the consideration of an entire event log. Moreover, our angle for approximation is different: We do not approximate alignments, but estimate a conformance result (fitness, deviation distribution) based on the distance between traces.

The sampling technique that we employ to avoid this is based on sampling used in sequence databases, i.e., datasets that contain traces. Sampling techniques for event logs have been previously applied for specification mining [10], for mining of Markov Chains [6], and for process discovery [5, 8]. However, we are the first to apply these sampling techniques to conformance checking, a use case in which computational efficiency is arguably even more important than in discovery scenarios.

7 Conclusion

In this paper, we argued that insights into the overall conformance of an event log with respect to a process model can be obtained without computing conformance results for *all* traces. Specifically, we presented two angles to achieve efficient conformance checking: First, through trace sampling, we achieve that only a small share of the traces of a log are considered in the first place. By phrasing this sampling as a series of random experiments, we are able to give guarantees on the introduced error in terms of a potential difference of the overall conformance result. Second, we introduced result approximation as a means to avoid the computation of conformance results even for some of the sampled traces. Exploiting similarities of two traces, we derive an upper bound for the conformance of one trace based on the conformance of another trace. Both techniques, trace sampling and result approximation, have been instantiated for two notions of conformance results, fitness as a numerical measure of overall conformance and the deviation distribution that highlights hotspots of non-conformance in terms of individual activities.

Our experiments highlight dramatic improvements in terms of conformance checking efficiency: Only 0.1% to 1% of the traces of real-world event logs (12% for synthetic data) need to be considered, which leads to corresponding speed-ups of the observed runtimes. At the same time, the obtained conformance results, whether defined as fitness or the deviation distribution, are virtually equivalent to those obtained for the total log.

In future work, we intend to lift our ideas to conformance checking that incorporates branching conditions in a process model or temporal deadlines. Moreover, we strive for an integration of divide-and-conquer schemes [18] in our approximation approach.

References

1. Van der Aalst, W.M.P.: Process Mining - Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-3-642-19345-3>
2. Aalst, W.M.P.: Data scientist: the engineer of the future. In: Mertins, K., Bénaben, F., Poler, R., Bourrières, J.-P. (eds.) Enterprise Interoperability VI. PIC, vol. 7, pp. 13–26. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04948-9_2
3. der Aalst, W.M.P.V., Verbeek, H.M.W.: Process discovery and conformance checking using passages. *Fundam. Inform.* **131**(1), 103–138 (2014)
4. Adriansyah, A., van Dongen, B., Van der Aalst, W.M.P.: Conformance checking using cost-based fitness analysis. In: EDOC, pp. 55–64 (2011)
5. Bauer, M., Senderovich, A., Gal, A., Grunse, L., Weidlich, M.: How much event data is enough? A statistical framework for process discovery. In: CAiSE, pp. 239–256 (2018)
6. Biermann, A.W., Feldman, J.A.: On the synthesis of finite-state machines from samples of their behavior. *IEEE Trans. Comput.* **100**(6), 592–597 (1972)
7. Busany, N., Maoz, S.: Behavioral log analysis with statistical guarantees. In: ICSE, pp. 877–887. ACM (2016)

8. Carmona, J., Cortadella, J.: Process mining meets abstract interpretation. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6321, pp. 184–199. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15880-3_18
9. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance Checking - Relating Processes and Models. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-99414-7>
10. Cohen, H., Maoz, S.: Have we seen enough traces? In: ASE, pp. 93–103. IEEE (2015)
11. De Leoni, M.M., Mannhardt, F.F.: Road traffic fine management process (2015). <https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5>
12. Dixit, P.M., Buijs, J.C.A.M., Verbeek, H.M.W., van der Aalst, W.M.P.: Fast incremental conformance analysis for interactive process discovery. In: Abramowicz, W., Paschke, A. (eds.) BIS 2018. LNBIP, vol. 320, pp. 163–175. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93931-5_12
13. Dongen, B.F.: Efficiently computing alignments - using the extended marking equation. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 197–214. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_12
14. van Dongen, B., Carmona, J., Chatain, T., Taymouri, F.: Aligning modeled and observed behavior: a compromise between computation complexity and quality. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 94–109. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_7
15. 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>
16. Evermann, J.: Scalable process discovery using map-reduce. IEEE TSC **9**(3), 469–481 (2016)
17. van Hee, K.M., Liu, Z., Sidorova, N.: Is my event log complete? - a probabilistic approach to process mining. In: International Conference on Research Challenges in Information Science, pp. 1–7 (2011)
18. Lee, W.L.J., Verbeek, H.M.W., Munoz-Gama, J., der Aalst, W.M.P.V., Sepúlveda, M.: Recomposing conformance: closing the circle on decomposed alignment-based conformance checking in process mining. Inf. Sci. **466**, 55–91 (2018)
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. Leemans, S.J.J., Fahland, D., der Aalst, W.M.P.V.: Scalable process discovery and conformance checking. Softw. Syst. Model. **2**, 599–631 (2018)
21. de Leoni, M., Marrella, A.: How planning techniques can help process mining: the conformance-checking case. In: Italian Symposium on Advanced Database Systems, p. 283 (2017)
22. Luo, C., He, F., Ghezzi, C.: Inferring software behavioral models with MapReduce. Sci. Comput. Program. **145**, 13–36 (2017)
23. Munoz-Gama, J., Carmona, J., van der Aalst, W.: Single-entry single-exit decomposed conformance checking. Inf. Syst. **46**, 102–122 (2014)
24. (Jorge) Munoz-Gama, J.: Conformance checking in the large (dataset) (2013). <https://data.4tu.nl/repository/uuid:44c32783-15d0-4dbd-af8a-78b97be3de49>

25. Reißner, D., Conforti, R., Dumas, M., Rosa, M.L., Armas-Cervantes, A.: Scalable conformance checking of business processes. In: Panetto, H., et al. (eds.) OTM 2017. LNCS, vol. 10573, pp. 607–627. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69462-7_38
26. Rogge-Solti, A., Senderovich, A., Weidlich, M., Mendling, J., Gal, A.: In log and model we trust? A generalized conformance checking framework. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 179–196. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_11
27. Rozinat, A., Van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* **33**(1), 64–95 (2008)
28. Taymouri, F., Carmona, J.: Model and event log reductions to boost the computation of alignments. In: Ceravolo, P., Guetl, C., Rinderle-Ma, S. (eds.) SIMPDA 2016. LNBIP, vol. 307, pp. 1–21. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74161-1_1
29. Taymouri, F., Carmona, J.: A recursive paradigm for aligning observed behavior of large structured process models. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 197–214. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_12
30. Taymouri, F., Carmona, J.: An evolutionary technique to approximate multiple optimal alignments. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 215–232. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_13
31. Van Dongen, B.: BPI Challenge 2012 (2012). <https://doi.org/10.4121/uiid:3926db30-f712-4394-aebc-75976070e91f>
32. Van Dongen, B.: BPI Challenge 2014 (2014). <https://doi.org/10.4121/uiid:c3e5d162-0cfd-4bb0-bd82-af5268819c35>
33. Verbeek, E., Buijs, J.C.A.M., van Dongen, B.F., der Aalst, W.M.P.V.: ProM 6: the process mining toolkit. In: Business Process Management (Demonstration Track) (2010)
34. Weber, B., Reichert, M., Rinderle-Ma, S.: Change patterns and change support features - enhancing flexibility in process-aware information systems. *DKE* **66**(3), 438–466 (2008)
35. Weidlich, M., Polyvyanyy, A., Desai, N., Mendling, J., Weske, M.: Process compliance analysis based on behavioural profiles. *Inf. Syst.* **36**(7), 1009–1025 (2011)



Trace Clustering on Very Large Event Data in Healthcare Using Frequent Sequence Patterns

Xixi Lu^{1(✉)}, Seyed Amin Tabatabaei², Mark Hoogendoorn²,
and Hajo A. Reijers¹

¹ Department of Information and Computing Sciences,
Utrecht University, Utrecht, The Netherlands
{x.lu,h.a.reijers}@uu.nl

² Department of Computer Science, Vrije Universiteit Amsterdam,
Amsterdam, The Netherlands
{s.tabatabaei,m.hoogendoorn}@vu.nl

Abstract. Trace clustering has increasingly been applied to find homogenous process executions. However, current techniques have difficulties in finding a meaningful and insightful clustering of patients on the basis of healthcare data. The resulting clusters are often not in line with those of medical experts, nor do the clusters guarantee to help return meaningful process maps of patients' clinical pathways. After all, a single hospital may conduct thousands of distinct activities and generate millions of events per year. In this paper, we propose a novel trace clustering approach by using sample sets of patients provided by medical experts. More specifically, we learn frequent sequence patterns on a sample set, rank each patient based on the patterns, and use an automated approach to determine the corresponding cluster. We find each cluster separately, while the frequent sequence patterns are used to discover a process map. The approach is implemented in ProM and evaluated using a large data set obtained from a university medical center. The evaluation shows F1-scores of 0.7 for grouping kidney injury, 0.9 for diabetes, and 0.64 for head/neck tumor, while the process maps show meaningful behavioral patterns of the clinical pathways of these groups, according to the domain experts.

Keywords: Trace clustering · Frequent sequential patterns · Process mining · Machine learning

1 Introduction

Clinical pathways are known to be enormously complex and flexible. Process mining techniques are often applied to analyze event data related to clinical pathways, in order to obtain valuable insights [1]. The resulting findings can help to improve process quality, patient outcomes and satisfaction, and optimizing resource planning, usages, and reallocation [2]. Finding coherent, relatively

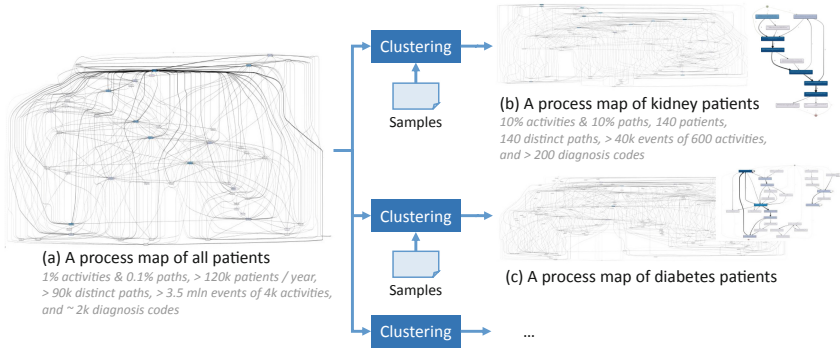


Fig. 1. An example of the partial trace clustering problem and an overview of our approach using sample sets and frequent sequence patterns.

homogenous patient groups helps process mining techniques to obtain accurate insights [3–5].

Many existing systems have tried to classify patients and provide such a well-defined group, known as Patient classification systems (PCSs). PCSs provide a categorization of patients based on clinical data (i.e. diagnoses, procedures), demographic data (i.e. age, gender), and resource consumption data (i.e. costs, length of stay) [6]. While useful, such systems often do not align well with the patient groups as clinicians would define them. Patients who have received the same diagnosis (codes) may be treated for different purposes. For example, patients who get reconstructive breast surgery caused by breast cancer or by gender change can be assigned to the same group, while they have different characteristics and should be assigned to different groups, according to medical experts [7]. Consequently, the process models derived for such a patient group is often also inaccurate and not aligned with the pathways which clinicians would have in their mind. As a result, much manual work involving medical experts is needed to obtain meaningful patient groups.

Emerging from the process mining discipline, trace clustering techniques aim to help finding such homogenous groups of process instances (in our case the patients) [3–5,8]. These techniques cluster the process instances based on the similarity between the sequences of executed activities. However, when applied on hospital data, these approaches face several challenges. Firstly, they have difficulties in scaling-up to handle such large data sets, which may contain hundreds of thousands of patients and millions of events per year. Secondly, they assume that the cases within a group show more homogenous behavior than the cases of different groups, whereas in healthcare, patients treated for the same purpose could have disjoint paths and vice versa. Thirdly, feature vectors (or other intermediate models) are often used to represent the cases and to compute similarity measures; the resulting clusters are often based on an average of the measures and, therefore, may not have clear criteria and may be difficult to explain. Finally, a resulting cluster of patients could still have thousands

of distinct activities, which prevents any process discovery algorithm to find a reasonable process model.

In this paper, we propose a novel perspective to the trace clustering problem. We use sample sets to find one patient cluster at a time by exploiting frequent sequential pattern mining techniques, exemplified in Fig. 1. More specifically, *we assume for each group a small sample set of patients (i.e., patient ids) that belongs to the group is made available by medical experts*. Using the available event data of patient pathways of the sample set, we compute frequent sequence patterns (FSPs) to learn the behavioral criteria of the group. All patients are ranked based on the behavioral criteria, and we use thresholds to automatically determine whether they belong to the group (see Fig. 2, discussed in Sect. 4). Each group is clustered independently. The obtained sequence patterns are used to discover simple process maps.

The approach is implemented in the Process Mining toolkit ProM¹ and evaluated using three real-life cases obtained from a large academic hospital in the Netherlands. The results are validated with a semi-medical expert and a data analyst of the hospital, both of them work closely with medical experts; the semi-medical expert is a manager who has acquired relevant medical knowledge regarding the patient pathways.

The contribution of this work is that it gives a concrete method to identify patient clusters from a wealth of data and high variety of pathways with relatively little input from experts. Moreover, this clustering will lead to simple process maps of frequent behavioral patterns in the clinical pathways that can be used in the communication with medical experts. Such a method may be useful to reason about clinical pathways within hospitals for the sake of process improvement or quality control.

In the remainder, we first discuss related work in Sect. 2. We recall the concepts and define the research problem in Sect. 3. The proposed approach is described in Sect. 4. The evaluation results are presented in Sect. 5, and Sect. 6 concludes the paper.

2 Related Work

In this section, we discuss three streams of trace clustering techniques, categorizing them by their similarity measures.

Feature-Vector-Based Similarity. Early work in trace clustering has followed the ideas in traditional data clustering. Each trace is transformed into a vector of features based on, for example, the frequency of activities, the frequency of *directly-followed* relations, the resources involved, etc. Between these feature vectors, various distance metrics in data mining are reused to estimate the similarity between the traces. Subsequently, distance-based clustering algorithms are deployed, such as k-means or agglomerative hierarchical clustering algorithms [3, 4, 8].

¹ <http://www.promtools.org/>, in the *TraceClusteringFSM* package (see [source code](#)).

In line with feature-vector based trace clustering techniques, the work of Greco et al. [3] was one of the first approaches that incorporated trace clustering into process discovery algorithms. Their work uses frequent (sub)sequences of activities to constitute feature vectors that represent traces. Hierarchical clusters are then built using a top-down approach which iteratively refines the most imprecise process model (represented as disjunctive workflow schemas). Song et al. [4] present a technique that generalizes the *feature space* by considering data attributes in other dimensions than solely focusing on the control-flow. Features of traces in one dimension are grouped into a so-called *trace-profile*, e.g., resource, performance, case attribute profiles, etc. Furthermore, a multitude of vector-based distance metrics and clustering techniques (both partitioning and hierarchical) are deployed. In [8], Bose and van der Aalst compute reoccurring sequences of activities, known as *tandem arrays*, and used these patterns as features in the feature space model in order to improve the way the control-flow information is taken into account in trace clustering.

Trace-Sequence Based Clustering. The second category proposes that the similarity can be measured by the syntax similarity between two trace sequences. A trace can be edited into another trace by adding and removing events. The similarity between two traces is measured by the number of the edit operations needed. An example of this category of measure is the Levenshtein Edit Distance (LED). Bose and van der Aalst [9] propose a trace-sequence distance by generalizing the LED and use the agglomerative clustering technique. Chatain et al. [10] assume that a normative process model is available and align the traces with the runs of the model. In essence, the traces that are close to the same run (in terms of sequence distance) are clustered into the same group.

Model-Based Trace Clustering. Recently, the aim of trace clustering to discover better models has become more prominent. Consequently, the definition of the similarity between traces has shifted from the traces themselves to the quality of the models discovered from those traces. In essence, it is proposed that a trace is more similar to a cluster of traces, if a more fitting, precise, and simple model can be discovered from the cluster [5, 11, 12].

Early work in sequence clustering used first-order Markov models as the intermediate models to represent the clusters. In 2003, Cadez et al. [11] proposed to learn a mixture of first-order Markov models from user behavior by applying the Expectation Maximization problem. The approach is evaluated on a web navigation data set. Later, Ferreira et al. [12] followed the same idea and qualitatively evaluated the clustering algorithm in a process mining setting using two additional data sets.

De Weerd et al. [5] use Petri nets as intermediate models and optimize a F-measure of the models discovered from the clusters. The algorithm, called ActiTraC, first samples distinct traces, based on frequency or distance, as initial clusters. The traces that “fit” into the intermediate-model of a cluster are assigned to the cluster. The remaining noisy traces either are distributed over the clusters or returned as a garbage cluster. More recently, De Koninck et al. [13] proposed to incorporate domain knowledge by assuming that a complete

clustering solution is provided by experts. The proposed technique then aims to improve the quality of such a complete expert-driven clustering in terms of the model qualities.

Discussion. Regarding the feature-based techniques, the number of possible features can be immense, especially in process mining [4]. For example, for n activities, we could have n^2 number of *directly-followed* relations and n^3 if we consider three activities. With thousands of distinct activities, it can be computationally expensive if we consider the full feature space. Moreover, as the clusters are calculated based on the average distances between feature vectors, it is often difficult to explain the reason of a particular clustering. In many cases, the feature-based techniques have difficulties in finding clusters that are in line with those of domain experts [13]. Sequence-based trace clustering faces similar limitations as the feature-based. Furthermore, patients who have disjoint sets of activities and diagnosis (codes) may belong to the same group. Both feature-based and sequence-based would have difficulties finding those. For model-based trace clustering techniques, it would be difficult to handle the complexity of the intermediate models. The clinical pathway of a well-defined patient group could still be extremely complex with thousands distinct activities being executed and each patient following a unique path tailored towards their conditions (see Sect. 5.1). Assuming that a complete expert-driven clustering is available would put too much effort on medical experts and is not feasible for this reason. Our approach, therefore, needs to be scalable and able to deal with this complexity. The approach should also put more emphasis on the abundant domain knowledge available and find clear behavioral criteria of the clusters such that the behavioral criteria are meaningful for domain or medical experts.

3 Research Problem

In this section, we first recall the preliminary concepts such as event logs, traces, and activities. Using these concepts, we define our research problem.

3.1 Preliminaries

A *process* describes a set of *activities* executed in a certain order. For example, each patient in a hospital follows a certain *process* to treat a certain diagnosis of a disease, also known as *clinical pathway*. An *event log* is a set of *traces*, each describing a sequence of *events* through the process. Each *event* records additional information regarding the executed *activity*. For example, Table 1 shows a snippet of an event log of a healthcare process. Each row records an executed event, which contains information such as the event id, the patient id, the activity, the timestamps, the diagnosis code (also known as diagnosis-related group (DRG) [6], or DBC in Dutch), and potentially some additional attributes regarding the event.

Table 1. An example of an event log of a healthcare process

Event	PID	Activity	Time stamps	DBC	Attr.
1	1001	Registration (Reg)	22-10-2018	DBC1	...
2	1001	Doctor appointment (Doc)	23-10-2018	DBC1	...
3	1001	Lab test (Lab)	24-10-2018	DBC1	...
4	1001	Surgery (Srg)	30-10-2017	DBC2	...
5	1001	Doctor appointment (Doc)	01-11-2017	DBC2	...
21	1002	Registration (Reg)	23-10-2017	DBC3	...
22	1002	Lab test (Lab)	25-10-2017	DBC3	...
23	1002	Surgery (Srg)	26-10-2017	DBC4	...
31	1003	Registration (Reg)	25-10-2017	DBC1	...
32	1003	Surgery (Srg)	26-10-2017	DBC1	...
...

Definition 1 (Universes). We write the following notations for universes: \mathcal{E} denotes the universe of unique events, i.e., the set of all possible event identifiers. U denotes the set of all possible attribute names. Val denotes the set of all possible attribute values. $Act \subset Val$ denotes the set of all possible activity names. $PI \subset Val$ denotes the set of all possible process instance identifiers.

Definition 2 (Event, Attribute, Label). For each event $e \in \mathcal{E}$, for each attribute name $d \in U$, the attribute function $\pi_d(e)$ returns the value of attribute d of event e . A labeling function $\pi_l : \mathcal{E} \rightarrow Val$ is a function that assigns the label to each event $e \in \mathcal{E}$.

If the value is undefined, $\pi_d(e) = \perp$. Examples of attribute names used in this paper are listed as follows: $\pi_{pi}(e) \in PI$ denotes the process instance identifier of e ; $\pi_{act}(e) \in Act$ is the activity associated with e ; $\pi_{time}(e)$ denotes the timestamp of e ; $\pi_{dbc}(e)$ denotes the diagnosis code of e . For example, given the log listed in Table 1, $\pi_{pi}(e_1) = 1001$, $\pi_{act}(e_1) = \text{Registration}$, $\pi_{dbc}(e_1) = \text{DBC1}$.

The labeling function $\pi_l(e)$ returns the activity label of event e in the process (also known as an *event classifier*). In this paper, we combine both the activities and the diagnosis codes and use them as labels, i.e., $\pi_l(e) := \pi_{act}(e) + \pi_{dbc}(e)$, because the data analyst from the hospital indicated that both are important for the clinical pathway. For example, given the log listed in Table 1, the label of event 1 is $\pi_l(e_1) := \pi_{act}(e_1) + \pi_{dbc}(e_1) = \text{“Reg-DBC1”}$; the label of event 23 is $\pi_l(e_{23}) = \text{“Srg-DBC4”}$.

A trace $\sigma = \langle e_1, e_2, \dots, e_n \rangle \in \mathcal{E}^*$ is a sequence of events, where for $1 \leq i < n$, $\pi_{time}(e_i) \leq \pi_{time}(e_{i+1})$ and $\pi_{pi}(e_i) = \pi_{pi}(e_{i+1})$. An event log $L = \{\sigma_1, \dots, \sigma_{|L|}\} \subseteq \mathcal{E}^*$ is a set of traces.

Definition 3 (Simplified Trace, Simplified Log). Let π_l be the labeling function. Let $L = \{\sigma_1, \dots, \sigma_{|L|}\}$ be a log and $\sigma = \langle e_1, e_2, \dots, e_{|\sigma|} \rangle$ a trace. We

overload the labeling function such that, given σ , the labeling function returns the sequence of labels of the events in σ , i.e., $\pi_l(\sigma) = \langle \pi_l(e_1), \pi_l(e_2), \dots, \pi_l(e_{|\sigma|}) \rangle \in \text{Val}^*$. Furthermore, given the log L , the labeling function returns the multi-set of the sequences of the labels of the traces in L , i.e., $\pi_l(L) = [\pi_l(\sigma_1), \dots, \pi_l(\sigma_{|L|})]$.

Let $\sigma = \langle e_1, \dots, e_n \rangle \in L$ be a trace. For $1 \leq i < n$, we say event e_i is *directly-followed* by e_{i+1} . For $1 \leq i < j \leq n$, we say event e_i is *eventually-followed* by e_j . For the sake of brevity, we write $L^l = \pi_l(L)$ and $\sigma^l = \pi_l(\sigma)$. For instance, the simplified trace of patient 1001 listed in Table 1 is $\pi_l(\sigma_{1001}) = \sigma_{1001}^{\text{act,dbc}} = \langle \text{“Reg-DBC1”}, \text{“Doc-DBC1”}, \text{“Lab-DBC1”}, \text{“Srg-DBC2”}, \text{“Doc-DBC2”} \rangle$. Note that a patient (an activity) could be associated with multiple diagnosis codes [6], e.g., patient 1001 (activity *Doc*).

3.2 Research Problem - Grouping Patients

Traditional trace clustering aims to divide the traces of a log into clusters, such that the traces of the same cluster show more homogenous behavior than the traces of different clusters. In the healthcare domain, we are facing a very large, complex data set and abundant domain knowledge. As discussed at the end of Sect. 2, we would like to (1) handle such a large data set, to (2) incorporate, leverage, and put more emphasis on the domain knowledge, in order to obtain clusters that are more in line with those of medical experts, while requiring little effort from such experts, and to (3) be able to find the clusters accurately and validate clusters quality, we propose the following.

We assume that medical experts can provide a small sample set P of the patients that belong to a patient group \hat{C} of interest. Giving a sample requires little effort from their side. We assume that \hat{C} is unknown (because when \hat{C} gets large, it would require too much effort for medical experts to exhaustively list all patients that belong to \hat{C} and to repeat this process). We use the available traces of all patients in the sample P , and the objective is to find a cluster C in such a way that C is as close to the group \hat{C} as possible (i.e., the highest recall and precision possible). We do this separately for each group \hat{C}_i where the sample set P_i is available. To generalize, we define the partial trace clustering formally as follows.

Definition 4 (Partial Trace Clustering). Let $L = \{\sigma_1, \dots, \sigma_n\}$ be the event log, and $PI = \{\pi_{p_i}(\sigma_1), \dots, \pi_{p_i}(\sigma_n)\}$ the set of case ids of L . Let $P_1, P_2, \dots, P_x \subset PI$ be the sets of samples that respectively belong to clusters $\hat{C}_1, \hat{C}_2, \dots, \hat{C}_x$, provided by experts (e.g., a doctor), with $x \in \mathbb{N}$. We would like to find the clusters $C_1, C_2, \dots, C_x \subset PI$, such that the set difference between C_i and \hat{C}_i is minimized.

Note that clusters C_1, \dots, C_x can be non-overlapping or form an incomplete clustering of PI (i.e., $C_1 \cup \dots \cup C_x \subseteq PI$), and x could be 1. Based on these properties, we do not have to find all clusters or to compute a complete clustering of all traces. It allows us to mine, cluster, and validate each cluster independently.

4 Approach

As explained above, we assume that for each cluster C to be found we have a small sample P of the cases that belongs to the true-but-unknown cluster \hat{C} . For all other cases it is unknown whether they belong to the cluster or not. By exploiting the available sample set P and the event log L of all cases, the objective is to find *behavioral criteria* for determining the cluster. To find the behavioral criteria and to handle the large number of features, we compute frequent *behavioral patterns*. In Sect. 4.1, we first explain the use of sequence pattern mining to learn the frequent sequence patterns (FSPs) of the sample set. In Sect. 4.2, we then match the FSPs to the other cases in the sample to train our parameters. Finally, we match all cases to the clustering criteria to return the computed cluster in Sect. 4.3. Figure 2 shows an overview of the approach.

4.1 Finding Frequent Sequence Patterns

The first step of the approach is to find frequent sequence patterns repeated among the samples. A *frequent sequence pattern* is a sequence that occurs in the traces with a frequency no less than a specified threshold. We adapt the definition of sequence patterns in our context as follows.

Definition 5 (Sequence Pattern). *A sequence pattern $\mathbf{s} = \langle a_1, \dots, a_m \rangle \in Val^*$ is a sequence of labels in which a_i is said to be eventually followed by a_{i+1} for $1 \leq i < m$.*

When a trace *matches* a sequence pattern, it means that the trace contains a sub sequence where the labels occur in the same order.

Definition 6 (Support of Sequence Pattern). *Let L be an event log and π_l the labeling function. Let $\sigma \in L$ be a trace, with $\pi_l(\sigma) = \langle a_1, a_2, \dots, a_n \rangle$. Let $\mathbf{s} = \langle s_1, \dots, s_m \rangle \in Val^*$ be a sequence pattern. We say σ matches \mathbf{s} if and only if there exist integers i_1, i_2, \dots, i_m such that $1 \leq i_1 < i_2 < \dots < i_m \leq n$ and $s_1 = a_{i_1}, s_2 = a_{i_2}, \dots, s_m = a_{i_m}$. We write $\mathbf{s} \sqsubseteq \pi_l(\sigma)$.*

The support of sequence \mathbf{s} in L is the number of traces in L that matches \mathbf{s} , i.e.,

$$supp(\mathbf{s}, L) = \frac{|\{\mathbf{s} \sqsubseteq \pi_l(\sigma) \mid \sigma \in L\}|}{|L|}$$

Let ϕ_s denote the minimum support threshold. A sequence pattern \mathbf{s} is said to be *frequent* if and only if $supp(\mathbf{s}, L) \geq \phi_s$. We write $SP(L, \phi_s)$ to denote the set of all sequence patterns in L with a support of at least ϕ_s , i.e.,

$$SP(L, \phi_s) = \{\mathbf{s} \in Val^* \mid supp(\mathbf{s}, L) \geq \phi_s\}$$

Step 1 in Fig. 2 exemplifies mining frequent sequence patterns, with the minimum support $\phi_s = 0.8$. Let $L' = \{\sigma_1, \sigma_2, \sigma_3\}$, as shown in Fig. 2. We have $SP(L', 0.8) = \{\langle A \rangle, \langle C \rangle, \langle D \rangle, \langle E \rangle, \langle F \rangle, \langle A, C \rangle, \langle A, D \rangle, \langle A, E \rangle, \dots, \langle A, C, D, F \rangle\}$.

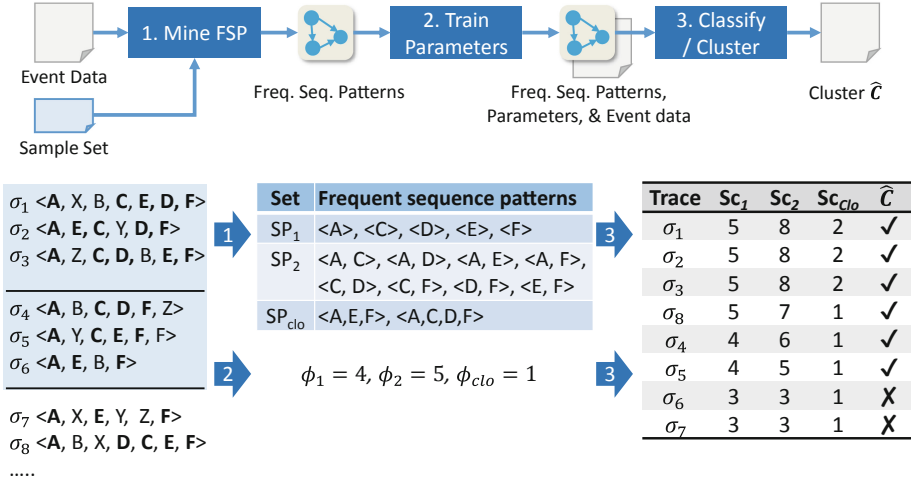


Fig. 2. An example of our approach applied on the event log (on the left), with the FSPs mined (in the middle), and the scores of the traces (on the right).

There are several well-known algorithms to compute frequent sequence patterns. In this paper, we use the CloFAST algorithm [14] and its SPMF implementation [15] due to its fast run-time, which is also used in [16] for next activity prediction.

4.2 Trace Ranking by Sequence Pattern Matching

To automatically find behavioral criteria for determining the cluster C , we divide the sample set P into a training set P_{tr} and use the entire P as our test set. On the training set P_{tr} , we compute the set of frequent sequence patterns (FSPs).

The FSPs mined on the training set P_{tr} could still be very large. Therefore, we select a subset of the FSPs. We use the FSPs of length 1, 2, and the closed sequence patterns as our behavioral criteria. Note that this step can be generalized with ease and any other subset of the FSPs can be selected as behavioral criteria. We select these three subsets because of the following. The FSPs of length 1 represent the frequent activity labels occurred in the training set; the FSPs of length 2 represent the frequent *eventually-followed* relations occurred. The *closed sequence patterns* $s \in SP(L, \phi_s)$ are the sequence patterns s such that for all other patterns which s satisfies have a lower support. Thus, these three provide a good coverage of all FSPs with less redundancy.

Definition 7 (Closed Sequence Pattern). Let $SP(L, \phi_s)$ denote all frequent sequence patterns in L with a support of at least ϕ_s . A sequence pattern $s \in SP(L, \phi_s)$ is a closed sequence pattern if and only if for all $s' \in SP(L, \phi_s)$, $s \sqsubseteq s' \Leftrightarrow (s = s' \vee supp(s, L) > supp(s', L))$.

Next, using these subsets of these patterns, we rank each trace in P based on the number of patterns it satisfies. Let $SP(P, \phi_s) = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$ be the set of frequent sequence patterns of P above support threshold ϕ_s . Let $SP_1, SP_2, SP_{clo} \subseteq SP(P, \phi_s)$ be the set of patterns of size 1, size 2, and closed sequence patterns, respectively. We give each case a score based on the number of patterns in SP_1, SP_2 , and SP_{clo} the trace satisfies and rank the cases based on their score. Thus,

$$score_k(\sigma) = |\{\mathbf{s} \in SP_k | \mathbf{s} \sqsubseteq \pi_l(\sigma)\}|$$

For example, see Fig. 2, step 1 shows SP_1 of five sequence patterns, SP_2 of eight, and SP_{clo} of two, which are mined on the $P_{tr} = \{\sigma_1, \sigma_2, \sigma_3\}$ using a minimum support of 0.8. Given trace $\sigma_4 \notin P_{tr}$, it matches to $\langle A \rangle, \langle C \rangle, \langle D \rangle$, and $\langle E \rangle$ in SP_1 , to $\langle A, C \rangle, \langle A, D \rangle, \langle A, F \rangle, \langle C, D \rangle, \langle C, F \rangle$, and $\langle D, F \rangle$ in SP_2 , and to $\langle A, C, D, F \rangle$ in SP_{clo} . Thus, $score_1(\sigma_4) = 4$, $score_2(\sigma_4) = 6$, $score_{clo}(\sigma_4) = 1$.

4.3 Computing Criteria Threshold

For each case, we have now computed $score_1$, $score_2$, and $score_{clo}$, as explained above. For the three scores, we respectively introduce three thresholds, $\phi_1 \in \mathbb{N}$, $\phi_2 \in \mathbb{N}_0$, and $\phi_{clo} \in \mathbb{N}_0$. We decide on whether a case belongs to cluster C based on whether the scores of the trace are above the corresponding thresholds, i.e.,

$$C_{\phi_1, \phi_2, \phi_{clo}} = \{\pi_{pi}(\sigma) | \sigma \in L \wedge score_1(\sigma) \geq \phi_1 \wedge score_2(\sigma) \geq \phi_2 \wedge score_{clo}(\sigma) \geq \phi_{clo}\}$$

To estimate the quality of $C_{\phi_1, \phi_2, \phi_{clo}}$, we then compute the estimated recall with respect to P , i.e., $\overline{recall}_{\phi_1, \phi_2, \phi_{clo}} = \frac{|C_{\phi_1, \phi_2, \phi_{clo}} \cap P|}{|P|}$. When the sample set P gets closer to the ideal cluster \hat{C} , the estimated \overline{recall} gets closer to the true recall. When we decrease ϕ_1, ϕ_2 , and ϕ_{clo} , more cases are included in C . After a certain point, the increase of \overline{recall} starts to flatten, which suggests that further lowering the thresholds does not help to retrieve a large number of true positive cases, which is likely to result in a low precision. To approximate such a point, we use $\max_{\phi_1, \phi_2, \phi_{clo}} \frac{\overline{recall}_{\phi_1, \phi_2, \phi_{clo}}^2}{|C_{\phi_1, \phi_2, \phi_{clo}}|}$ [17], but only consider the thresholds when $\overline{recall} \geq 0.8$. The number of iterations to find such a maximum depends on the maximal values of $score_1$, $score_2$, and $score_{clo}$.

5 Evaluation

We implemented the described approach in the process mining toolkit ProM. We used a real-life data set to evaluate our approach with respect to the following three objectives:

- (EO1) How accurate (in terms of F1-scores) are the clusters returned by our automated approach, compared to the optimal scores?

Table 2. General information of the real-life data set and the ground truth clusters.

Data	#cases	#dpi	#avg. c/dpi	#events	#avg. e/c	#max. e/c	#acts	#dbcs	#dst. labels	Perc. of all
All17	128,505	97,771	1.3	$3.70 * 10^6$	28.8	2,924	4,666	1,915	150,244	-
$\hat{C}_{Kidney17}$	140	140	1.0	40,071	286.2	2,167	676	237	4,777	0.11%
$\hat{C}_{Diabetes17}$	1,521	1,520	1.0	139,454	91.7	2,861	1,414	646	16,496	1.18%
$\hat{C}_{HNTumor17}$	1,050	1,048	1.0	105,613	100.6	905	1,001	380	9,211	0.82%
...										
All13	133,438	99,196	1.3	$4.32 * 10^6$	32.4	2,558	4,871	1,813	168,096	-
$\hat{C}_{Kidney13}$	81	81	1.0	26,949	332.7	1,577	651	159	3,601	0.06%
$\hat{C}_{Diabetes13}$	1,573	1,573	1.0	142,737	90.7	1,057	1,427	663	16,966	1.18%
$\hat{C}_{HNTumor13}$	1,350	1,334	1.0	147,491	109.3	2,227	1,237	437	12,678	1.01%

- (EO2) How accurate can we find the clusters using our approach, compared to a related approach that uses frequent item sets (FIS) [7]?
- (EO3) Can we discover a simple and insightful behavioral criteria for each patient group such that the criteria can be used to communicate with medical experts?

In the following, we first discuss the data set in Sect. 5.1 and then report our results Sect. 5.2 with respect to these three objectives. All experiments are run on an Intel Core i7- 8550U 1.80 GHZ with a processing unit of 16 GB running Windows 10 Enterprise. The maximal queue size of CloFAST algorithm [14, 15] is set to 10^5 . The obtained results were discussed with the semi-medical expert and the data expert in the hospital who cooperate closely with medical experts in their daily work.

5.1 Experimental Setup

For the evaluation, we used anonymized patient records provided by the VU University Medical Center Amsterdam, a large academic hospital in the Netherlands. All patients that have a diagnosis code registered between 2013 and 2017 are selected. The administrative and dummy activities are filtered out. As a result, we have in total 328,256 patients over the five years. There are 7,426 unique activities and 2,251 unique diagnosis codes registered. In total more than 15.5 million events are recorded in the logs.

In addition, lists of patients of three groups divided over the five years are provided by the analyst, patients with kidney failure, with diabetes, or with head/neck-tumor. We use $\hat{C}_{KidneyYY}$, $\hat{C}_{DiabetesYY}$, and $\hat{C}_{HNTumorYY}$ to refer to them, respectively, where YY denotes the particular year. Table 2 lists the number of cases (c), distinct process instances (dpi), events (e), activities (acts), and other statistical information related to the event logs of 2013 and 2017 as examples. For instance, in Table 2 row 2, 3, and 4 show an overview of $\hat{C}_{Kidney17}$, $\hat{C}_{Diabetes17}$, and $\hat{C}_{HNTumor17}$, respectively. These 15 clusters are used as the ground truth. For each cluster, 30 patient ids are provided by medical experts as

the sample set (i.e., $|P| = 30$), the same as a previous study [7]. For finding the clusters, we use all the patient records of the same year and the provided P to compute our cluster C . The quality of C is evaluated against the corresponding ground truth cluster \hat{C} by calculating the recall, precision, and F1-score, i.e., $recall(C, \hat{C}) = \frac{|C \cap \hat{C}|}{|\hat{C}|}$, $precision(C, \hat{C}) = \frac{|C \cap \hat{C}|}{|C|}$, and $F1_measure(C, \hat{C}) = 2 \cdot \frac{precision(C, \hat{C}) \cdot recall(C, \hat{C})}{precision(C, \hat{C}) + recall(C, \hat{C})}$.

It is worthwhile to mention that the data analyst stated that it took a lot of time and effort to obtain each of these ground truth clusters. Multiple intensive discussion sessions were scheduled with different groups of medical experts to come to the definitions and criteria for each of these clusters. This suggests that if our algorithmic approach can identify the behavioral criteria and the clusters with a reasonable accuracy using only a small sample set, it would help reducing the workload of both the analysts and the medical experts and making this process feasible to be repeated for other patient clusters or in other hospitals. Another important remark is that the ground-truth clusters which we are trying to find are very small and unbalanced compared to the full event logs, making this trace clustering problem a very challenging task. For instance, the $\hat{C}_{Kidney17}$ ($\hat{C}_{Diabetes17}$) contains only 140 (1521) patients, about 0.1% (1.2%) of the 128,505 patients in the log of 2017.

5.2 Results

(EO1) F1 Scores of Automated Approach Compared To Maximum.

To determine the support threshold ϕ_s , we started with 1.0 and decreased the value by 0.1 until a reasonable large amount of patterns are found and the F1 scores stopped increasing. For the C_{Kidney} groups, ϕ_s ranges from 1.0 down-to 0.6, for $C_{Diabetes}$, 0.4 down-to 0.2, and for $C_{HNTumor}$, 0.5 down-to 0.2. We used either 10 or 15 (of the 30 in the sample) as the training set to learn the frequent sequential patterns, i.e., $k = |P_{tr}| \in \{10, 15\}$. We write TC-FSMa for our approach with the automatically determined ϕ_1 , ϕ_2 , and ϕ_{clo} ; TC-FSM* for the maximum F1 score using the same ϕ_s and k but based on the optimal ϕ_1 , ϕ_2 , and ϕ_{clo} .

Figure 3 shows the difference in F1-scores between TC-FSMa (dotted lines) and TC-FSM* (filled lines). We observe that in most cases the F1-scores of the automated TC-FSMa (dotted line) are very close to the ones of the optimal TC-FSM* (filled line). For some clusters, for example Diabetes16&17 and HNTumor15&17, TC-FSMa returns the exact same F1-scores as the maximum for all ϕ_s and k . Only in a few cases, for example, for Diabetes13 when k is 15 and the support ϕ_s is 0.2, TC-FSMa scores considerably lower than TC-FSM* with a difference of 0.26. Nevertheless, for the same ϕ_s when k is set to 10, this difference is immediately decreased to 0.01. Taking into account that we only have a sample set of 30 patients and the number of activities ranges in the thousands, the TC-FSMa is able to approximate the optimal F1-scores very well.

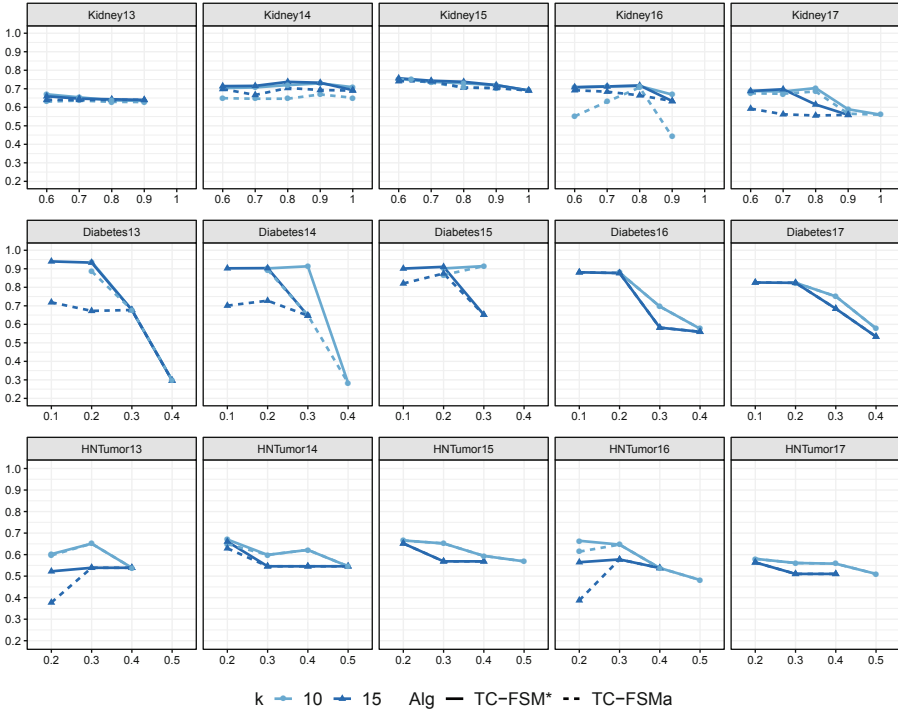


Fig. 3. The differences in the F1-scores of the automated approach (TC-FSMa) shown in dotted lines and the maximum scores achieved (TC-FSM*) shown in filled lines, using various support threshold (on the x-axis) and training sample size k (Color figure online).

For the support threshold ϕ_s , we observe overall a slight increase in the F1-scores for the Kidney and HNTumor groups when we decrease ϕ_s . For the Diabetes groups, there is a considerable increase in F1-scores during the beginning (when ϕ_s is decreased from 0.4 to 0.2), but this improvement also fades out. One reason for this is because when the support threshold is low, more patterns are found and used as criteria; thus, more patients are included in the cluster including false positives. While the recall increases, the precision becomes lower, which led to a small increase in the F1-scores. For the diabetes group, when the support ϕ_s is 0.4, the number of sequence patterns is extremely small (1 or 3). When the support decreases, it allowed the algorithm to find a consider number of defining patterns that is significant to retrieve the patients of the ground truth clusters. This increases the recall dramatically without a significant decrease in precision. Furthermore, Fig. 3 also shows that using fewer training samples ($k = 10$, denoted using light blue), our approach can achieve the same scores as when using a larger training sample set ($k = 15$, denoted using darker blue). In many cases, the former (i.e., $k = 10$) even achieved a better result. This may be due to that the training test set $P \setminus P_{tr}$ is larger.

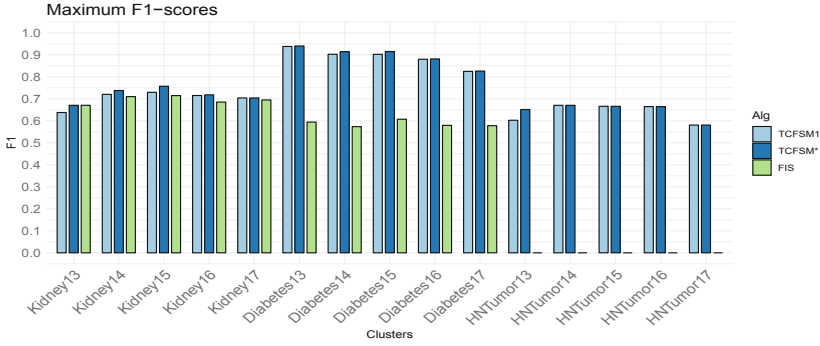


Fig. 4. F1-measures achieved by our approaches TC-FSM1 and TC-FSM* for the three groups, compared to the ones achieved by the previous approach FIS [7].

(EO2) Comparing the F1-Scores Achieved. We write TC-FSM* to refer to the maximum score of our approach using the above settings. We write TC-FSM1 to denote our approach with a single setting (0.8 for kidney, 0.2 for diabetes, and 0.2 for NH-tumor, with sample size 30 and $k_{tr} = 10$), to compare our results to the previous work [7]. The parameters are selected on the results of EO1. We write FIS for referring to the previous approach that uses frequent item sets [7].

Figure 4 shows the maximum F1-scores of TC-FSM*, TC-FSM1, and FIS on the 15 clusters over 5 years and three groups. For the diabetes group, we achieved a considerable improvement of 0.2–0.3 in the F1-scores, compared to the FIS approach [7]. Overall, our approach achieved a better result. One reason for this improvement is with the use of sequential patterns (instead of frequent item sets), our approach is able to decrease the number of false positives and find the clusters with a higher precision.

(EO3) Frequent Sequence Patterns to Simple Process Maps. We used the closed sequence patterns mined on the samples as the traces that represent the frequent behavior shared by the group. Using these sequence patterns, the discovered process maps overall seem to be simple and insightful, representing only the crucial behavioral criteria of the patient groups. We show three process maps in Figs. 5, 6 and 7, for $C_{kidney17}$, $C_{HNTumor16}$, and $C_{Diabetes16}$ to illustrate our results. All the process maps contain all activities and paths (thus no filters applied). As can be seen in Fig. 5, the number of activity labels in the process is reduced from about 4,700 to 11. The number of distinct variants is reduced from 140 to 8.

The process maps are shown to the semi-medical expert and the data analyst. The semi-medical expert observes and confirms that the activities shown (e.g., “*kalium*” (potassium), “*kreatinine*” (creatinine), “*calcium*” (calcium), “*fosfaat*” (phosphate), “*albumine*” (albumin), “*natrium*” (sodium), “*ureum bloed*” (ureum blood), etc.) are important activities (e.g., lab activities) in the clinical pathway of the kidney groups (patients with renal insufficiency). The data analyst

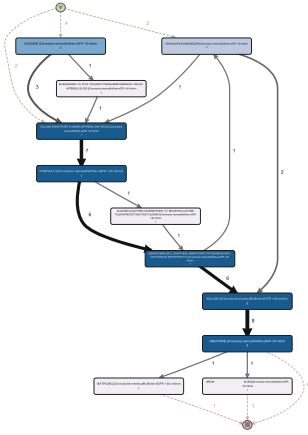


Fig. 5. The full process map based on the closed FSPs of the kidney group; 600 activities are reduced to 11 labels.

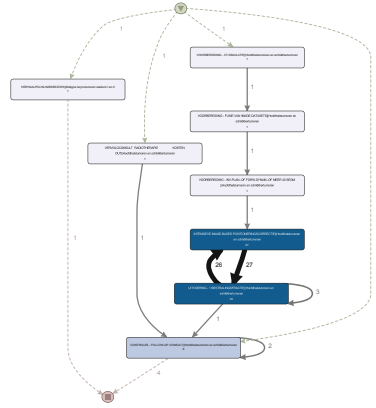


Fig. 6. The full process map based on the closed FSPs of the HNTumor group.

confirms that the diagnosis code “Chronic renal failure eGFR < 30 ml/min” associated with these activities is a crucial criteria for defining the kidney groups.

In Figs. 6 and 7, we also observe that multiple distinct diagnosis codes are used for the HNTumor and diabetes group, respectively. In the process map for $C_{Diabetes16}$, we found the process map being divided into three sub processes based on the diagnosis codes: [SG1] “diabetes mellitus without secondary complications”, [SG2] “diabetes mellitus with secondary complications”, and [SG3] “diabetes mellitus chronic pump therapy” (see Fig. 7, highlighted in red).

The semi-medical expert also observes and confirms that some of these activities are important indicators for different groups. For example, “creatinine” is important for both the kidney and diabetes groups. Nevertheless, because our approach is able to combine and handle the activities with their diagnosis codes as activity labels (in terms of the large variety of distinct labels and process variants), it enabled us to accurately distinguish the “creatinine” for the kidney group (i.e., “creatinine||Chronic renal failure eGFR < 30 ml/min”) versus the same “creatinine” but for the diabetes group (i.e., “creatinine||SG1” and “creatinine||SG3”, see Fig. 7, highlighted in blue).

5.3 Discussion

The results have shown that our approach using the discovered and selected frequent sequence patterns can help to cluster the patient groups with a reasonably high accuracy (e.g., a maximum of 0.75 for the Kidney group, 0.94 for Diabetes, and 0.67 for HNTumor), despite the very large data sets (on average, about 130,000 of patients and 3.9 millions of events per year) and the relatively

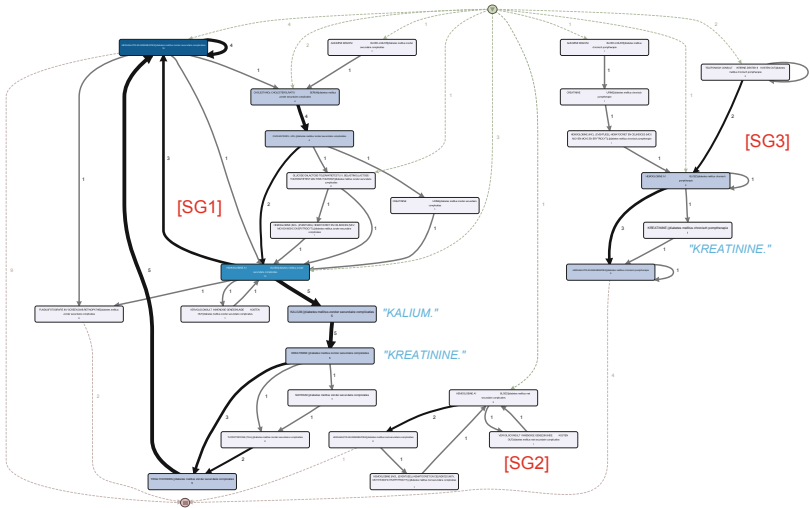


Fig. 7. The full process map based on the closed FSPs of the diabetes group; three distinct subgroups SG1, SG2, and SG3 are found (Color figure online).

very small and unbalanced clusters. Moreover, the proposed approach that automatically determines the parameters achieved F1-scores that are very close to the optimal scores. This means after setting the support ϕ_s and with no further input, we can find the clusters with a reasonable quality as well. Using the process maps, we show the meaningful and insightful behavioral patterns and criteria in the clinical pathways of the patient groups. According to the semi-medical expert, the maps can be a useful tool in the communication with the domain experts regarding the pathways. Note that we do not have any prior knowledge of the specific activities or diagnosis codes of the patient groups.

A remark is that these process maps of FSPs have a different semantics than the formal process models discovered using the traces. For example, an edge from A to B in the map, in essence, means that such an *eventually-followed* relation is frequent. To obtain formal models, we may project the patterns on the traces and use the instances of the patterns in the traces to discover models [18].

6 Conclusion and Future Work

In this paper, we investigated the trace clustering problem in healthcare and proposed an approach that can handle the characteristics of healthcare data. Using a small sample set of patients, the proposed approach finds frequent sequence patterns and uses these as behavioral criteria for determining a cluster. The results of the evaluations show that the approach is able to identify patient clusters with a very reasonable quality on the basis of very limited input from medical experts, despite the very large data sets and the small, unbalanced clusters (ground-truth). The obtained behavioral criteria also led to the generation

of simple process maps, where we have some first insights that these could be actually used by medical experts. The semi-medical expert who works closely with medical experts was able to recognize the important activities in the clinical pathways of the patient groups. Such a method may be useful to reason about clinical pathways within hospitals for the sake of process improvement or quality control.

For future work, we plan to investigate other strategies for selecting sequence patterns as behavioral criteria to further improve the F1-scores. Also, the effect of sample size on the F1-scores is worth investigating. Another interesting direction is to exploit the frequent sequence patterns to discover formal process models for the clinical pathways of each cluster. Finally, we would like to validate the maps with medical experts and apply our approach to other patient clusters and other hospitals.

Acknowledgments. This research was supported by the NWO TACTICS project (628.011.004) and Lunet Zorg in the Netherlands. We would also like to thank the experts from the VUMC for their extremely valuable assistance and feedback in the evaluation.

References

1. Rojas, E., Munoz-Gama, J., Sepúlveda, M., Capurro, D.: Process mining in health-care: a literature review. *J. Biomed. Inform.* **61**, 224–236 (2016)
2. Caron, F., Vanthienen, J., Vanhaecht, K., van Limbergen, E., De Weerd, J., Baesens, B.: Monitoring care processes in the gynecologic oncology department. *Comput. Biol. Med.* **44**, 88–96 (2014)
3. Greco, G., Guzzo, A., Pontieri, L., Saccà, D.: Discovering expressive process models by clustering log traces. *IEEE Trans. Knowl. Data Eng.* **18**(8), 1010–1027 (2006)
4. 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
5. De Weerd, J., vanden Broucke, S.K.L.M., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. *IEEE Trans. Knowl. Data Eng.* **25**(12), 2708–2720 (2013)
6. Schreyögg, J., Stargardt, T., Tiemann, O., Busse, R.: Methods to determine reimbursement rates for diagnosis related groups (DRG): a comparison of nine european countries. *Health Care Manag. Sci.* **9**(3), 215–223 (2006)
7. Tabatabaei, S.A., Lu, X., Hoogendoorn, M., Reijers, H.A.: Identifying patient groups based on frequent patterns of patient samples. *CoRR* abs/1904.01863 (2019)
8. Bose, R.P.J.C., van der Aalst, W.M.P.: Trace clustering based on conserved patterns: towards achieving better process models. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) *BPM 2009*. LNBIP, vol. 43, pp. 170–181. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_16
9. Bose, R.P.J.C., van der Aalst, W.M.P.: Context aware trace clustering: towards improving process mining results. In: *Proceedings of the SDM 2009*, pp. 401–412 (2009)
10. Chatain, T., Carmona, J., van Dongen, B.: Alignment-based trace clustering. In: Mayr, H.C., Guizzardi, G., Ma, H., Pastor, O. (eds.) *ER 2017*. LNCS, vol. 10650, pp. 295–308. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69904-2_24

11. Cadez, I.V., Heckerman, D., Meek, C., Smyth, P., White, S.: Model-based clustering and visualization of navigation patterns on a web site. *Data Min. Knowl. Discov.* **7**(4), 399–424 (2003)
12. Ferreira, D., Zacarias, M., Malheiros, M., Ferreira, P.: Approaching process mining with sequence clustering: experiments and findings. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 360–374. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_26
13. De Koninck, P., Nelissen, K., Baesens, B., vanden Broucke, S., Snoeck, M., De Weerd, J.: An approach for incorporating expert knowledge in trace clustering. In: Dubois, E., Pohl, K. (eds.) *CAiSE 2017*. LNCS, vol. 10253, pp. 561–576. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_35
14. Fumarola, F., Lanotte, P.F., Ceci, M., Malerba, D.: CloFAST: closed sequential pattern mining using sparse and vertical id-lists. *Knowl. Inf. Syst.* **48**(2), 429–463 (2016)
15. Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani, A., Wu, C., Tseng, V.S.: SPMF: a Java open-source pattern mining library. *J. Mach. Learn. Res.* **15**(1), 3389–3393 (2014)
16. Ceci, M., Spagnoletta, M., Lanotte, P.F., Malerba, D.: Distributed learning of process models for next activity prediction. In: *IDEAS*, pp. 278–282. ACM (2018)
17. Lee, W.S., Liu, B.: Learning with positive and unlabeled examples using weighted logistic regression. In: *ICML*, vol. 3, pp. 448–455 (2003)
18. Lu, X., et al.: Semi-supervised log pattern detection and exploration using event concurrence and contextual information. In: Panetto, H., et al. (eds.) *OTM 2017*. LNCS, vol. 10573, pp. 154–174. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69462-7_11



ProcessExplorer: Intelligent Process Mining Guidance

Alexander Seeliger^(✉), Alejandro Sánchez Guinea, Timo Nolle^{},
and Max Mühlhäuser

Telecooperation Lab, Technische Universität Darmstadt, Darmstadt, Germany
{seeliger,sanchez,nolle,max}@tk.tu-darmstadt.de

Abstract. Large amount of data is collected in event logs from information systems, reflecting the actual execution of business processes. Due to the highly competitive pressure in the market, organizations are particularly interested in optimizing their processes. Process mining enables the extraction of valuable knowledge from event logs, such as deviations, bottlenecks, and anomalies. Due to the increase of process complexity in flexible environments, visual exploration is increasingly becoming more challenging. In this paper, we propose ProcessExplorer, an interactive process mining approach to enable fast data analysis and exploration. ProcessExplorer takes an event log as input to automatically suggest subsets of similar process behavior, evaluate each subset, generate interesting insights, and suggest the subsets with the most interesting characteristics. We implemented our approach into an interactive visual exploration system, which we use as part of a user study conducted to evaluate our approach. Our results show that ProcessExplorer can be successfully applied to analyze and explore real-life data sets efficiently.

Keywords: Process mining · Interactive data exploration · Data analytics · Business intelligence · Recommender system

1 Introduction

Information systems in organizations support and automate the processing of business transactions. Typically, these systems are highly integrated into companies' business processes. Each executed activity in an information system is recorded in an event log, storing information about what activity was executed at a certain point in time. Due to the highly competitive pressure in the market, organizations are strongly interested in analyzing these event logs to optimize their business processes. Process mining provides an accurate view on how processes are actually executed. Specifically, process discovery helps analysts to visually understand the underlying relationship between activities, find potential bottlenecks, and compare the actual execution with the desired one.

Due to the growth of available event data and the increase of process complexity, finding interesting insights is more challenging. Often extensive domain

knowledge is required to analyze highly complex process models (e.g., spaghetti-like models) to obtain valuable insights [16]. Efforts, such as the PM^2 or the L^* methodology, have been made to systematize the workflow of analysts to guide the planning and execution of process mining projects. Despite the increasing number of available tools that incorporate process mining capabilities, existing tools lack proper computer-assisted support. In practice, the work of analysts is characterized as largely manual, leading to many ad-hoc tasks.

Let us consider the following scenario, which exemplifies a common situation an analyst may encounter today. *Julia is an analyst who is interested in the process performance of the BPIC 2019 event log. The process discovery returns a spaghetti-like process model which reflects the actual behavior of the process. To simplify the view, Julia manually filters cases based on domain knowledge. For instance, she selects cases that start with a requisition and are of item type “subcontracting”. Afterwards, she computes the case duration of the subset and compares it with the case duration over all cases. For this particular selection, the case duration turns out to be significantly lower with 31.8d compared to 71.5d. Next, Julia considers a different selection and computes the case duration again.* As we can see, this repetitive work is time-consuming and error-prone, hampering efficient exploration and analysis.

In this paper, we present ProcessExplorer, a novel approach that provides automatic guidance during process discovery. Our work is inspired by the typical workflow of analysts, intended to obtain a general overview of the different process behaviors and their performance observed in the event log. ProcessExplorer recommends subsets of interesting cases that follow a certain behavior pattern. Based on the identified subsets, ProcessExplorer recommends insightful process performance indicators (PPIs) that characterize each subset. As part of our work, we developed an interactive visual exploration system that implements the underlying techniques of our approach. Our system does not make any assumptions about the process or the event log, which makes it specifically useful for exploring unknown processes. We evaluated the usefulness of our system by conducting a user study with process analysts (Sect. 6). The results show that both types of recommendations are valuable and useful for analysts, specifically for event logs with a large set of events.

In summary, our contributions are:

1. A novel approach to automatically compute recommendations that guide process analysts during the analysis of large event logs (Sect. 4). Specifically, we introduce a number of innovative techniques:
 - An adapted version of trace clustering capable of automatically generating subset recommendations of cases with interesting process behavior, related to the control-flow and data perspectives (Sect. 4.1).
 - A mechanism, based on statistical significance analysis, that identifies the most deviating PPIs that are relevant and insightful for the analyst to explore (Sect. 4.2).
 - A diversifying top-k ranking approach that generates a ranked list of subset and insights recommendations (Sect. 4.3).

2. An interactive visual exploration system that enhances analytic support to quickly explore large event logs (Sect. 5).

The rest of the paper is organized as follows. In Sect. 2 we provide an overview of the related work, followed in Sect. 3 by some preliminaries used throughout the paper. In Sect. 4 we describe the details of our approach. Then, in Sect. 5 an implementation system of our approach is presented, which is used for evaluating the usefulness of our approach, as described in Sect. 6. In Sect. 7 we present a brief discussion on the validity of our evaluation and limitations of our approach. Finally, in Sect. 8 we conclude the paper with a discussion and future work.

2 Related Work

Exploratory Data Analysis. Our work is highly related to exploratory data analysis (EDA), which deal with the issue that users do not know beforehand the characteristics of the data set. One direction to support is to provide the user with automatic visualization recommendations during the analysis. SEEDB [24] evaluates the interestingness of columns from relational data to obtain such recommendations. Voyager [26] suggests data charts according to statistical and perceptual measures, and provides faceted browsing. Both approaches focus on the selection of data attributes and aggregations. DeepEye [14] suggests which type of visualization yields the best data representation. VizRec [18] additionally provides personalized visualizations based on the user’s perception. Our approach supports the visual exploration of process executions in event data.

Data Insights. Another direction of exploratory data analysis is to automatically provide interesting insights rather than exploring data dimensions or encodings. Foresight [7] recommends visual insights by analyzing the statistical properties (e.g., correlations between data attributes). DBExplorer [22] aims to improve the understanding of the characteristics of the data attributes and querying of hidden attributes by using conditional attribute dependency views. In [11] a smart interactive drill-down approach is presented which discovers and summarizes interesting data groups. Milo and Somech [17] introduce a next step recommendation engine which suggests follow-up analysis actions based on prior action recordings. In process mining, a similar analysis is applicable to case attributes and PPIs. Our approach investigates how to extract interesting case attributes and PPIs from event logs automatically.

Interactive Browsing in Process Mining. Process mining tools such as ProM, Fluxicon Disco or Celonis are highly exploratory, enabling the user to interactively inspect and analyze event logs. P-OLAP [3] enables to analyze and summarize big process graphs as well as it provides multiple views at different granularity. VIT-PLA [27] summarizes traces and automatically creates data explanations from trace attributes using regression analysis. A linguistic summary approach is presented in [8]. In [2] the authors evaluate PPIs to identify the key differences between process variants. Similarly, Bolt et al. [4] use annotated

transition systems to obtain the differences between process variants. Business rules on decision points are generated from the case attributes, revealing the influence of data values to the control-flow. However, different from our approach, existing tools lack of explicit analysis guidance.

Complexity Reduction of Event Logs. Many different approaches have been investigated to reduce the complexity of event logs such as trace or activity clustering. Smaller sub-logs that are obtained by grouping similar process behavior together tend to be less complex and easier to analyze. Clustering can be divided into alignment based [5, 25] and model-based [21, 23] approaches. An artifact-centric approach is presented in [9] where complex models are split into smaller and simpler ones with fewer states and transitions using statistical significance testing. In our approach, we build on the existing trace clustering technique to extract subsets with similar process behavior to reduce analytic complexity.

3 Preliminaries

We first provide some formal definitions, derived from [1], which are used in the exposition of our approach.

Let \mathcal{E} be the set of all possible event identifiers, \mathcal{A} be the set of attributes, and \mathcal{V}_a the set of all possible values of attribute $a \in \mathcal{A}$. For an event $e \in \mathcal{E}$ and an attribute $a \in \mathcal{A}$: $\#_a(e)$ is the value of attribute a for event e .

Let \mathcal{C} be the set of all possible case identifiers. For a case $c \in \mathcal{C}$ and an attribute $a \in \mathcal{A}$: $\#_a(c)$ is the value of attribute a for case c . Each case contains a mandatory attribute *trace*: $\#_{trace}(c) \in \mathcal{E}^*$, denoted as $\hat{c} = \#_{trace}(c)$.

A *trace* is a finite sequence of events $\sigma \in \mathcal{E}^*$ such that each event only occurs once: $1 \leq i < j \leq |\sigma| : \sigma(i) \neq \sigma(j)$. Finally, an *event log* is a set of cases $L \subseteq \mathcal{C}$ such that each event only occurs at most once in the log.

4 ProcessExplorer Approach

ProcessExplorer follows three main steps to provide intelligent guidance for event log exploration:

1. *Discovery of Subset Recommendations.* Our approach automatically discovers subsets that contain process cases of similar behavior using trace clustering based on the control-flow and the data perspective. These subsets are computed by splitting the event log into smaller and relevant subsets that are much more structured and easier to understand for analysts.
2. *Discovery of Insights Recommendations.* Our approach automatically discovers relevant PPIs and identifies the most interesting ones. The criteria to decide on the interestingness of PPIs is based on how much the subset deviates from a given reference.
3. *Ranking of Recommendations.* We rank the recommendations based on the analysis goals by applying diversifying top-k ranking. This ensures that our approach highlights most diversifying recommendations instead of showing very similar subsets or PPIs.

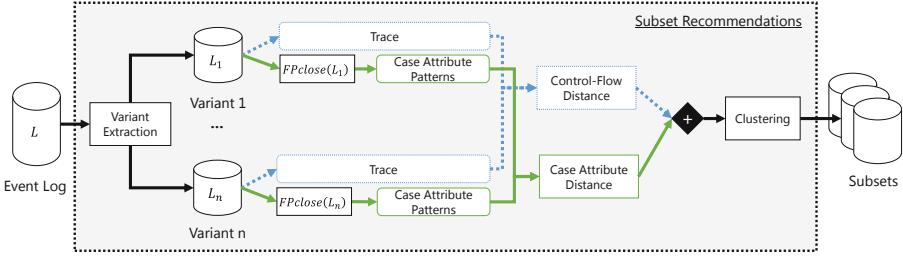


Fig. 1. Architecture of the subset recommendation generation.

Next, we provide a detailed description of each step.

4.1 Discovery of Subset Recommendations

In this step, our approach automatically finds interesting process behaviors based on a modified version of trace clustering. The event log is split into cases of similar behavior, which are then used as the basis for the subset recommendations. Subset recommendations are similar to filters typically found in existing tools, but they are generated automatically from the event log.

We adapt the trace clustering algorithm introduced in [21], which incorporates the control-flow and data perspective to generate clusters of cases. It defines a combined similarity function that balances the two perspectives. Let L be an event log, and let $L_1, L_2, \dots, L_n \subseteq L$ be the variants of L solely based on the control-flow. Furthermore, let $L_i \cap L_j = \emptyset$ for all $i, j \in [1, \dots, n]$ and $i \neq j$, i.e., cases are assigned to exactly one variant. For computing the similarity between cases, the two process perspectives are inspected separately before they are combined (see Fig. 1):

- For the *control-flow perspective*, we compute the Levenshtein edit distance $lev(\underline{\hat{x}}, \underline{\hat{y}})$ between two cases $x, y \in L$. The Levenshtein edit distance is defined as the minimum edit operation costs to modify the trace of a case to another by insertion, deletion, or substitution of activities. For comparing traces of different length we use the normalized edit distance, which is the edit costs divided by the sum of the lengths of both traces. We compute the control-flow similarity between two sets of cases using the normalized Levenshtein edit distance for all variants, where $X, Y \subseteq L$:

$$sim_C(X, Y) = \sum_{x \in X} \sum_{y \in Y} lev(\underline{\hat{x}}, \underline{\hat{y}}) / (|X| \cdot |Y|) \quad (1)$$

- For the *data perspective*, we explore the relationships between case attributes. We are particularly interested in values of case attributes that co-occur with other values. We extract the case attribute relationships by applying frequent pattern mining, which is an unsupervised data mining technique that searches for recurring patterns or relationships among itemsets. In particular, we use

the FPclose algorithm [10] on case attributes (as itemsets) to find co-occurring case attribute values. The application of the FPclose algorithm is denoted as

$$\mathcal{I}_{L_i} = \{FPclose(X, s) : X \in L_i\} \quad (2)$$

where s is the *minimum support threshold* in the range $0 \leq s \leq 1$, and \mathcal{I}_{L_i} is the set of all case attribute value patterns in L_i . The support threshold s determines the percentage of how often a pattern must be observed in the event log to be considered frequent. The FPclose algorithm only returns closed patterns, reducing thus the number of patterns considered. A pattern I is closed if there exists no proper superset J that has the same support as I . Case attribute value patterns are calculated for each variant L_i separately and mapped to the corresponding cases. We denote all case attribute patterns as $\mathcal{I} = \mathcal{I}_{L_1} \cup \dots \cup \mathcal{I}_{L_n}$. Extracted case attribute value patterns are compared with the following formula, which returns the proportion of case attribute values that are contained in both sets:

$$sim_D(I_i, I_j) = \frac{2 \cdot |I_i \cap I_j|}{|I_i| + |I_j|} \quad (3)$$

where $I_i, I_j \subseteq \mathcal{I}$.

For calculating the clusters, the two similarity functions are combined such that the influence of the control flow and the data perspective can be controlled by a weighting factor w , with $0 \leq w \leq 1$. We define a similarity matrix over the case attribute value patterns instead of cases:

$$M = (m_{ij}) = w \cdot sim_C(cases(I_i), cases(I_j)) + (1 - w) \cdot sim_D(I_i, I_j) \quad (4)$$

where $cases$ is the mapping function $\mathcal{I} \rightarrow \mathcal{C}$ that returns the set of cases of the case attribute value pattern. We use *Hierarchical Agglomerative Clustering* with ward linkage to compute the clusters for the similarity matrix M . Because we cluster case attribute value patterns, we need to obtain the corresponding cases to generate the subsets. These subsets may contain overlapping cases because similar case attribute value patterns may be shared among different clusters which may come from the same case. This issue is resolved by assigning the cases to the cluster with the minimum distance concerning the control flow.

Finally, our method automatically determines the optimal number of clusters k , the minimum support threshold s , and the weighting factor w . We formulate an optimization problem that maximizes the fitness of the underlying process models of each cluster, maximizes the silhouette coefficient of the clusters, and minimizes the number of clusters to obtain k , s , and w . The optimization problem is solved by Particle Swarm Optimization [12], which is a genetic optimization algorithm inspired by the group dynamics of bird swarms. For each identified subset, we discover a C-Net using the Data-Aware Heuristics Miner [15] and compute the replay fitness.

Table 1. Case- and subset-based process performance indicators (PPIs).

PPI		Description
Case	Control-flow	Directly/eventually followed by Loops
	Resource	Number/distribution of resources
	Data	Case/event attribute value
	Time	Duration of events/trace
	Function	Number/(Co-) Occurrence of events
Subset	Control-flow	Start/End event distribution
	Resource	Attribute resources
	Data	Distribution of case/event attribute values

4.2 Discovery of Insights Recommendations

In this step, our approach automatically obtains relevant process performance indicators (PPIs) yielding only the most interesting insights for each of the subset recommendations. PPIs, which are visually prepared for analysis, measure potential process bottlenecks, inefficiencies, and compliance violations. Our approach helps alleviating the abundant manual and repetitive work caused by the wide range of different PPIs that need to be evaluated.

We adapt the idea introduced in SEEDB [24] where the interestingness of visualizations is judged by how large the deviation of the visualized data is from a reference (e.g., a different event log, historical data, or other subsets). Similarly, we calculate PPIs from the selected subset and compare them to a reference to assess interestingness. Although there are other criteria that may make PPIs interesting, process analysts are particularly interested in deviations as these are indicators for process inefficiencies or anomalies. We propose a set of basic PPIs which distinguishes between case- and subset-based PPIs (see Table 1). Case-based PPIs describe a case-specific characteristic, and subset-based PPIs are defined as aggregations over all cases within a subset. The set of PPIs does not necessarily consider all existing aspects of a process, but our approach is largely agnostic to the particular definition of the PPI.

We gather interesting insights from the event log by performing statistical significance testing on the selected PPIs. These tests determine whether the PPI values of the subset and the reference set are drawn from different distributions. Specifically, the null hypothesis of the statistical significance test is that the two PPI value sets are drawn from the same distribution. For each test, a significance level is set beforehand which is the accepted probability that the null hypothesis is true, but wrongly rejected. Based on the p-value and the significance level, we decide to reject the null hypothesis or not. If the null hypothesis is rejected, the corresponding PPI is considered as an insight recommendation. We apply two different statistical significance test based on the type of the PPI: For case-based PPIs, we use the Kolmogorov-Smirnov test, and for subset-based PPIs, we use the Jensen-Shannon divergence.

We additionally calculate Cohen’s d effect size because the significance value does not necessarily indicate how large or small such a deviation is. For two measurement series x_1, x_2 , it is defined as follows:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{(s_1^2 + s_2^2)/2}} \quad \text{with} \quad s_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (x_{j,i} - \bar{x}_i)^2$$

The effect size illustrates the amount of deviation in a comprehensive and understandable scale. Cohen introduces ranges that determine how large or small the deviation is: $0.2 < d \leq 0.5$ indicates a small effect, $0.5 < d \leq 0.8$ a medium effect, and $d > 0.8$ a large effect.

In our experiments, we found that some PPIs have a strong correlation between each other which leads to additional redundant insights. We summarize correlating insights by grouping them together using clustering. The pairwise Spearman correlation matrix is computed over all relevant PPIs, and it is used as the input for the clustering. The optimal number of clusters is determined by the elbow method.

4.3 Ranking of Recommendations

In this step, our approach ranks subset recommendations based on their relevance, yielding a ranked top- k recommendation list. The score given to each subset is the average score of the identified insights within, as defined by the product of effect size and coverage. The coverage is the proportion of cases in a subset that fulfill the specific insight. This criterion ensures that subsets with high representative insights are ranked higher.

In addition, we apply diversifying top- k ranking [20] over the list of relevant subsets, which returns a list of k items based on the score and diversity. We do this to avoid the typical issue of top- k lists, which tend to have very similar items on top of the list. Users tend to pay more attention to items on top of the list which can cause low selection diversity and the filter bubble effect [19]. We define diversity as the combination of trace and case attribute similarity of the cases within each subset. We use the same similarity function that was used for generating the subset recommendations to identify most diversifying subsets. Based on the similarities of the subsets and the calculated relevance score, we return a ranked top- K list of subset recommendations.

5 ProcessExplorer Implementation System

We implemented our approach into an interactive exploration system to illustrate and evaluate its usefulness. Our system allows analysts to interactively explore event logs with the automatic guidance provided by the ranked subset and insights recommendations. We describe our system using the publicly available event log of the BPI Challenge 2019¹ as a use case example. This event

¹ van Dongen, B.F., Dataset BPI Challenge 2019. 4TU.Centre for Research Data.

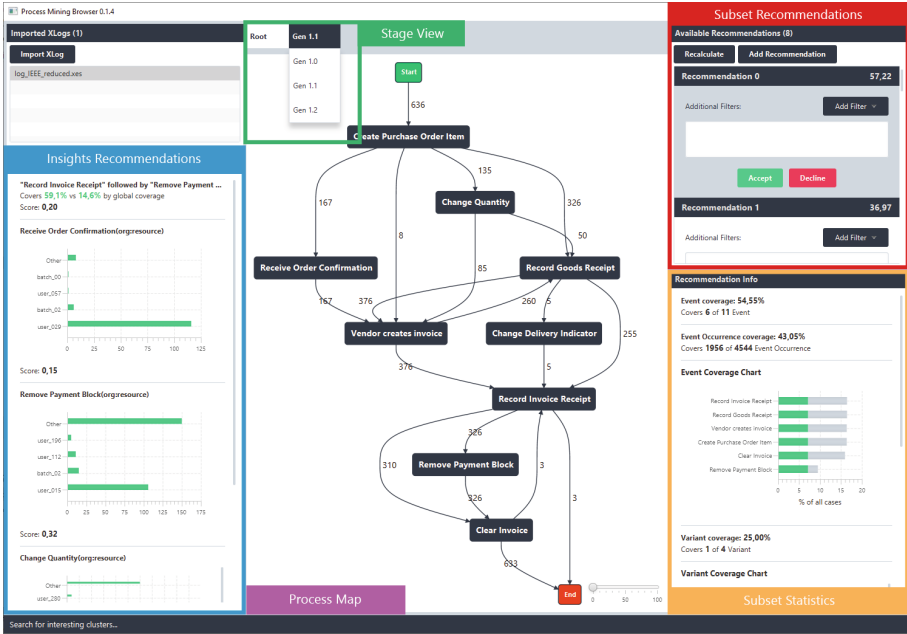


Fig. 2. User interface of ProcessExplorer showing the subset and insights recommendations, the process map of the selected subset, the stage view, and the subset statistics.

log contains data of the purchase order handling process collected from a multinational company in the Netherlands that deals with coatings and paints. The event log consists of 1,595,923 events relating to 42 activities in 251,734 cases. The events are executed by 627 different users (607 human users and 20 batch users). The overall user interface of our system is depicted in Fig. 2 and consists of the subset recommendations, the insights recommendations, the process map, the stage view, and the subset statistics. A detailed view of the components is presented in Fig. 3.

Process Map. Similar to other process mining tools, the process map visually shows the underlying process with navigation and visualization features found in existing process mining tools. This is depicted in Fig. 2. The user can hide activities and transitions using the relative occurrence slider at the bottom right of the process map. The figure presents the process map of an applied subset recommendation.

Subset Recommendations. The list of subset recommendations shown in Fig. 3 top-right are obtained from the event log. Subset recommendations are presented to the user as a list sorted by their assigned score. The user can modify subset recommendations by adding additional filters, such as the variant filter, the start and end activity filter, or happy path filter. Subset recommendations

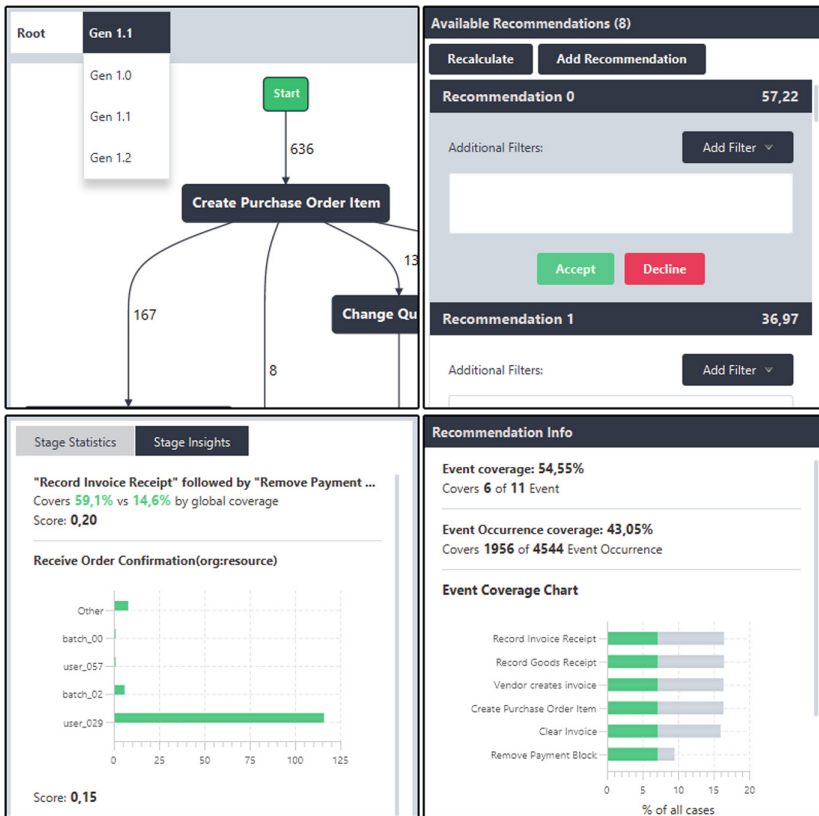


Fig. 3. Detailed overview of the stage view, the subset recommendations, the insights recommendations, and the subset statistics (from top-left to bottom-right).

can be applied or deleted by the user. For the use case event log, 10 subset recommendations are returned.

Subset Statistics. The subset statistics depicted in Fig. 3 bottom-right give an overview of the activity distribution, variants, traces, and transitions of the containing cases. The subset statistic of our example use case shows that the selected subset only contains 6 out of 11 events, covers 1956 event occurrences, and shows the distribution of events compared to the previous stage.

Insights Recommendations. The insights recommendations depicted in Fig. 2 are the insights of the subset that the user decided to apply. Trace-based PPIs are rendered as text, describing the identified deviations. Cluster-based PPIs are rendered as bar charts, showing the distribution of values. Specifically for our running example, Fig. 3 bottom-left shows 2 out of 6 insights recommendations. In the example, subset PPIs were compared to all cases in the event log. The first indicates that the “Record Invoice Receipt” activity is directly followed by

“Remove Payment Block” activity in 59.1% of the cases, compared to 14.6% in the overall event log. The second insight shows the distribution of resources for the activity “Receive Order Confirmation” which highlights the “user.029”.

Stage View. The stage view shown in Fig. 3 top-left allows simplified navigation between subset recommendations. A stage view records the applied subset recommendations in a hierarchical structure. Applying a subset creates a new stage for which new recommendations are computed based on the containing cases. Three different stages are depicted in our example use case. The stages were generated by accepting the first three subset recommendations.

6 Evaluation

In this section, we evaluate our proposed system in two different studies. First, we present a preliminary study that helped us defining the requirements of our ProcessExplorer implementation system. Then, we present a user study aimed at evaluating the usefulness of our approach through the application of our implementation system to a particular case.

6.1 Preliminary Study: Identify Key Requirements

A preliminary study based on an early prototype was conducted to define, based on expert analysts opinions, the key requirements to build a fully-fledged version of the ProcessExplorer system. In particular, we are interested on which kind of guidance may be more helpful to analysts for obtaining valuable knowledge from the event log more efficiently.

Setup. The prototype was shown to five process mining consultants (1 female) which are all familiar with state-of-the-art tools, such as Fluxicon Disco, PAFnow Process Mining (all participants), QPR Process Analyzer (4), Celonis and ProM (2). For the user study, we used the publicly available real-life event log from the BPI Challenge 2017. The interface was compared in three different modes. In the first mode, we consider only manual filtering with no guidance being provided. In the second mode, we introduce the stage view functionality allowing the analyst to focus on specific parts. In the third mode, we further introduce the subset recommendation feature which enables to analyze certain parts of the process in a semi-automatically fashion. All participants were asked to get familiar with the provided guidance and explore the given event logs. After exploring the event log with a specific mode, the participants were asked to fill out a post-stage questionnaire to rate their explicit experience and preference with the interfaces. Finally, we asked the participants to fill out the *User Experience Questionnaire* (UEQ) [13]. The UEQ consists of 5 scales: *Attractiveness* which reflects the overall impression, *perspicuity*, *efficiency*, *dependability* (pragmatic quality), *stimulation* and *novelty* (hedonic quality).

Results. According to the responses we obtained from the post-stage questionnaire, participants rated the stage view with the ability to navigate between the

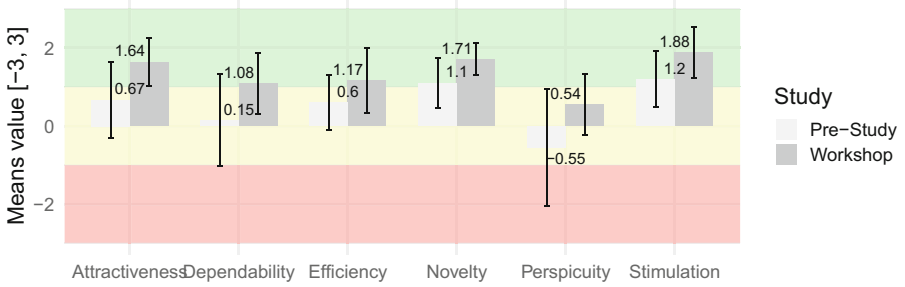


Fig. 4. Results of the UEQ for the preliminary study and for the final prototype. Range from -3 (horribly bad) to 3 (extremely good).

different subsets as beneficial and useful for the systematic exploration of large event logs. However, two participants took longer to analyze the event log due to the increased navigation capabilities. The participants rated negatively the need to navigate between the different views to change the selection. The subset recommendation feature received positive comments, but the information presented was not sufficient to let users directly see what the subset is all about. Furthermore, one participant criticized the top-down approach which makes it difficult to find the really important parts of the process. He suggested recommending potential interesting subsets of groups which can be inspected one-by-one.

In Fig. 4 we present the results of the UEQ. We observed that stimulation and novelty got the highest score, attractiveness, efficiency, and dependability only an average score and perspicuity a negative score. Furthermore, our study shows that participants consider the idea of a systematic analysis innovative and promising. However, our results do not offer conclusive evidence that the analysis experience is improved with our preliminary prototype’s user interface. The detailed comments from the participants show a steep learning curve for the preliminary prototype implementation, indicating that more guidance support was required.

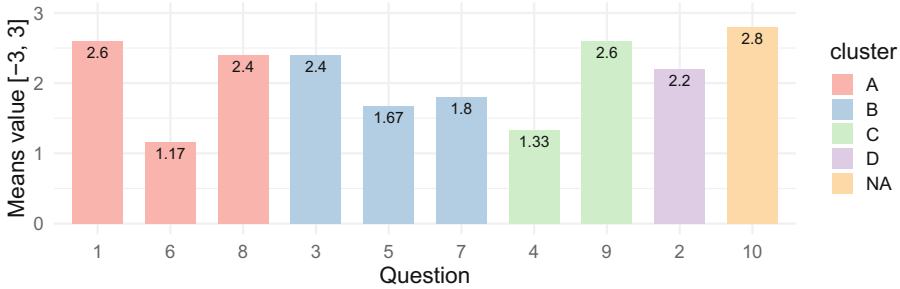
6.2 User Study: Evaluation of Usefulness

A user study with the ProcessExplorer system was conducted to evaluate the usefulness of the system and the underlying concepts. Particularly, we are interested in how useful the subset and insights recommendations are for exploring large event logs.

Setup. The system was shown to six, different from the pre-study, process mining experts in a user study workshop. The workshop took 60 min and was divided into two parts. In the first part, ProcessExplorer was presented to the participants, and we introduced the implemented guidance features. We used BPI Challenge 2019 event log which was known to all study participants, so that little or no explanation was required about the inspected process. Participants

Table 2. The questions and results of the TAM usefulness estimation.

Question	Cluster	Mean	SD
1. Using ProcessExplorer improves the quality of the work I do	A	2.60	0.49
6. Using ProcessExplorer improves my job performance	A	1.17	2.11
8. Using ProcessExplorer enhances my effectiveness on the job	A	2.40	0.80
3. ProcessExplorer enables me to accomplish tasks more quickly	B	2.40	0.80
5. Using ProcessExplorer increases my productivity	B	1.67	2.21
7. Using ProcessExplorer allows me to accomplish more work than would otherwise be possible	B	1.80	1.94
4. ProcessExplorer supports critical aspects of my job	C	1.33	2.21
9. Using ProcessExplorer makes it easier to do my job	C	2.60	0.49
2. Using ProcessExplorer gives me greater control over my work	D	2.20	0.75
10. Overall. I find the ProcessExplorer system useful in my job	NA	2.80	0.4

**Fig. 5.** Overview of the TAM usefulness estimation for ProcessExplorer according to the question clusters which scales from -3 (strong disagree) to 3 (strong agree).

were asked to express their explicit opinion about the system. In the second part of the workshop, the participants were able to explore the implemented guidance features to see how ProcessExplorer guides their analysis work.

For the evaluation of ProcessExplorer, we applied the *Technology Acceptance Model* (TAM) [6], which is a quantitative method to evaluate the potential acceptance of a given technology by end users. The standard evaluation form of the TAM consists of 10 statements (see Table 2) each assigned to a specific cluster related to the usefulness aspects (A = job effectiveness, B = productivity and time savings, C = importance of the system to the users' job, D = control over the job). The statements were rated by the study participants to what extent they agree on a scale of -3 “*Strongly disagree*” to 3 “*Strongly agree*”.

Results. Our results in Fig. 5 show that the ProcessExplorer system received a positive overall usefulness mean score of 2.8 ($SD = 0.4$). For each of the specific cluster of the TAM, our system obtained a positive mean score. Statement 1 in cluster A received a high positive rate with a mean of 2.6 ($SD = 0.49$).

This can be explained by the fact that the ProcessExplorer system provides analysts a more in-depth view into the event log. Analysts can quickly explore the different process behaviors. In spite of this, we observed that the job performance (statement 6) did not improve, showing a lower score of 1.17 ($SD = 2.11$), suggesting that not all participants share the same positive opinion about the system. This might be caused by the short period of exploration time. Most study participants agreed that productivity and time savings can be achieved through the system (cluster B). The participants agreed on the importance of the system to their job (cluster C) because our system improves the exploration of the different process behaviors.

We also asked the participants to fill out the UEQ (see Fig. 4). In comparison to the preliminary study, all values improved. *Attractiveness*, *novelty*, and *stimulation* are now in the above average range.

Lastly, we report on some individual feedback we received from the workshop participants. Most of the participants liked the idea of getting additional guidance during the analysis, instead of beginning without any hints. The idea of generating sub logs and presenting them as subset recommendations was described as “*very innovative*” (P1) and “*super interesting*” (P4). One participant wrote, that “*It’s a very useful tool to gain quick control over unknown data*” (P5). However, one participant (P6) does not think that the insights recommendations are useful. One could argue that the visual representation of these insights need to be clarified. Still, the subset recommendation was seen “*as the real added value to process mining.*” (P6). Two participants found the user interface (P2, P4) too overloaded with all the information shown at the same time.

7 Discussion

Validity. While our evaluation on the usefulness of ProcessExplorer provided relevant observations, other aspects specific to our approach deserve further analysis. For instance, the effectiveness of the subset recommendation approach was discussed in the prior paper [21] which introduced the multi-perspective trace clustering. However, we did not conduct an effectiveness analysis of the insights proposed by ProcessExplorer for practical use. In terms of our evaluation methodology, we consider that a larger number of study participants could help finding more insights on the practicality of our approach.

Limitations. We consider that our approach exhibits mainly two limitations. One is related to the user interface, which is based on other process discovery tools and extended the process model view with recommendations. A non-static dashboard-like interface with customization capabilities could further improve our system. The second limitation we consider is that we score the interestingness of insights based on deviations. This may limit our approach to only work with PPIs that follow this behavior pattern. However, related work [24] has shown the efficacy of this deviation-based metric.

8 Conclusion and Future Work

In this paper, we presented ProcessExplorer, a novel interactive process mining guidance approach. It automatically generates ranked subsets of interesting cases based on control flow and data attributes, similar to the workflow of analysts who manually select certain cases. ProcessExplorer also evaluates a range of relevant PPIs and suggests those with the most significant deviation. We implemented our approach into an interactive exploration system, which we built based on requirements of expert analysts gathered during a preliminary study. To evaluate the usefulness of our approach we conducted a user study with business process analysts. Our results show that our approach can be successfully applied to analyze and explore real-life data sets efficiently.

As future work, we plan to extend our subset recommendation mechanism by applying activity clustering, which will allow to further narrow the analysis to specific parts of the process. It may also be of high interest to investigate different interestingness measures. We also plan to extend the user study to consider a longer period of time and larger number of participants. It may also be of interest to investigate in the ranking as well as the negative implications of automatic guidance to experts' analysis performance.

Acknowledgements. This work is funded by the German Federal Ministry of Education and Research (BMBF) Software Campus project “AI-PM” [01IS17050] and the research project “KI.RPA” [01IS18022D].

References

1. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-3-642-19345-3>
2. Ballambettu, N.P., Suresh, M.A., Bose, R.P.J.C.: Analyzing process variants to understand differences in key performance indices. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 298–313. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_19
3. Beheshti, S.M.R., Benatallah, B., Motahari-Nezhad, H.R.: Scalable graph-based OLAP analytics over process execution data. *Distrib. Parallel Databases* **34**(3), 379–423 (2015)
4. Bolt, A., de Leoni, M., van der Aalst, W.M.P.: Process variant comparison: using event logs to detect differences in behavior and business rules. *Inf. Syst.* **74**, 53–66 (2018)
5. Chatain, T., Carmona, J., van Dongen, B.: Alignment-based trace clustering. In: Mayr, H.C., Guizzardi, G., Ma, H., Pastor, O. (eds.) ER 2017. LNCS, vol. 10650, pp. 295–308. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69904-2_24
6. Davis, F.D.: A technology acceptance model for empirically testing new end-user information systems: theory and results. Ph.D. thesis, MIT (1985)
7. Demiralp, Ç., Haas, P.J., Parthasarathy, S., Pedapati, T.: Foresight: recommending visual insights. *Proc. VLDB* **10**, 1937–1940 (2017)

8. Dijkman, R., Wilbik, A.: Linguistic summarization of event logs – a practical approach. *Inf. Syst.* **67**, 114–125 (2017)
9. van Eck, M.L., Sidorova, N., van der Aalst, W.M.P.: Guided interaction exploration and performance analysis in artifact-centric process models. *Bus. Inf. Syst. Eng.* 1–15 (2018)
10. Grahne, G., Zhu, J.: Fast algorithms for frequent itemset mining using FP-trees. *IEEE Trans. Knowl. Data Eng.* **17**(10), 1347–1362 (2005)
11. Joglekar, M., Garcia-Molina, H., Parameswaran, A.: Interactive data exploration with smart drill-down. In: *Proceedings of the 32nd ICDE*. IEEE (2016)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of ICNN 1995 - International Conference on Neural Networks*. IEEE (1995)
13. Laugwitz, B., Held, T., Schrepp, M.: Construction and evaluation of a user experience questionnaire. In: Holzinger, A. (ed.) *USAB 2008*. LNCS, vol. 5298, pp. 63–76. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89350-9_6
14. Luo, Y., Qin, X., Tang, N., Li, G.: DeepEye: towards automatic data visualization. In: *Proceedings of the 34th ICDE* (2018)
15. Mannhardt, F., De Leoni, M., Reijers, H.A.: Heuristic mining revamped: an interactive, data-aware, and conformance-aware miner. In: *BPM Demos*, vol. 1920 (2017)
16. Mendling, J., Reijers, H.A., Cardoso, J.: What makes process models understandable? In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 48–63. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_4
17. Milo, T., Somech, A.: Next-step suggestions for modern interactive data analysis platforms. In: *Proceedings of the 24th SIGKDD*. ACM Press (2018)
18. Mutlu, B., Veas, E., Trattner, C.: VizRec: recommending personalized visualizations. *ACM Trans. Interact. Intell. Syst.* **6**, 1–39 (2016)
19. Nguyen, T.T., Hui, P.M., Harper, F.M., Terveen, L., Konstan, J.A.: Exploring the filter bubble: the effect of user recommender systems on content diversity. In: *Proceedings of the 23rd WWW*. ACM Press (2014)
20. Qin, L., Yu, J.X., Chang, L.: Diversifying top-k results. *Proc. VLDB* **5**, 1124–1135 (2012)
21. Seeliger, A., Nolle, T., Mühlhäuser, M.: Finding structure in the unstructured: hybrid feature set clustering for process discovery. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *BPM 2018*. LNCS, vol. 11080, pp. 288–304. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_17
22. Singh, M., Cafarella, M.J., Jagadish, H.V.: DBExplorer: exploratory search in databases. In: *EDBT*, pp. 89–100 (2016)
23. Sun, Y., Bauer, B., Weidlich, M.: Compound trace clustering to generate accurate and simple sub-process models. In: Maximilien, M., Vallecillo, A., Wang, J., Oriol, M. (eds.) *ICSOC 2017*. LNCS, vol. 10601, pp. 175–190. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69035-3_12
24. Vartak, M., Madden, S., Parameswaran, A., Polyzotis, N.: SeeDB. *Proc. VLDB* **7**, 1581–1584 (2014)
25. Wang, P., Tan, W., Tang, A., Hu, K.: A Novel trace clustering technique based on constrained trace alignment. In: Zu, Q., Hu, B. (eds.) *HCC 2017*. LNCS, vol. 10745, pp. 53–63. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74521-3_7
26. Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J.: Voyager: exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. Vis. Comput. Graph.* **22**(1), 649–658 (2016)
27. Yang, S., et al.: VIT-PLA: visual interactive tool for process log analysis. In: *KDD IDEA Workshop*, vol. 5, pp. 130–137 (2016)



Machine Learning-Based Framework for Log-Lifting in Business Process Mining Applications

Ghalia Tello¹(✉), Gabriele Gianini^{2,3}(✉), Rabeab Mizouni¹(✉),
and Ernesto Damiani^{1,2}(✉)

¹ Khalifa University of Science and Technology, Abu Dhabi, UAE
{ghalia.tello,rabeab.mizouni,ernesto.damiani}@ku.ac.ae

² Etisalat British Telecom Innovation Center (EBTIC), Abu Dhabi, UAE
gabriele.gianini@ku.ac.ae

³ Computer Science Department, Università degli Studi di Milano, Milan, Italy

Abstract. Real-life event logs are typically much less structured and more complex than the predefined business activities they refer to. Most of the existing process mining techniques assume that there is a one-to-one mapping between process model activities and events recorded during process execution. Unfortunately, event logs and process model activities are defined at different levels of granularity. The challenges posed by this discrepancy can be addressed by means of log-lifting. In this work we develop a machine-learning-based framework aimed at bridging the abstraction level gap between logs and process models. The proposed framework operates of two main phases: *log segmentation* and *machine-learning-based classification*. The purpose of the segmentation phase is to identify the potential segment separators in a flow of low-level events, in which each segment corresponds to an unknown high-level activity. For this, we propose a segmentation algorithm based on maximum likelihood with n-gram analysis. In the second phase, event segments are mapped into their corresponding high-level activities using a supervised machine learning technique. Several machine learning classification methods are explored including ANNs, SVMs, and random forest. We demonstrate the applicability of our framework using a real-life event log provided by the SAP company. The results obtained show that a machine learning approach based on the random forest algorithm outperforms the other methods with an accuracy of 96.4%. The testing time was found to be around 0.01s, which makes the algorithm a good candidate for real-time deployment scenarios.

Keywords: Process mining · Segmentation · Log lifting · Machine learning

1 Introduction

With the evolution of Information Technology (IT), companies and organizations increasingly rely on IT services to support their business processes in different

forms. Managing business processes in an accurate, efficient, and well-organized way has an important influence on the organization output and on its success [12]. Today, a number of existing process mining methods can automatically discover process models, check the conformance of process execution to model specification, and even propose enhancement to the currently used model itself [2]. Most of those process mining methods, however, assume that there is a one-to-one mapping between events in the log and the process model activities. Whereas, in real-life scenarios, event logs and process model activities are at different level of granularity [13].

The common praxis of obtaining an activity sequence simply by means of a one-to-one mapping from one events to an activity (e.g. by simple string substitution of the event names with activities names) is ineffective and produces models, less meaningful, difficult to mine and to interpret [24]. Indeed, events in the log are finer grained than activities in process models with $n : m$ relationship [22]. In other words, a high-level activity typically consists of multiple low-level events, also, one type of low-level event might be related to multiple high-level activities. For example, activity *notify the request's outcome* might contain multiple low-level events such as *make a phone call* and *send an email*. On the other hand, *send an email* event might be as well part of *notify the request's outcome* and of *file a complaint*. Therefore, in order to cover all the possible cases, a mapping approach that can support $n : m$ relationship between events and activities is required. One way around this is to map a “sequence” of low-level events to the corresponding activity. However, generating such sequences is also a challenge.

Existing techniques focus on grouping similar events to form higher level activities, which are then used to discover simpler process models of the underlying behavior. A critical assumption in such techniques is that the discovered event clusters correspond to occurrences of meaningful business activities. In such case, the abstracted log might support model discovery, however, it is not very useful for detecting violations and deviations from the expected business process model. Accordingly, a mapping approach that can map flow of low-level events to predefined business model activities is needed. Besides, it is important that the manual work required by such approach should be minimal, in order to provide fast and feasible performance.

In line with these challenges, in this paper, we design and implement a Machine Learning (ML) framework that is able to learn the mapping between low-level event logs and process model activities. Our framework operates in two main phases: *log segmentation* and ML-based classification. The purpose of the former phase is to identify the potential segment separators in a flow of low-level events: each segment corresponds to an unknown high-level activity. The purpose of the second phase is to map each segment to a corresponding high-level activity, using a supervised machine learning algorithm. We show that this approach can automatically and accurately bridge the abstraction level gap.

The paper is structured as follows. We start summarizing the related work in Sect. 2. Section 3 provides preliminary definitions and presents the proposed

framework. Section 4 describes the implementation of this framework and demonstrates its applicability using the case study of a real-life event log (from the software company SAP). Section 5 concludes the paper.

2 Related Work

Most of the existing approaches focus on clustering techniques to group coherent low-level events and then map them to high-level activities. For example, Günther and van der Aalst [16] developed a clustering approach to group event instances to high-level activities according to their closeness to other events in their trace. In this approach an event class can exist in multiple high-level activities. However, this approach has serious performance issues such as high memory consumption and a run-time complexity exponential with the log size [17]. In line with this, Günther et al. [17] proposed a new approach based on global trace segmentation that can cluster the events based on the class level. It is a bottom up technique that starts with low-level events in a trace. Then, it tries to find coherent event sub-sequences and successively combine them into clusters. The correlation among events is found from the co-occurrence of events in the log. From correlation, a hierarchy of event classes is inferred. Although their approach provides better performance with linear complexity, it doesn't support $n : m$ relations between events and activities, since event classes can only belong to one activity. Moreover, they provide no clear guidance regarding when to stop clustering low-level events into higher level of abstraction.

Pérez-Castillo et al. [23] proposed an approach to study the correlation between events using event attributes. Again, this approach only considers 1 : 1 mapping between events and activities. Li et al. [20] proposed an approach to identify semantically related patterns of events. This approach can handle $n : m$ relations. However, it doesn't allow more than two events to occur between the events of the same activity instance.

Bose et al. [5] introduced an approach to discover abstractions (high-level activities). It consisted in recognizing frequent event patterns in the event log. After that, each frequent pattern is considered as one high-level activity. Then, the abstracted log is created replacing the frequent patterns. Finally, the abstracted log can be used to discover process models using discovery techniques such as the α -algorithm [1], the genetic miner [10], or the heuristics miner [25]. The drawback of this approach is that there is no guarantee that the discovered frequent event patterns actually correspond to meaningful activities in the business process.

Mannhardt et al. [22] proposed an event abstraction method based on behavioral activity patterns. It basically aims at developing an abstracted log by aligning low-level event logs with activity patterns. An activity pattern is a low-level process model of a certain activity. An integrated model is built by combining the activity patterns in one model. After that, the integrated model is aligned with the event log using existing alignment techniques in order to create the aligned log. Although this approach seems easy to follow, however it has some

disadvantages. First, it seems unreasonable to expect the user to create a low-level model (e.g. petri nets) for each high-level activity. Moreover, the formalized pattern of the user might miss important patterns that occur in reality.

Several works have addressed the use of taxonomies and ontologies to operate at different levels of abstractions [6, 15, 19]. They try to relate elements in the logs to concepts they represent in ontologies [8, 9]. For example, in [18] the authors use company-specific ontologies and databases to get multiple abstraction levels. However, not all organizations have taxonomies and ontologies. It is important to mention that our work target the case when taxonomies and ontologies are not available. Therefore, these works are only loosely related to our contribution.

To the best of our knowledge, in the field of process mining, our approach is the first to utilize and compare different machine learning techniques to fill the gap between low-level event logs and high-level activities defined in the process model. Moreover, our approach overcomes the limitations of the existing works as it can map the event logs into meaningful and predefined activities, does not require ontologies, supports $n : m$ relationships, and operates with minimal human intervention.

3 Overview of the Proposed Framework

Figure 1 shows an overview of the overall log-lifting framework. First, a segmentation algorithm is needed to locate the segment separators in a flow of low-level events where each segment corresponds to an unknown high-level activity that is yet to be discovered. For this, a supervised machine learning-based algorithm is used in order to classify a real-time flow of event segments (testing-set). However, in order to apply a supervised algorithm, a number of labeled examples are needed to train the classifier. In fact, labeled logs are usually unavailable, and manual labeling might be infeasible, time consuming or too expensive. To cope up with this challenge, a clustering-based labelling approach is introduced to label a set of event sequences which will then be used to train the classifier. After training, only the decision boundaries are kept as mathematical formulas in order to classify future real-time flow of low-level event sequences.

The proposed framework can be seen as a two-phase approach. The first phase is log-segmentation, followed by a machine learning-based classification phase. In the following sections, we will discuss each phase separately.

3.1 Log Segmentation

The flow of low-level events has to undergo a segmentation process: the segments yielded by such a process would represent estimated high level activities. Conceptually it is analogous to taking a long text provided without separating spaces and breaking down the text into words (adding spaces). It is analogous to a decoding problem, where one has to guess which pairs of low level symbols/letters need to be decoded by adding a space symbol. At the end of this

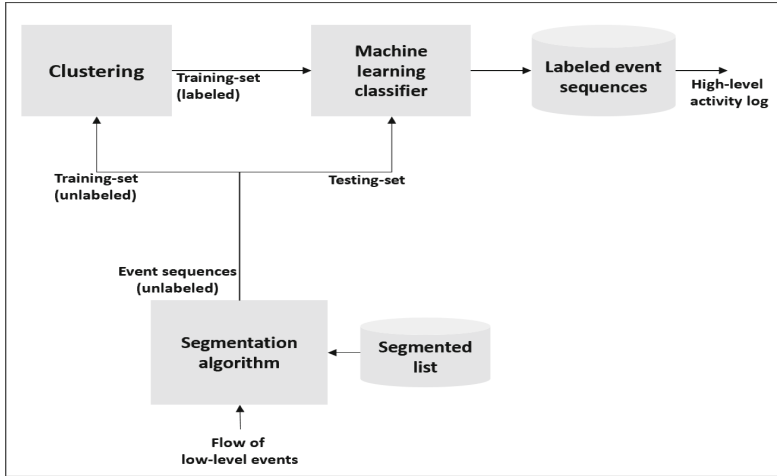


Fig. 1. Schematic description of the log-lifting framework

step, each segment contains a number of low-level events and corresponds to an activity; we call it low-level sequence, or event sequence.

This kind of segmentation is very sensitive to the assumptions about the potential segment separators. The presence of common prefixes is a critical challenge that needs to be addressed. In other words, one activity could be the prefix of another. For example, given four activities A , B , C and D with the corresponding low-level event sequences: $\{\alpha, \beta, \textit{gamma}\}$, $\{\alpha, \beta\}$, $\{\delta, \psi\}$ and $\{\rho, \delta, \psi\}$, respectively. In such case, there is an ambiguity in selecting the potential segment separators for the flow: $\{\alpha, \beta, \gamma, \delta, \psi\}$, as two possible mappings might be generated: $A + C$ or $B + D$. Starting the mapping by both activities A or B is a good start. However, in the long term perspective, one mapping (i. e. $A + C$) might be better than the other with respect to the subsequent flow of events. Because segmenting the flow as $A + C$ has a perfect match between the low-level events and high-level activities. However, the second mapping option, i.e. $B + D$, has less similarity as D in fact map to $\{\rho, \delta, \psi\}$ rather than $\{\gamma, \delta, \psi\}$.

In line with this, we propose an optimization approach based on maximum likelihood technique to address the segmentation problem. The segmentation requires a relatively small set of examples, provided by the data owner (e.g. SAP) annotated with the correct separation between one activity and the next, we call it “segmented list”. In fact, the same agent naturally sets a break between sequences of high level activities. The break can correspond to the end of a working session or to a pause within a working session. Alternatively, the separation between two sentences can be realized using time gaps. Since activities in organizations are usually repetitive, the segmented list is relatively small and concise.

3.1.1 Walkthrough

For the sake of clarity hereafter we will use mostly the English text analogy terms as follows:

- Low level events correspond to letters/symbols: e.g. “e”, “d”, “i”, “t”.
- Low level event sequences correspond to words: e.g. “edit”.
- In the decoded text one word is separated from the next by a space: e.g. “open edit save close”.
- The non-decoded text is analogous to a sequence without spaces where all the words are attached one after the other: e.g. “openeditsaveclose”.
- The words form sentences (i.e. high level sequences of activities), that we distinguish from the time gap between one and the next, analogous to the period: e.g. “open edit save edit save edit save close”.
- Some sentences are normal and highly probable in a log/text, some others are anomalous: e.g. “open save close open save close open save close”.

The segmented list allowed to run a standard n -gram analysis. For example, counting digrams allowed creating a prefix-suffix table with one letter as a prefix and one as suffix. The frequencies of each row would provide an estimate of the likelihood of a given digram actually occurring. The choice of n for n -gram matrix depends on the dataset. For this, we plot a histogram with a bar for each segment length in the segmented list. The length corresponds to the highest bar is chosen as n .

Given an un-decoded sequence and given the n -gram matrix one can compute the likelihood of any candidate decoding. By standard optimization techniques one could find the optimal decoding, but a satisfactory decoding can also be accepted to limit the run time of the procedure. The output of this phase is sequence of words representing high-level activities.

3.1.2 Running Example

We demonstrate steps taken by the algorithm using a simple example. Assume we have the following segmented list: $\{\{A\}, \{AB\}, \{BC\}, \{CG\}, \{DE\}, \{F\}, \{DBCA\}\}$, and we want to segment the following flow of events: $\{D E F D B C A C G\}$:

1. Build a digram matrix. For this, we have to create a unique events vector in order to construct a dictionary with event-index values. Then, we construct the digram matrix by looping over the segmented list and incrementing the corresponding digrams in the matrix. Accordingly, each cell in the matrix table will be representing the frequency of occurrence for the corresponding event digram. Initially, the digram matrix looks as follows:

	-	A	B	C	G	D	E	F
-	0	2	1	1	0	2	0	1
A	2	0	1	0	0	0	0	0
B	1	0	0	2	0	0	0	0
C	1	1	0	0	1	0	0	0
G	1	0	0	0	0	0	0	0
D	0	0	1	0	0	0	1	0
E	1	0	0	0	0	0	0	0
F	1	0	0	0	0	0	0	0

2. The likelihood of each digram is calculated by dividing its frequency by the total sum of its row. In order to allow for unobserved combinations a small value (e.g. 0.01) can be added to all matrix values. The resulted matrix is as follows:

	-	A	B	C	G	D	E	F
-	0.0014	0.2838	0.1426	0.1426	0.00141	0.2838	0.0014	0.1426
A	0.6525	0.0032	0.3279	0.0032	0.00324	0.0032	0.0032	0.0032
B	0.3279	0.0032	0.0032	0.6525	0.00324	0.0032	0.0032	0.0032
C	0.3279	0.3279	0.0032	0.0032	0.32792	0.0032	0.0032	0.0032
G	0.9351	0.0092	0.0092	0.0092	0.00925	0.0092	0.0092	0.0092
D	0.0048	0.0048	0.4855	0.0048	0.0048	0.0048	0.4855	0.0048
E	0.9351	0.0092	0.0092	0.0092	0.00925	0.0092	0.0092	0.0092
F	0.9351	0.0092	0.0092	0.0092	0.00925	0.0092	0.0092	0.0092

3. Compute the likelihood of the given flow using the following equation, which is inspired by the Markov chain Monte Carlo algorithm [11]. Given a matrix $M(x; y)$, the likelihood of the flow with n events $\{e_1, e_2, \dots, e_n\}$ is:

$$\mathcal{L} = \prod_i M(e_i, e_{i+1}) \tag{1}$$

Alternatively:

$$\mathcal{L} = \sum \log M(e_i, e_{i+1}) \tag{2}$$

4. Scan the flow and insert a separator in a random position within a window. The total likelihood is then calculated again to check whether there is an improvement in the segmentation likelihood. If it is the case, then we insert the separator and update the flow, otherwise, we skip and shift the window and repeat same steps again until we reach the end of the flow.

5. Keep scanning the flow and randomly update the position of the separator until reaching a convergence in segmentation likelihood.

The resulted final positions of the separators: {D E _ F _ D B C A _ C G}.

3.2 Machine Learning-Based Classification

After segmentation, each segment contains a number of low-level events that corresponds to an unknown activity, namely low-level sequence. These sequences have to undergo a mapping approach in order to map them to the corresponding high-level activity. As shown in Fig. 2, a two-stage approach is proposed. The first stage consists of a clustering-based labeling approach. As high-level target labels should be available to learn the mapping between low-level events and high-level activities, manual labeling should be performed, which is time consuming and could be even unfeasible. Therefore, a clustering-based labeling approach is proposed to create sample of labeled examples. For every activity, a cluster of low-level event sequences should be formed. And then, all samples will be labeled with same labels as their corresponding clusters. After that, these labeled examples are refined by taking only subset of examples that contribute to the goodness of the labeling task. In the second phase, the labeled examples are used to train a machine learning classifier. The classifier will learn the mapping between low-level event sequences and high-level activities and map a new set of unlabeled low-level sequences of events to the corresponding high-level activities. Hence, a high-level activity log can be created.

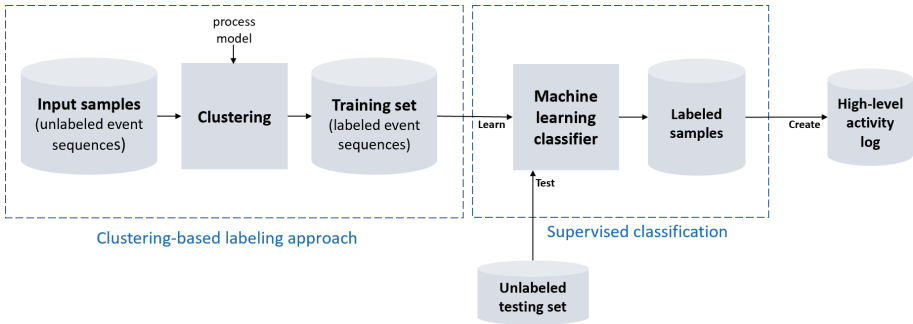


Fig. 2. Overview of the proposed mapping approach

3.2.1 Stage 1: Clustering-Based Labeling

The goal of this phase is to create labeled examples to feed the supervised machine learning classifier of the next phase. The input samples of this phase are unlabeled event sequences. Each sample is a set of low-level events. In order to provide labels for these samples, we will cluster them into different clusters

representing high-level activities. Different clustering algorithms can be used to achieve this task. For example, k -prototypes can be utilized to cluster multivariate time series log sequences with numerical and categorical attributes, while k -medoids is applied to cluster samples with categorical attributes. Furthermore, the number of high-level activities can be known from the high-level process model, which accordingly represents the number of clusters k that should be formed. Each sample within one cluster will be labeled with its cluster label. In fact, in order to label the clusters, the domain expert should label only the cluster centers, and then each cluster label will be the same as the label of its center.

In the process mining field, the execution of processes is reflected by event logs. The eXtensible Event Stream (XES) defines a standard for recording information system's events. Typically, an event in the log is defined as $e = (n, c, r, s, t)$, representing the occurrence of an event n , in a case c , using the resource r , having a status $s \in \{\text{start, complete}\}$, at time-stamp t [7]. Additional attributes for the events can be included such as price, originator, location, etc.

It is worth to mention that clustering is an unsupervised approach with high level of ambiguity that will probably result in a number of mislabeled examples. These mislabeled examples will degrade the clustering accuracy and hence will affect the quality of the training set in the next phase. Therefore, in order to enhance the accuracy of clustering approach, cluster members should be prioritized. Cluster members are refined by taking only the subset that contributes more to the goodness of the clustering process. For this, we adopt a variant of the silhouette cutting method, called hereafter Second Best Based Selection (SBBS), and based on distance comparison. Whereas the standard silhouette method removes the points around the centroids, and risks discarding points which fall at a high distance from a centroid but are unambiguously assigned to it, the SBBS takes into account the ambiguity in the decision on whether discarding a point or not. For example, as seen in Fig. 3, there are some misclustered points in the triangles cluster although they are very close to the centroid of the triangles. While there are some points very far from the green centroid but they are still correctly classified.

This is more appropriate to the final purpose of the operation in the current context, i.e. classification of sequences into activities: in principle, the more unambiguous points are left to the upcoming learner, the more accurate the model will be. The SBBS algorithm is described in the following steps:

1. At the last iteration of the clustering algorithm, for each object store in a table its distance/dissimilarity to all the centers (each object is assigned to the cluster-center with minimal distance, i.e. the first best center)
2. For each object compute the difference between the distance to the first best and the distance to the second best cluster-center. Call this discrepancy D .
3. For each cluster, sort the objects by increasing D .
4. Start deletion, using the predefined distance based criterion: e.g. delete a fixed percentage of the elements in order of increasing D , or delete a fixed percentage of the elements of each cluster in order of increasing D .

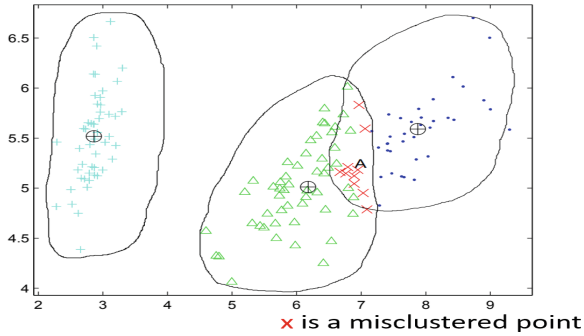


Fig. 3. Ambiguity in overlapped clusters

At this point the dataset L^* is ready to be used as a training dataset for classification algorithms.

3.2.2 Stage 2: Classification

After preparing the training set via the previous clustering-based labeling module, machine learning techniques is used in order to classify event segments to the corresponding high-level activities. To do this, several machine learning classification methods have been explored including Random Forest (RF), Artificial Neural Network (ANN), and Support Vector Machine (SVM). Results of the classification will be detailed in the next section.

4 Case Study

To evaluate the proposed framework, a log provided by SAP company¹ has been used. The event log contains around 12 weeks of activities collected from SAP systems deployed in a test environment. The following features of the dataset samples has been considered: “subobject”, “User”, “Time”, “Date”, “Role”, and “Action”. Below is description of each feature with its possible values:

- *subObject*: which describes the operation being performed on the object. Its possible values are: IOBJ_SAVE, IOBJ_DEL, IOBJ_ACC (related to infocubes, kind of analytic objects meta-data), SDL (for personal/sensitive data operation), and TCD (for maintenance).
- *User*: which represents the name of the user performing the action.
- *Role*: which represents the role of the user. Its possible values are: ADMIN, CREATOR, LAMBDA, and CLEARED.
- *Action*: is the operation name. Its possible values are: CHANGE, DELETE, READ, and ADMIN.
- *Time*: is the start time of the performed action.
- *Date*: is the start date of the performed action.

¹ SAP dataset <https://doi.org/10.5281/zenodo.2566022>.

4.1 Experimental Setup

In order to validate our research hypotheses, we had to label the SAP dataset by direct observation (i.e. empirical evidence). For this, the separation between two sentences/activities was realized using time gaps determined experimentally. By considering a specific range of dates, we identified the activities of a specific user which has a specific role. In other words, we performed two level ordering: the log was first ordered by ‘user’ then within each user we ordered events by ‘time’. Accordingly, the ground truth was generated by labeling each block of low-level events as a separate activity. The low-level event blocks were identified using time gap of 15 min. The size of the dataset used for this experiment is 1,730 low-level event instances with 20 classes of high-level activities.

4.2 Log Segmentation Phase Results

As mentioned before, the first phase of the proposed framework is log segmentation. For this step, we applied the maximum likelihood-based segmentation (introduced in Sect. 3.1). A relatively small set of examples annotated with the correct separation between one activity and the next are taken to form the segmented list. The resulting segmented list used contains 50 sequences, where each sequence contains a series of low-level events.

4.2.1 n-gram Analysis

Following the first step of the segmentation algorithm, a standard n-gram analysis is run on the segmented list. The choice of n in n-gram analysis is specified based on the segment lengths existed in the segmented list. For this, we plot a histogram with a bar for each segment length found in the segmented list. In this example, the segment length correspond to the highest bar is $n = 3$. Hence, we performed trigram analysis.

Accordingly, we constructed a likelihood 3D data structure by counting the number of occurrences for each trigram in the segmented list as follows:

```

for (int i = 0; i < segmentedListSize; ++i)
{
    int segmentedListSequenceSize = (int)segmentedList[i].size();

    for (int j = 1; j < segmentedListSequenceSize-1; ++j)
    {
        x = eventIndecis[segmentedList[i][j - 1]];
        y = eventIndecis[segmentedList[i][j]];
        z = eventIndecis[segmentedList[i][j + 1]];

        ++trigramsMatrix[x][y][z];
    }
}

```

In the above C code excerpt, “*eventIndecis*” is a dictionary constructed using the map standard implementation. It contains key-value pairs, where the keys are unique and used to find the associated values. In our case, the key is the unique event and the values is the associated index in the trigram matrix. Using

this idea of dictionary is more efficient as it reduces the time needed to locate an event in a trigram matrix.

After that, the likelihood of trigrams is calculated by dividing its frequency by the total sum of the row.

4.2.2 Maximum Likelihood Analysis

After that, a flow of 243 low-level events is fed to the segmentation module and the initial likelihood of the flow is calculated. Then, a window of size 3 scans the flow and tries to insert a separator in a random position within the window. The total likelihood is then calculated again to check whether there is an improvement in the likelihood value after inserting the separator. We keep scanning the flow again and update the position of the separator within the window until reaching a convergence in the likelihood value. The output of this phase is segmented low-level sequences that can be fed to the next phase (machine learning phase). We utilized the ground-truth that was constructed using time gap procedure in order to validate our approach, the resulted number of miss-segmented sequences is 11 out of 130 (total number of flow segments). Thus, the percent error of the estimated segments to the actual amount is 0.084, which corresponds to segmentation accuracy of 91.54%.

4.3 Mapping Phase Results

After the segmentation phase, each segment contains a number of low-level events, which corresponds to an unknown activity. These sequences have to undergo a classification approach in order to classify/map each sequence to the corresponding high-level activity. For this, the dataset samples of 1,730 low-level events have been split into training and testing sets. The total number of the sequences/segments is 466. Accordingly, 70% of the data, i.e., 326 sequences have been used for training and the remaining for testing. Before starting with the classification, the training samples are fed into clustering-based labeling module in order to automatically label them and form the training set, which will be then used to learn the classifier. Below we describe the result of each stage.

4.3.1 Clustering-Based Labelling

The clustering analysis is an unsupervised learning, thus, we aim to find similarities between the underlying sequences according to their characteristics and to properly group similar sequences into clusters. We deployed the k-medoids algorithm to group similar low-level events into different 20 clusters ($k = 20$), where each cluster represents a high-level activity.

Table 1 shows an excerpt of the constructed distance matrix containing the distance between each point and another (for all points in dataset). With this matrix, we can directly access the distance between the desired points instead of re-computing it each time we need to update the centroids or members.

The output of k-medoids on a collection of sequences U is a labeling of the sequences by the cluster identifier (i.e. a labeled dataset L). To evaluate the

Table 1. Excerpt of the distance matrix

	p1	p2	p3	p4	p5	p6	p7
p1	0	0.8	0.75	0.2	0.4	1	0.666
p2	0.8	0	0.4	0.6	0.75	0.2	0.4
p3	0.75	0.4	0	0.8	0.8	0.3	0.55
p4	0.2	0.6	0.8	0	0.333	1	0.75
p5	0.4	0.75	0.8	0.333	0	0.75	0.666
p6	1	0.2	0.3	1	0.75	0	0.333
p7	0.666	0.4	0.55	0.75	0.666	0.333	0

accuracy of the k-medoids, we compare the labels obtained by the clustering approach against the ground-truth labels. The resulted clustering accuracy is 80.06%.

After that, we reduced each cluster based on the similarity of objects to the centroids, by a variant of the silhouette method (SBBS), yielding a reduced dataset L^* , fit to be used by a supervised algorithm. Accordingly, the accuracy increased to 91.15%

4.3.2 Machine Learning-Based Classification

In this section, we explore and compare the performance of different machine learning classifiers namely: Support Vector Machines (SVMs), Artificial Neuron Network (ANN), and Random Forest (RF).

A. Support Vector Machines (SVMs):

Since this dataset is non-separable, SVM utilizes a method called soft margin to maximize the margin by allowing some objects to be miss-classified so that a linear separability is still possible. In fact, a slack variable is added to lead to a margin called “soft”. As increasing a penalty variable C , more weights are imposed on the slack variables. Whereas, reducing C towards 0 makes the margin softer. Consequently, this has an affect on overfitting and underfitting the dataset. The penalty variable C is also called “*BoxConstraint*”, as it sets bounds on Lagrange multipliers.

Regarding the type of the separator (kernel), we verified the SVM performance using the most popular kernel types: linear, polynomial and Gaussian. The best testing accuracy is achieved using the Gaussian kernel function. However, the sigma value of the Gaussian kernel, a.k.a, “*kernelScale*” needs to be tuned. In fact, tuning both the penalty variable C and sigma is very important as they collectively manage the tradeoff between underfitting and overfitting. For this, we utilized the “*OptimizeHyperparameters*” option for SVM, where a cross validation approach is followed to find the best values for sigma, C , coding design, etc.

In order to apply a multiclass SVM, the problem should be transformed into a series of binary problems. To achieve this, there are two possible coding designs

one-versus-all and one-versus-one. In one-versus-all, each binary learner has to consider only one class as positive and the rest as negative. Accordingly, for k number of classes, k binary learners have to be trained. Whereas in one-versus-one coding design, one class is considered as a positive class and one as negative while the remaining classes are ignored. This leads to $k(k-1)/2$ binary learners. In this case study, the best coding design turned out to be “one-versus-all”.

After training the model, we tested it on a separate testing set sequences and it achieved an accuracy of 93.57%. It is worth mentioning that SVM might be computationally expensive for large datasets, as it depends on the square root of the number of examples when finding the maximum margin. Moreover, SVM needs to store all support vectors.

B. Artificial Neural Networks

In this section we verify the performance of a feed forward multi-layer perceptrons (MLP). In this case study, the network is trained with training goal as 0.001 and learning rate as 0.01. Moreover, Levenberg-Marquardt optimization algorithm is utilized to update weights and bias values of the MLP network.

The transfer function used in output layer is the sigmoidal function since the activation in output units ranges between 0 and 1. In fact, the output coding in this experiment is based on a binary conversion of the class labels. As mentioned earlier, this dataset contains 20 possible class labels (activities). Therefore, we used a 5-digit binary number to represent each class (e.g. 00001 for activity 1, 00010 for activity 2, 00011 for activity 3, and so on), resulting in 5 output units in the output layer.

Figure 4 shows the accuracy of ANN model using different number of units in the hidden layer. Empirically, ANN achieved an accuracy of 95.71% using 35 hidden units, increasing the number of units further did not add a significant improvement in the accuracy. Thus, we chose 35 units as the optimal number of hidden units for this model.

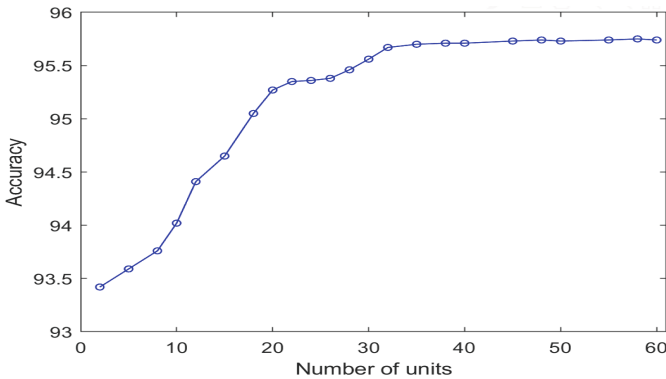


Fig. 4. ANN accuracy vs number of units

C. Random Forest (RF)

Training a single tree is known to be very easy and straightforward. However, decision trees lack stability as a minor change in input might change the tree structure drastically. Accordingly, we employed the rationale of combining learning models (i.e. bagging). With bagging we can reduce the variance of the base learner. Hence, we utilized a large collection of decorrelated decision trees (i.e. random forest).

During the training phase, we constructed the forest which includes n -trees. Each tree contains random event sequences from the training set with replacement. With all of these trees we create different variations of the main activity classification. After building the forest, we tested the random forest model with a separate testing set sequences, where each tree gives an activity class (vote). The final activity decision is the class with the most votes in the forest. Indeed, the more trees in the forest the more robust is the model, and thus the higher the accuracy. As seen in Fig. 5, the classification accuracy of RF is 96.42% with 100 trees, then it stabilizes. Hence, adding more than 100 trees will unnecessarily require additional memory and computation cost for learning these additional trees.

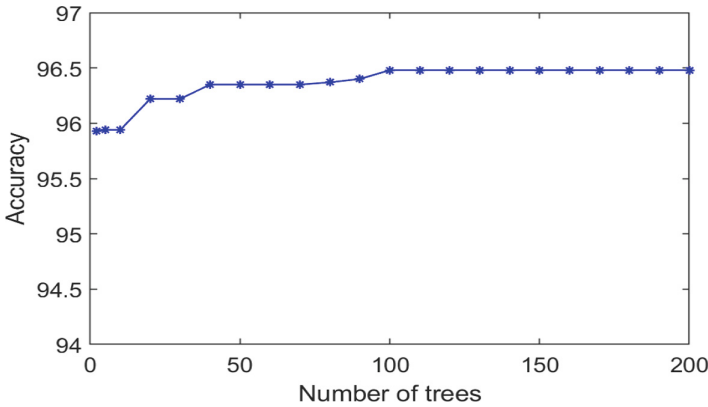


Fig. 5. RF accuracy vs number of trees

Table 2 compares the performance results of the machine learning classifiers in terms of accuracy, ROC, and time to test the model. Random forest outperforms other machine learning classifiers in terms of all performance measures. RF achieved accuracy of 96.48%. It is worth to mention that, as found in the literature, RF is known to be a robust classifier to label-noise in several classification and prediction studies [3, 4, 14, 21, 26]. Accordingly, RF was able to maintain a good classification result even with the presence of some label-noise in the training set, which was generated by the previous phase (clustering-based labeling).

Table 2. Performance comparison of different ML-models

Model	Accuracy (%)	ROC	Time to test model (s)
ANN (MLP)	95.71	0.983	0.02
SVM	93.57	0.978	0.01
Random Forest	96.42	0.984	0.01

5 Conclusions

With our proposed machine learning based log-lifting framework we are able to bridge the gap between low-level event logs and high-level model activities, as commonly found in real business scenarios. The proposed framework overcomes the limitations of the existing work as it can map the event logs into meaningful and predefined activities, does not require ontologies, supports n:m relations, and operates with minimal human intervention.

The proposed log-lifting module consists of two main phases: log segmentation and machine learning-based classification. The purpose of the segmentation phase is to identify the potential segment separators in a flow of low-level events, in which each segment corresponds to an unknown high-level activity. For this, we adopt our proposed segmentation algorithm based on maximum likelihood with n -gram analysis. In the second phase, the event sequences are mapped into their corresponding high-level activities using a supervised machine learning algorithm. Several machine learning classification methods are explored including artificial neural networks, support vector machine, and random forest. We demonstrate the applicability of our approach by conducting a case study using a real-life event log provided by SAP company. Results obtained show that a machine learning approach based on random forest method outperforms the other methods with an accuracy of 96.4%. The testing time was found to be around 0.01 s, which verifies that the adopted algorithm is a successful candidate for real-time deployment scenarios.

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. Van der Aalst, W.M.: *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-3-642-19345-3>
3. Altendorf, J., Brende, P., Lessard, L.: Fraud detection for online retail using random forests. Technical report (2005)
4. Boinee, P., De Angelis, A., Foresti, G.L.: Ensembling classifiers-an application to image data classification from Cherenkov telescope experiment. In: *IEC (Prague)*, pp. 394–398 (2005)

5. Bose, R.P.J.C., Verbeek, E.H.M.W., van der Aalst, W.M.P.: Discovering hierarchical process models using ProM. In: Nurcan, S. (ed.) CAiSE Forum 2011. LNBIP, vol. 107, pp. 33–48. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29749-6_3
6. Casati, F., Shan, M.-C.: Semantic analysis of business process executions. In: Jensen, C.S., et al. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 287–296. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45876-X_19
7. Ceravolo, P., Damiani, E., Torabi, M., Barbon, S.: Toward a new generation of log pre-processing methods for process mining. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNBIP, vol. 297, pp. 55–70. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65015-9_4
8. Alves de Medeiros, A.K., van der Aalst, W.M.P.: Process mining towards semantics. In: Dillon, T.S., Chang, E., Meersman, R., Sycara, K. (eds.) Advances in Web Semantics I. LNCS, vol. 4891, pp. 35–80. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89784-2_3
9. de Medeiros, A.K.A., et al.: An outlook on semantic business process mining and monitoring. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2007. LNCS, vol. 4806, pp. 1244–1255. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76890-6_52
10. van der Aalst, W.M.P., de Medeiros, A.K.A., Weijters, A.J.M.M.: Genetic process mining. In: Ciardo, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536, pp. 48–69. Springer, Heidelberg (2005). https://doi.org/10.1007/11494744_5
11. Diaconis, P.: The Markov chain Monte Carlo revolution. *Bull. Am. Math. Soc.* **46**(2), 179–205 (2009)
12. Dumas, M., Van der Aalst, W.M., Ter Hofstede, A.H.: *Process-Aware Information Systems: Bridging People and Software Through Process Technology*. Wiley, New York (2005)
13. Fazzinga, B., Flesca, S., Furfaro, F., Masciari, E., Pontieri, L.: Efficiently interpreting traces of low level events in business process logs. *Inf. Syst.* **73**, 1–24 (2018)
14. Folleco, A., Khoshgoftaar, T.M., Van Hulse, J., Bullard, L.: Software quality modeling: the impact of class noise on the random forest classifier. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), CEC 2008, pp. 3853–3859. IEEE (2008)
15. Grando, M.A., Schonenberg, M., van der Aalst, W.M.: Semantic process mining for the verification of medical recommendations. In: HEALTHINF, pp. 5–16 (2011)
16. Günther, C.W., van der Aalst, W.M.: Mining activity clusters from low-level event logs. *Beta, Research School for Operations Management and Logistics* (2006)
17. Günther, C.W., Rozinat, A., van der Aalst, W.M.P.: Activity mining by global trace segmentation. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 128–139. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_13
18. Jareevongpiboon, W., Janecek, P.: Ontological approach to enhance results of business process mining and analysis. *Bus. Process. Manag. J.* **19**(3), 459–476 (2013)
19. Leonardi, G., Striani, M., Quaglini, S., Cavallini, A., Montani, S.: Towards semantic process mining through knowledge-based trace abstraction. In: Ceravolo, P., van Keulen, M., Stoffel, K. (eds.) SIMPDA 2017. LNBIP, vol. 340, pp. 45–64. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11638-5_3
20. Li, J., Bose, R.P.J.C., van der Aalst, W.M.P.: Mining context-dependent and interactive business process maps using execution patterns. In: zur Muehlen, M., Su, J. (eds.) BPM 2010. LNBIP, vol. 66, pp. 109–121. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20511-8_10

21. Ma, Y., Guo, L., Cukic, B.: A statistical framework for the prediction of fault-proneness. In: *Advances in Machine Learning Applications in Software Engineering*, pp. 237–263. IGI Global (2007)
22. 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
23. Pérez-Castillo, R., Weber, B., de Guzmán, I.G.R., Piattini, M., Pinggera, J.: Assessing event correlation in non-process-aware information systems. *Softw. Syst. Model.* **13**(3), 1117–1139 (2014)
24. Veiga, G.M., Ferreira, D.R.: Understanding spaghetti models with sequence clustering for ProM. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) *BPM 2009*. LNBIP, vol. 43, pp. 92–103. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_10
25. Weijters, A., van der Aalst, W., Alves de Medeiros, A.: Process mining with the heuristics algorithm. Technical report, BETA Working Paper Series 166, TU Eindhoven (2006)
26. Zhang, J., Zulkernine, M.: Network intrusion detection using random forests. In: *PST*. Citeseer (2005)



Approximate Computation of Alignments of Business Processes Through Relaxation Labelling

Lluís Padró^(✉) and Josep Carmona

Computer Science Department, Universitat Politècnica de Catalunya,
Barcelona, Spain
{padro, jcarmona}@cs.upc.edu

Abstract. A fundamental problem in conformance checking is aligning event data with process models. Unfortunately, existing techniques for this task are either complex, or can only be applicable to restricted classes of models. This in practice means that for large inputs, current techniques often fail to produce a result. In this paper we propose a method to approximate alignments for unconstrained process models, which relies on the use of relaxation labelling techniques on top of a partial order representation of the process model. The implementation on the proposed technique achieves a speed-up of several orders of magnitude with respect to the approaches in the literature (either optimal or approximate), often with a reasonable trade-off on the cost of the obtained alignment.

1 Introduction

Conformance checking is expected to be the fastest growing segment in process mining for the next years¹. The main reason for this forthcoming industrial interest is the promise of having event data and process models aligned, thus increasing the value of process models within organizations [5]. On its core, most conformance checking techniques rely on the notion of *alignment* [1]: given an observed trace σ , query the model to obtain the run γ most similar to σ . The computation of alignments is a computational challenge, since it encompasses the exploration of the model state space, an object that is worst-case exponential with respect to the size of the model or the trace.

Consequently, the process mining field is facing the following paradox: whilst there exist techniques to discover process models arbitrarily large, most of the existing alignment computation techniques will not be able to handle such models. This hampers the widespread applicability of conformance checking in industrial scenarios.

In some situations, one can live with approximations: For instance, when the model must be *enhanced* with the information existing in the event log (e.g.,

¹ <https://www.marketsandmarkets.com/Market-Reports/process-analytics-market-254139591.html>.

performance, decision point analysis), or when one aims to *animate* the model by replaying the log on top of it (two of the most celebrated functionalities of commercial process mining tools). Examples of approximations are *token-replay* techniques [14], which do not guarantee optimality, or the techniques in [16, 17], which do not guarantee replayability in general, but that significantly alleviate the complexity of the alignment computation. The method presented in this paper is of this latter type.

We propose a method that is applied on a partial order representation of the process model [7]. A pre-processing step is then done once on the partial order, to gather information (shortest enabling paths between event activations and computing the behavioral profiles) that is used for aligning traces. We assume this is a plausible scenario in many situations, where the model is well-known and it is admissible to have some pre-processing before of aligning traces. For computing alignments, the method uses *Relaxation Labeling* algorithm to map events in each trace to nodes in the partial order. On a training phase, the weights that guide the relaxation labelling problem are tuned. Once this information is obtained, the approach is ready to be applied in the second phase. It is remarkable that several modes can be considered corresponding to different objectives, e.g., strive for replayability, optimality, or a weighted combination.

Experimental results computed over existing benchmarks show promising speedups in computation time, while still being able to derive reasonable approximations when compared to reference techniques.

The paper is organized as follows: next section provides related work for the problem considered in this paper. Then in Sect. 3 we introduce the background of the paper, necessary for understanding the main content in Sect. 4. Experimental evaluation and tool support is provide in Sect. 5, before concluding the paper.

2 Related Work

The work in [1] proposed the notion of alignment, and developed a technique based on A^* to compute optimal alignments for a particular class of process models. Improvements of this approach have been presented in [20]. Alternatives to A^* have appeared very recently: in the approach presented in [6], the alignment problem is mapped as an *automated planning* instance. Automata-based techniques have also appeared [10, 13].

The work in [17] presented the notion of *approximate* alignment to alleviate the computational demands by proposing a recursive paradigm on the basis of structural theory of Petri nets. In spite of resource efficiency, the solution is not guaranteed to be executable. A follow-up work of [17] is presented in [21], which proposes a trade-off between complexity and optimality of solutions, and guarantees executable results. The technique in [16] presents a framework to reduce a process model and the event log accordingly, with the goal to alleviate the computation of alignments. The obtained alignment, called *macro-alignment* since some of the positions are high-level elements, is expanded based on the information gathered during the initial reduction. Techniques using local search

have recently been also proposed [15]. Decompositional techniques have been presented [11, 19] that instead of computing optimal alignments, they focus on the *decisional problem* of whether a given trace fits or not a process model.

Recently, two different approaches have appeared: the work in [3] proposes using binary decision diagrams to alleviate the computation of alignments. The work in [4], which has the goal of maximizing the synchronous moves of the computed alignments, uses a pre-processing step on the model.

The method of this paper is an alternative to the methods in the literature, useful when computation time and/or memory requirements hamper their applicability, and suboptimal solutions are acceptable. In such a scenario, our approach produces solutions close to the optimum with a much smaller computational cost.

3 Preliminaries

3.1 Petri Nets, Unfoldings and Process Mining

A Process Model defined by a *labeled Petri net system* (or simply *Petri net*) consists of a tuple $N = \langle P, T, F, m_0, m_f, \Sigma, \lambda \rangle$, where P is the set of places, T is the set of transitions (with $P \cap T = \emptyset$), $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation, m_0 is the initial marking, m_f is the final marking, Σ is an alphabet of actions, and $\lambda : T \rightarrow \Sigma \cup \{\tau\}$ labels every transition with an action or as silent. The semantics of Petri nets is given in terms of *firing sequences*. A *marking* is an assignment of a non-negative integer to each place. A transition t is *enabled* in a marking m when all places in its preset $\bullet t \stackrel{\text{def}}{=} \{y \in P \cup T \mid (t, y) \in F\}$ are marked. When a transition t is enabled, it can *fire* by removing a token from each place in $\bullet t$ and putting a token to each place in its postset $t \bullet \stackrel{\text{def}}{=} \{y \in P \cup T \mid (y, t) \in F\}$. A marking m' is *reachable* from m if there is a sequence of firings $\langle t_1 \dots t_n \rangle$ that transforms m into m' , denoted by $m[\langle t_1 \dots t_n \rangle m']$. The set of reachable markings from m_0 is denoted by $[m_0]$. A Petri net is *k-bounded* if no marking in $[m_0]$ assigns more than k tokens to any place. A Petri net is *safe* if it is 1-bounded. In this paper we assume safe Petri nets. A firing sequence $u = \langle t_1 \dots t_n \rangle$ is called a *run* if it can fire from the initial marking: $m_0[u]$; it is called a *full run* if it additionally reaches the final marking: $m_0[u]m_f$. We write $Runs(N)$ for the set of full runs of Petri net N . Given a full run $u = \langle t_1 \dots t_n \rangle \in Runs(N)$, the sequence of actions $\lambda(u) \stackrel{\text{def}}{=} \langle \lambda(t_1) \dots \lambda(t_n) \rangle$ is called a (*model*) *trace* of N .

A *finite and complete unfolding prefix* π of a Petri net N is a finite acyclic net which implicitly represents all the reachable states of N , together with transitions enabled at those states. It can be obtained through unfolding N by successive firings of transitions, under the following assumptions: (a) for each new firing, a fresh transition (called an *event*) is generated; (b) for each newly produced token a fresh place (called a *condition*) is generated. The unfolding is infinite whenever N has an infinite run; however, if N has finitely many reachable states, then the unfolding eventually starts to repeat itself and can be truncated (by identifying a set of *cut-off* events) without loss of information, yielding a finite and complete

prefix. We denote by B , E and $E_{cut} \subseteq E$ the sets of conditions, events and cut-off events of the prefix, respectively. Efficient algorithms exist for building such prefixes [7–9].

In this paper we use *behavioral profiles* [23] to guide the search for alignments.

Definition 1 (Behavioral Profiles [23]). *Let x, y be two transitions of a Petri net N . $x \succ y$ if there exists a run of N where x appears before of y . A pair of transitions (x, y) of a Petri net is in at most one of the following behavioral relation:*

- The strict order relation $x \rightsquigarrow y$, if $x \succ y$ and $y \not\succeq x$
- The exclusiveness order relation $x + y$, if $x \not\succeq y$ and $y \not\succeq x$
- The interleaving order relation $x \parallel y$, if $x \succ y$ and $y \succ x$

Definition 2 (Log, Alignment). *A log over an alphabet Σ is a finite set of words $\sigma \in \Sigma^*$, called log traces. Given a Petri net $N = \langle P, T, F, m_0, m_f, \Sigma, \lambda \rangle$, and a log trace σ , an alignment is a full run of the model $\gamma \in \text{Runs}(N)$ with minimal edit distance to σ , i.e., $\forall \gamma' \in \text{Runs}(N) : \gamma' \neq \gamma \implies \text{dist}(\sigma, \gamma') \geq \text{dist}(\sigma, \gamma)$.*

3.2 Relaxation Labelling Algorithm

Relaxation labelling (RL) is a generic name for a family of iterative algorithms which perform function optimization based on local information, from a constraint satisfaction approach. Although other optimization algorithms could have been used (e.g. genetic algorithms, simulated annealing, or even ILP) we found RL to be suitable to our purposes, given its ability to use models based arbitrary context constraints, to deal with partial information, and to provide a solution even when fed with inconsistent information (though the solution will not necessarily be consistent if that is the case).

Given a set of variables $\mathcal{V} = \{v_1, \dots, v_n\}$, the algorithm goal is to assign a value (label) to each of them. Values for each $v_i \in \mathcal{V}$ are chosen from a finite discrete set of labels $\mathcal{L}(v_i) = \{t_{i_1}, \dots, t_{i_{m_i}}\}$. Variable-label assignments are rewarded or penalized by a set of constraints \mathcal{C} . Each constraint $r \in \mathcal{C}$ has the form:

$$C_r \ (v_i : t_{ij}) \ [(v_{i_1} : t_{i_1 j_1}), \dots, (v_{i_{d_r}} : t_{i_{d_r} j_{d_r}})]$$

where $(v_i : t_{ij})$ is the *target assignment* of the constraint (i.e the assignment that is rewarded or penalized by the constraint), $[(v_{i_1} : t_{i_1 j_1}), \dots, (v_{i_{d_r}} : t_{i_{d_r} j_{d_r}})]$ are the constraint *conditions* (i.e. the assignments of other variables required for the constraint to be satisfied), and C_r is a real value expressing *compatibility* (or *incompatibility* if negative) of the target assignment with respect to the conditions.

Algorithm 1 shows the pseudo-code of the used RL variant (a variety of formulas can be used to compute S_{ij} or $p_{ij}(s + 1)$). See [18] for a summary), where:

- p_{ij} is the current weight for the assignment $(v_i : t_j)$. Assignment weights are normalized so that $\forall i \sum_{j=1}^{m_i} p_{ij} = 1$.

- $Inf(r) = C_r \times p_{i_1 j_1}(s) \times \dots \times p_{i_{d_r} j_{d_r}}(s)$, is the *influence* of constraint r on its target assignment, computed as the product of the current weights (at time step s) of the assignments in the constraint *conditions* (representing *how satisfied* the conditions are in the current context) multiplied by the constraint compatibility value C_r (stating *how compatible* is the target assignment with the context).
- $\mathcal{C}_{ij} \subseteq \mathcal{C}$ is the subset of constraints that have the pair $(v_i : t_j)$ as target assignment.
- S_{ij} is the total support received by pair $(v_i : e_j)$ from all constraints targeting it. Since S_{ij} depends on the conditioning pairs, it will change over time.

```

/* Start in a uniformly distributed labelling P */
P := {{p11...p1m1}, ..., {pn1...pnmn}};
/* Time step counter */
s := 0;
repeat
  /* Compute the support Sij that each label receives from the
  current weights for the labels of the other variables and the
  constraints contributions */
  for each variable vi ∈ V do
    for each label tij ∈ L(vi) do
      Sij := ∑r ∈ Cij Inf(r)
    end
  end
  /* Compute (and re-normalize) weights for each variable label at
  time step s + 1 according to the support they receive */
  for each variable vi ∈ V do
    for each label tij ∈ L(vi) do
      pij(s + 1) :=  $\frac{p_{ij}(s) \times (1 + S_{ij})}{\sum_{k=1}^{m_i} p_{ij}(s) \times (1 + S_{ik})}$ 
    end
  end
  s := s + 1
until no more changes;

```

Algorithm 1. Pseudo code of the RL algorithm.

At each time step, the algorithm updates the weights of each possible labels for each variable. The results are normalized per variable, raising weights for labels with higher support, and reducing them for those with lower support. Advantages of the algorithm are:

- Its expressivity: The problem is stated in terms of assigning *labels* to variables, and a set of constraints between variable-label assignments, allowing to model

any discrete combinatorial problem. The algorithm can deal with any kind of constraints encoding any relevant domain information.

- Its highly local character (each variable can update its label weights given only the state at previous time step), which makes the algorithm highly parallelizable.
- Its flexibility: Total consistency or completeness of constraints is not required.
- Its robustness: It can give an answer to problems without an exact solution (incomplete or partially incompatible constraints, insufficient data, etc.)
- Its complexity. Being n the number of variables, v the average number of possible labels per variable, c the average number of constraints per label, and I the average number of iterations until convergence, the average cost is $n \times v \times c \times I$. Note that some of these factors can be made constants: The algorithm can be stopped if convergence is not reached after a maximum number of iterations. In most problems v and c do not depend on n , or if they do, they can be bounded (e.g. generating constraints only for nearby neighbors instead of all variables, or pre-filtering unlikely values). In general, for problems with a large amount of variables, the complexity can be controlled at the price of reducing expressivity and/or result accuracy, obtaining accurate enough models with linear or quadratic asymptotic costs.

Drawbacks of the algorithm are:

- Found optima are local, and convergence is not guaranteed in the general case.
- Constraints must be designed manually, since they encode the domain knowledge about the problem.
- Constraint weights must be assigned manually and/or optimized on tuning data.

4 Framework to Approximate Alignments

Figure 1 presents an overall description of the framework: A preprocessing step, *(a)* inside the gray box, is executed only once per model to compute the model unfolding, its behavioural profile, and the shortest enabling path between each pair of nodes. Then, it is used as many times as needed to align log traces. The alignment algorithm, *(c)* relaxation labeling, uses weighted constraints *(b)*, and although their weights can simply be set manually, better results are obtained if they are tuned using available training data. The algorithm produces partial alignments without model moves, which are added –if needed– by a completion post-process *(d)*. The weight tuning procedure is exactly the same: The system is run on different combinations of constraint weights on a separate section of the dataset, and the combination producing the best results is chosen to be used on test data (or used in production).

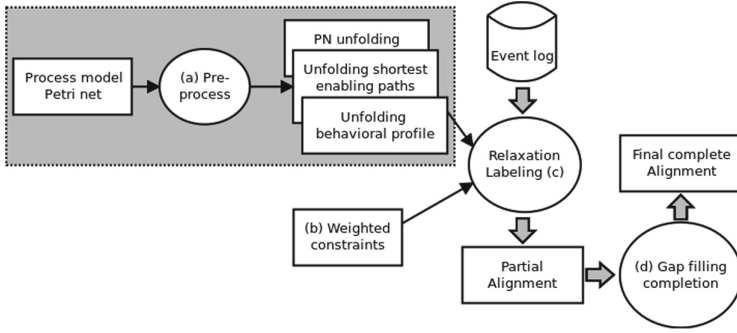


Fig. 1. Overall framework representation.

4.1 Stage 1: Pre-computation of Model Unfolding and Additional Information

We use one of the state-of-the-art techniques to compute an unfolding π of the Petri net [9]. There are two main reasons to use the unfolding instead of the Petri net. First, events in the unfolding correspond to a particular firing of a transition in a Petri net, thus making the correspondence between events in the trace and events in the unfolding meaningful². Second, by being well-structured (e.g., having a clear initial and final node), the computation of alignments is facilitated.

Two types of information between any pair of events in the unfolding are required in our setting: (1) Behavioral relations are used to guide RL in order to reward/penalize particular assignments between events in the trace and unfolding transitions. (2) Shortest enabling paths are necessary for completing the alignment when gaps exist in alignment arising from the solution found by the RL algorithm. Notice that this information is computed only once per model, before aligning each trace in the log.

Behavioral Relations Between Unfolding Events. As it has been pointed out [2,22], not all runs of the Petri net are possible in the complete unfolding, which impacts the behavioral information between events in the unfolding. To amend this, either the unfolding is extended beyond cut-off events so that all relations are visible [2], or the behavioral relations are adapted to consider the discontinuities due to cut-off events [22].

In this paper we opted instead for a pragmatic setting: next to the original unfolding π , a copy π^r where the backward-conflicts branches and loops corresponding to the cut-off events are computed (see Fig. 2). We call π^r *reconnected unfolding*. Notice that, in contrast to the original unfolding, in a reconnected unfolding all the runs of the original Petri net are possible.

² Notice that a transition can correspond to several different firing modes, that depend on the context, which will be represented as different events in the unfolding.

Next, the behavioral profiles (c.f. Definition 1) for both π and π^r are computed. Apart from obtaining the behavioral relations for events, computing these relations both in π and π^r is useful to elicit loop behavior: for two events e_1, e_2 , if $e_1 \not\parallel e_2$ in π , but $e_1 \parallel e_2$ in π^r , then the concurrency of e_1 and e_2 is due to the existence of a loop in the original Petri net³, while if $e_1 \parallel e_2$ in π , then e_1 and e_2 are in a parallel section (which may or may not be inside a loop). These behavioral relations (ordering, exclusiveness, interleaving and loop relations) are then used to assign different constraint weights in the created constraint satisfaction problem instance (see next Section).

Shortest Enabling Paths Between Unfolding Events. Given two events e_1, e_2 in π^r , the *shortest enabling path* is the minimal set of events needed to enable e_2 after e_1 fires. Since we pose the problem as choosing an event (transition) in the unfolding for each event in the trace, the RL algorithm will not suggest new events to be inserted in the trace (i.e. *model moves*). To complete the alignment with required model moves, we fill the gaps in the trace with the shortest enabling path between events, which is precomputed off-line, only once per model.

The length of the shortest enabling path between two nodes is also used to modulate the weight of the constraints (see Sect. 4.2).

4.2 Stage 2: Computation of Mapping Through RL

Given π^r and a trace $\sigma = a_1 \dots a_n \in L$, we post the alignment problem as a *consistent labelling problem* (CLP), which can be solved via suboptimal constraint satisfaction methods, such as RL. We will illustrate how we build our labelling problem, as well as how it is handled by the RL algorithm, with the example M8 model from the dataset described in [12]⁴, and shown in Fig. 2. Below, to avoid ambiguities, we will refer to events in the unfolding as *transitions*.

The CLP is built as follows:

- Each event $a_i \in \sigma$ is a variable v_i for the CLP problem. The set of variables is $\mathcal{V} = \{v_1, \dots, v_n\}$.
- For each variable $v_i \in \mathcal{V}$, we have a set of labels $\mathcal{L}(v_i) = \{e_{i_1}, \dots, e_{i_{m_i}}, \text{NULL}\}$, containing all transitions e_{i_k} in π^r such that $\lambda(e_{i_k}) = a_i$, plus one NULL label to allow for the option to not align a particular event in the trace (a *log move*). Figure 3 shows the aforementioned encoding for the trace *BCGHEFDA* and the M8 model in Fig. 2⁵. Notice that selecting the possible labels for each

³ In case models do not have duplicate labels, the detection of loops can alternatively be performed as it was done in [2].

⁴ <https://data.4tu.nl/repository/uuid:44c32783-15d0-4dbd-af8a-78b97be3de49>.

⁵ Notice that, for the sake of simplicity, the example in Fig. 2 only contains one unfolding event per label. In general several events in the unfolding can have the same label, and our technique handles that general case.

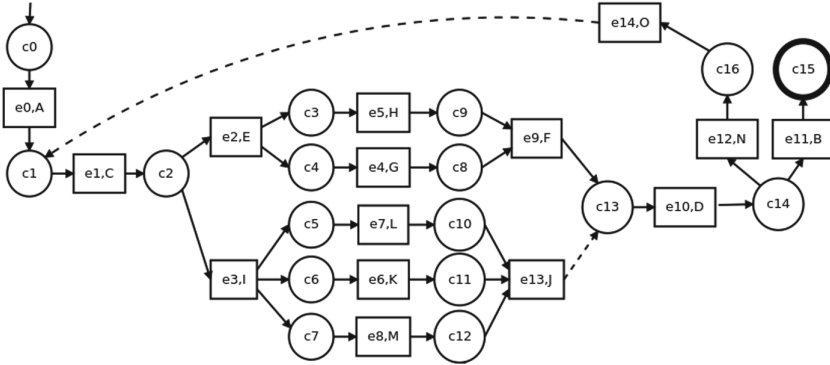


Fig. 2. Reconnected unfolding π^r for model M8. Dashed edges indicate reconnected cut-offs.

variable we can introduce hard constraints (e.g. if an assignment is not possible because of, e.g., data type incompatibility, that value is excluded from $\mathcal{L}(v_i)$, avoiding the need of having to encode this fact as a constraint).

- In this work, we base our constraints on the similarity between the order of the events in the trace and the model, although other kinds of information could be used if available. For instance, a possible constraint for the trace in Fig. 3 and the M8 process model in Fig. 2 could be $+10.0 (v_3 : e4) [(v_2 : e1), (v_4 : e5)]$, stating that the assignment of label $e4$ to variable v_3 gets a positive reward of $+10.0$ from a context where v_2 is assigned label $e1$ and v_4 is assigned label $e5$.

Trace events (variables)	v_1 (B)	v_2 (C)	v_3 (G)	v_4 (H)	v_5 (E)	v_6 (F)	v_7 (D)	v_8 (A)
Possible alignments (labels)	e11	e1	e4	e5	e2	e9	e10	e0
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig. 3. Mapping of the trace $BCGHEFDA$ alignment with model M8 as a consistent labelling problem. Boldface labels indicate the solution selected by the RL algorithm.

To avoid an explosion of the number of constraints, we restricted ourselves to use binary constraints –that is, involving just one target assignment and one condition–, except in the case of the *Deletion* constraint (see below). We now provide a description of the constraints used.

Compatibility Constraints. Each constraint has a *compatibility value* that may be either positive (to reward consistent assignments) or negative (to penalize inconsistent combinations). The weight for each constraint is tuned experimentally.

In what follows, $d(v_i, v_j) \stackrel{\text{def}}{=} |i - j|$ refers to the distance between events a_i and a_j in σ , and $d(e_i, e_j)$ corresponds to the length of a shortest enabling path between transitions e_i and e_j in π^r .

Constraint Patterns: For each combination of two possible assignments ($v_i : e_p$), ($v_j : e_q$), we create the following constraint instances:

$$\begin{aligned} C_r \quad (v_i : e_p) \quad [(v_j : e_q)] \\ C_r \quad (v_j : e_q) \quad [(v_i : e_p)] \end{aligned}$$

for each of the following cases that are applicable. The compatibility value C_r depends on each case:

- *Right order.* If v_i precedes v_j in σ (i.e., $i < j$), and $e_p \rightsquigarrow e_q$ in π^r , C_r is positive, and inversely proportional to $|d(v_i, v_j) - d(e_p, e_q)|$, rewarding assignments in the right order, with higher rewards for closer assignments.
- *Wrong order.* If v_i follows v_j in σ (i.e., $i > j$), and $e_p \rightsquigarrow e_q$ in π^r , C_r is negative, penalizing assignments with crossed ordering in the trace with respect to the model.
- *Exclusive.* If v_i and v_j co-occur in the trace but $e_p + e_q$ in π^r , C_r is negative, penalizing assignments that should not happen in the same trace.
- *Parallel.* If v_i and v_j co-occur in the trace, and $e_p \parallel e_q$ in π , indicating the presence of a parallel section, C_r is positive, and inversely proportional to $|d(v_i, v_j) - d(e_p, e_q)|$, rewarding this combination in any order, with higher rewards for closer assignments.
- *Loop.* If v_i and v_j co-occur in the trace, $e_p \parallel e_q$ in π^r , and $e_p \not\parallel e_q$ in π indicating that the interleaving is due to the presence of a loop, C_r is positive, which allows the repetition and alternation of looped events.

Deletion. Also, for each combination of three possible assignments ($v_{i-1} : e_m$), ($v_i : e_p$), ($v_{i+1} : e_q$) such that $1 < i < n$ (i.e. three consecutive events in the trace) if the shortest enabling path from e_m to e_q via e_p in π^r is longer than the shortest enabling path from e_m to e_q not crossing e_p , we create the constraint instance:

$$C_r \quad (v_i : e_p) \quad [(v_{i-1} : e_m), (v_{i+1} : e_q)]$$

where C_r is negative. This constraint penalizes the alignment of an event if that would require more model moves (and thus a higher cost) than its deletion.

Figure 4 shows examples of how these patterns are instantiated in the M8 example. Note that the high negative weight of the *wrong order* constraints will cause that in every pair, at least one of the variables (that with less positive contribution from others) will end up selecting any other label (which in this case will be the NULL label). Weights for *right order* constraints are inversely proportional to $|d(v_i, v_j) - d(e_p, e_q)|$. The other constraints in the example use a constant value.

It is important to remark that a single constraint does not determine the alignment chosen for a particular event. All constraints affecting the assignment (v_i, e_j) are combined in S_{ij} . The re-normalization of the label weights for each variable ensures that there will always be one value selected: even if all values for a variable had a negative support, the weight for the one with less negative S_{ij} would be increased. In our case, we have the NULL value, which has neither penalization nor reward ($S_{ij} = 0$) causing its weight to be raised when all the other possible values have negative support.

The algorithm stops when convergence is reached –i.e. no more changes in the weight assignment–. Typical solutions consist of weight assignments of 1 for one label in each variable, and zero for the rest. However, if constraints are incomplete or contradictory, the final state may be a uniform distribution among a subset of values for some variables. Also, since the optimized cost function depends on the constraints, convergence is not theoretically guaranteed (since they may be incomplete or contradictory), although empirical results show that –if constraints are properly defined as it is the case of our formalization– the algorithm normally converges.

As described in Sect. 3.2 the complexity of RL is $n \times v \times c \times I$. In our particular trace alignment problem, v is a small constant (about 2 or 3 possible labels per variable). Since we generate constraints for every pair of trace events, the number of constraints per variable c is proportional to n . The number of required iterations I is in the order of a few dozens, though a safety stop is forced after 500 iterations. Thus, in our case, the complexity is $\mathcal{O}(n \times c \times K) = \mathcal{O}(n^2)$ (though it could be reduced to linear limiting the created constraints to only nearby neighbor events).

4.3 Stage 3: Generation of Approximate Alignment

The CLP solved via RL will produce a partial alignment, where some trace events will be assigned to some transitions in the unfolding, and some events will be assigned the NULL label (see Fig 3). If the solution is consistent, it represents synchronous moves (events in the trace are mapped to a transition in the unfolding) and log moves (events in the trace are assigned to NULL). It may only lack model moves, i.e., necessary transitions in the unfolding to recover a full model run.

The approach used to add the needed model moves is to simulate the partial trace on the Petri net, until a mismatch is found (notice that this is a deterministic procedure, since unfolding transitions are unique). Assuming the RL solution alignments and deletions are correct, the mismatch can only be caused

Constraint examples		
<i>Right order</i>	+8.3 ($v_2 : e1$)	$[(v_3 : e4)]$
	+12.5 ($v_2 : e1$)	$[(v_6 : e9)]$
	+25 ($v_6 : e9$)	$[(v_7 : e10)]$
<i>Wrong Order</i>	-500 ($v_1 : e11$)	$[(v_2 : e1)]$
	-500 ($v_4 : e5$)	$[(v_5 : e2)]$
	-500 ($v_7 : e10$)	$[(v_8 : e0)]$
<i>Parallel</i>	+5 ($v_3 : e4$)	$[(v_4 : e5)]$
<i>Deletion</i>	-200 ($v_5 : e2$)	$[(v_4 : e5), (v_6 : e9)]$

Fig. 4. Some example constraint pattern instantiations for the M8 alignment example.

by a missing event in the trace. Thus, the shortest enabling path (previously computed) connecting the transition where the mismatch was detected and the transition corresponding to the next event in the trace is inserted at this point, and the simulation is continued. Note that this completion procedure is also able to re-insert events that were wrongly deleted by the RL algorithm. However, if the RL solution contains crucial errors (i.e. alignment of an event that should have been deleted), the resulting alignment may not be fitting.

To handle the insertions at the beginning or end of the trace, we add two *phantom* events, one at the beginning and one the end of the trace, respectively aligned to the initial and final states. In this way, the simulation will detect if there are missing events before the first trace element or after the last one.

Figure 5 shows an example of the results of the completion process, i.e., the technique computes the run *ACEGHFDB*, which is at edit distance 6 (counted as number of insertions and deletions) for the input trace *BCGHEFDA*.

Variables (trace events)	<i>v</i> ₁ (B)		<i>v</i> ₂ (C)		<i>v</i> ₃ (G)	<i>v</i> ₄ (H)	<i>v</i> ₅ (E)	<i>v</i> ₆ (F)	<i>v</i> ₇ (D)	<i>v</i> ₈ (A)	
Selected (or inserted) labels	NULL	<i>e0</i> (A)	e1	<i>e2</i> (E)	e4	e5	NULL	e9	e10	NULL	<i>e11</i> (B)

Fig. 5. Complete alignment for example in Fig. 3, after adding necessary insertions to make the trace fitting. Boldface labels correspond to the alignment produced by RL. Transitions in italics are model moves inserted by the completion postprocess.

5 Experiments and Tool Support

To evaluate the performance of our approach, we resorted to datasets previously used in the state-of-the-art to test the performance of alignment techniques [12, 16, 17]. Some of these benchmarks are either very large, and/or contain loops and duplicate activities in the model. We also applied the tool to a real-world case: We used the Inductive Miner [10] (with default parameters) to extract a model for BPIC 2017 loan application data⁶, and then we aligned it with the whole set of traces. Source code for our tool is available at <https://github.com/lluisp/RL-align>.

Since RL results largely depend on the constraint compatibility values, we used part of the data as a development set to tune the constraint weights, and we evaluated on the rest. We compared the solution of our approach with a reference solution: Optimal A* alignment by ProM for the models where it is available, backing off to an approximate method (ILPSDP, see [15]) when ProM failed to process the model trace file due to memory or time limitations. The used partition and some statistics about the models and traces can be found in

⁶ <https://data.4tu.nl/repository/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b>.

Table 1. Statistics about dataset used in the experiments.

	Model	#places	Trace length			#traces	Reference alignment			Preprocess CPU time (s)	
			Avg	Max	Min		Avg. cost	Avg. fitness	Method	Paths	BPs
Tuning	M1	40	13.1	37	8	500	5.8	0.65	ProM	1	2
	M3	108	35.9	217	10	500	8.9	0.79	ProM	4	26
	M5	35	34.0	71	27	500	14.7	0.64	ProM	1	2
	M7	65	37.6	147	20	500	26.3	0.49	IPLSDP	1	5
	M9	47	44.3	216	16	500	21.3	0.61	ProM	1	6
	ML1	27	28.9	123	11	500	17.9	0.51	ProM	1	4
	ML3	45	26.4	194	8	500	22.9	0.35	ProM	1	3
	ML5	159	42.0	595	12	500	30.0	0.55	IPLSDP	12	53
	prAm6	347	31.6	41	19	1,200	4.1	0.90	ProM	133	829
	prCm6	317	42.8	59	15	500	29.3	0.51	IPLSDP	95	394
	prEm6	277	98.7	116	80	1,200	4.0	0.96	IPLSDP	64	153
	prGm6	357	143.0	159	124	1,200	26.3	0.83	IPLSDP	136	134
	<i>TOTAL</i>		<i>59.3</i>	<i>595</i>	<i>8</i>	<i>8,100</i>	<i>16.0</i>	<i>0.71</i>		<i>447</i>	<i>1,611</i>
Evaluation	M2	34	17.6	52	14	500	10.3	0.56	ProM	1	2
	M4	36	26.8	176	8	500	22.7	0.35	ProM	1	6
	M6	69	53.3	125	42	500	42.3	0.46	IPLSDP	1	5
	M8	17	16.5	109	8	500	7.3	0.65	ProM	1	1
	M10	150	58.2	240	30	500	42.7	0.47	IPLSDP	10	28
	ML2	165	87.4	582	27	500	80.9	0.30	IPLSDP	14	33
	ML4	36	28.1	89	17	500	25.6	0.34	ProM	1	2
	prBm6	317	41.5	59	14	1,200	0.0	1.00	ProM	96	388
	prDm6	529	248.4	271	235	1,200	3.6	0.99	IPLSDP	341	100
	prFm6	362	240.6	245	234	1,200	36.7	0.86	IPLSDP	107	34
	<i>TOTAL</i>		<i>109.9</i>	<i>582</i>	<i>8</i>	<i>7,100</i>	<i>24.0</i>	<i>0.70</i>		<i>570</i>	<i>599</i>
Total			83.0	595	8	15,200	19.8	0.71		1,017	2,210
Realistic	BPIC2017	280	38.1	180	10	31,509	38.2	0.10	IPLSDP	27	1,479

Table 1. Cost is computed as edit distance (number of log moves plus number of model moves). Fitness is computed as the ratio of sync moves over the length of the trace. The *average cost* and *average fitness* columns show the average cost/fitness per trace over the whole log. Last two columns show the CPU time required to precompute behavioural profiles and shortest enabling paths.

The tuning procedure consisted on a grid search of weights for each constraint type. Since *Loop* and *Parallel* use the same weight (the former as a constant, the latter in inverse proportion to the distance), we have 5 weights to set. We explored between 6 and 8 possible values for each –totalling over 16,000 combinations– and selected the weight combinations that maximized the desired measure over the tuning dataset.

Tables 2 and 3 show the results for the performed experiments. We report the percentage of cases where a fitting alignment was found, in how many of those

Table 2. Results obtained in scenario 1 (Maximize alignment F_1 score)

	Model	% fitting	% same cost	Obtained alignment				CPU time (sec)		
				Avg. cost	Δ with reference	Avg. fitness	Δ with reference	RL	ILPSDP	ProM
Tuning	M1	99.4	81.9	6.0	0.3	0.64	-0.01	1	23	4
	M3	90.8	75.8	8.9	1.3	0.78	-0.02	5	234	142
	M5	44.4	49.5	14.9	1.7	0.64	-0.01	6	59	587
	M7	45.2	25.2	16.8	-1.7	0.61	0.06	5	103	-
	M9	57.0	62.8	16.1	2.0	0.61	-0.02	10	123	51
	ML1	44.8	47.3	14.9	3.8	0.53	-0.03	7	67	18
	ML3	43.8	13.2	46.3	27.3	0.30	-0.08	7	89	61
	ML5	87.3	51.6	20.3	3.9	0.60	0.01	23	688	-
	prAm6	100.0	91.5	4.3	0.2	0.90	-0.003	5	822	58
	prCm6	89.8	21.6	27.0	-2.5	0.54	0.04	4	476	-
	prEm6	100.0	100.0	4.0	0.0	0.96	0.00	21	3,145	-
prGm6	0.0	-	-	-	-	-	114	7,757	-	
<i>TOTAL</i>	<i>66.8</i>	<i>71.2</i>	<i>11.7</i>	<i>1.6</i>	<i>0.75</i>	<i>-0.001</i>	<i>208</i>	<i>13,586</i>	-	
Test	M2	97.6	55.1	11.0	0.8	0.55	-0.004	1	30	20
	M4	54.8	22.3	31.4	14.6	0.34	-0.05	5	99	29
	M6	4.4	4.5	21.7	-7.0	0.68	0.11	8	165	-
	M8	62.6	70.3	6.5	1.9	0.68	-0.03	2	19	3
	M10	22.4	16.1	32.0	-1.6	0.59	0.08	11	411	-
	ML2	52.2	4.6	54.4	-9.0	0.61	0.26	61	1,743	-
	ML4	28.0	6.4	30.8	11.0	0.33	-0.05	4	63	579
	prBm6	100.0	100.0	0.0	0.0	1.00	0.00	5	856	54
	prDm6	61.0	0.0	42.5	39.1	0.84	-0.15	177	34,653	-
	prFm6	57.2	0.0	9.1	-27.8	0.96	0.10	159	20,631	-
<i>TOTAL</i>	<i>59.5</i>	<i>42.3</i>	<i>18.0</i>	<i>3.2</i>	<i>0.79</i>	<i>0.003</i>	<i>433</i>	<i>58,670</i>	-	
Realistic	BPIC2017	99.9	0.4	43.8	5.6	0.15	0.05	2,091	8,702	-

the solution had the same cost than the reference approach (ProM or ILPSDP), the average cost and fitness of the alignments, and their differences with the cost and fitness achieved by the reference approach. In some cases the cost difference is negative (and/or the fitness difference is positive) showing that RL obtained better solutions than ILPSDP.

We also report the required CPU time to process the trace file for each model. Dashes in CPU time columns for ProM correspond to files were ProM run out of memory (using a 8 GB Java heap) or did not end after 8 h (wall clock time). Reported CPU times exclude time required to preprocess each model computing two behavioural profiles (original and reconnected unfolding) and shortest enabling paths for all event pairs (see Table 1). Note that the preprocessing is performed only once per model, so it is amortized in the long run when the number of aligned traces is large enough.

Scenario 1: Maximize Quality of Obtained Alignments. Our first scenario is selecting weights that get better alignments, even this may cause a lower percentage of cases with a fitting solution. In order to keep a balance between the quality of the alignments and the number of solved cases, we measure *precision* ($P = \#sync / (\#sync + \#log)$, maximized when there are no log moves) and *recall* ($R = \#sync / (\#sync + \#model)$, maximized when there are no model moves), and we aim at maximizing their harmonic mean, or F_1 score ($F_1 = 2PR / (P + R)$). The weight combination obtaining higher F_1 on tuning data is: *Right Order* = +15, *Wrong Order* = -100, *Exclusive* = -300, *Deletion* = -20, *Parallel/Loop* = +5.

Results of this configuration both on tuning and test data are shown in Table 2.

Scenario 2: Maximize Number of Aligned Traces. A second configuration choice consists of selecting the weights that maximize the number of fitting alignments, even if they have a higher cost. The weight combination obtaining a higher percentage of fitting alignments on tuning data is: *Right Order* = +5, *Wrong Order* = -500, *Exclusive* = -400, *Deletion* = -300, *Parallel/Loop* = +5.

Results of this configuration both on tuning and test data are shown in Table 3.

Discussion. Selecting constraint weights that maximize the percentage of fitting traces (scenario 2) results on large negative values for constraints penalizing inconsistent assignments (i.e. *Wrong Order*, *Exclusive*, and *Deletion*), which create a larger number of NULL assignments. Thus, the obtained alignments will contain more deletions (including wrong deletions of events that could have been aligned), creating gaps that will be filled by the completion post-process, solving more cases with a fitting alignment, though more likely to differ from the original trace, and thus with a higher cost.

On the other hand, when selecting weights that maximize F_1 score of the obtained solution (scenario 1), milder penalization values are selected. Thus, less events are deleted, causing less alignments to be fitting (a single wrongly aligned event can cause the whole trace to become non-fitting), but for those that are, the cost is closer to the reference (since the alignment does not discard trace events unless there is a strong evidence supporting that decision).

It is interesting to note that the proposed algorithm allows us to choose the desired trade-off between the percentage of fitting alignments and the quality of the obtained solutions. Moreover, it is also worth remarking that we tuned the weights for the dataset as a whole, but that they could be optimized per-model, obtaining configurations best suited for each model, if our use case required so.

Regarding computing time, the polynomial cost of the algorithm offers competitive execution times, making it suitable for real-time conformance checking, and feasible to explore configuration space to customize the weights to specific use cases, even on large models. Specifically, our computation times are about

Table 3. Results obtained in scenario 2 (Maximize number of aligned traces)

	Model	% fitting	% same cost	Obtained alignment				CPU time (sec)		
				Avg. cost	Δ with reference	Avg. fitness	Δ with reference	RL	ILPSDP	ProM
Tuning	M1	100.0	58.8	7.5	1.7	0.59	-0.06	1	23	4
	M3	89.4	62.9	9.9	2.2	0.76	-0.03	5	234	142
	M5	100.0	11.4	21.7	7.0	0.55	-0.10	3	59	587
	M7	99.8	11.8	30.7	4.6	0.49	-0.01	3	103	-
	M9	58.4	18.8	27.4	12.4	0.42	-0.21	5	123	51
	ML1	69.4	19.3	26.2	10.3	0.36	-0.16	3	67	18
	ML3	49.2	8.1	46.5	26.3	0.27	-0.10	2	89	61
	ML5	86.7	13.7	34.1	16.8	0.36	-0.22	24	688	-
	prAm6	100.0	77.1	5.5	1.4	0.88	-0.02	4	822	58
	prCm6	100.0	4.4	61.3	32.0	0.17	-0.33	3	476	-
	prEm6	100.0	100.0	4.0	0.0	0.96	0.00	40	3,145	-
prGm6	98.9	4.5	35.3	9.1	0.78	-0.05	65	7,757	-	
<i>TOTAL</i>	<i>90.8</i>	<i>42.1</i>	<i>22.0</i>	<i>7.8</i>	<i>0.66</i>	<i>-0.05</i>	<i>158</i>	<i>13,586</i>	-	
Test	M2	100.0	21.4	15.0	4.7	0.44	-0.12	1	30	20
	M4	60.6	11.9	35.0	16.2	0.30	-0.08	2	99	29
	M6	63.6	4.1	37.7	0.5	0.51	0.02	5	165	-
	M8	62.0	59.7	7.0	2.4	0.66	-0.05	1	19	3
	M10	73.2	4.1	57.3	18.0	0.35	-0.13	6	411	-
	ML2	85.4	4.0	65.8	-9.6	0.57	0.26	45	1,743	-
	ML4	54.6	0.4	44.1	20.1	0.14	-0.20	2	63	579
	prBm6	100.0	100.0	0.0	0.0	1.00	0.00	8	856	54
	prDm6	99.6	0.0	57.2	53.6	0.80	-0.19	258	34,653	-
	prFm6	100.0	5.2	35.0	-1.7	0.87	0.01	160	20,631	-
<i>TOTAL</i>	<i>85.8</i>	<i>26.9</i>	<i>33.4</i>	<i>12.8</i>	<i>0.70</i>	<i>-0.05</i>	<i>488</i>	<i>58,670</i>	-	
Realistic	BPIC2017	100.0	0.0	40.3	2.2	0.06	-0.04	1,576	8,702	-

two orders of magnitude smaller than those offered by ILPSDP and ProM, as presented in Tables 2 and 3.

Our tool also performs well on BPIC 2017 real-world data, achieving results comparable to other state-of-the-art methods, and solving them in a shorter time (although the speed-up is not as large in this case).

We must remark that ProM offers optimal solutions (when computational resources are enough), while relaxation labeling does not. Also, even ILPSDP is also suboptimal, it produces a fitting alignment for all cases, while RL may produce non-fitting solutions for some traces. However, we believe that our approach can be used as fast preprocess to obtain accurate enough suboptimal alignments, before resorting to more complex and computationally expensive approaches. RL solutions, either fitting or not, can also be useful as heuristic information to guide optimal search algorithms such as A*.

6 Conclusions and Future Work

We presented a flexible approach to align log traces with a process model. The used problem representation allows a trade-off between amount of solved cases

and quality of the obtained solutions. The behaviour can be customized to particular use cases tuning the weights of the used constraints. Weights can be optimized for a whole dataset (as in presented scenarios 1 and 2), but better results can be obtained if they are optimized for each model, which may be useful for some use cases.

The algorithm requires one-time preprocessing to compute model unfolding, behavioural profile, and shortest enabling paths. Once this is done, any number of traces can be aligned in linear time, with a CPU time orders of magnitude smaller than other state-of-the-art methods. The obtained results show that the method is able to achieve competitive alignments with reasonable costs.

Further research lines include exploring higher-order constraints that allow the algorithm to use more fine-grained context information, and use the results as heuristic information to guide optimal search algorithms.

Acknowledgments. This work has been supported by MINECO and FEDER funds under grant TIN2017-86727-C2-1-R.

References

1. Adriansyah, A.: Aligning observed and modeled behavior. Ph.D. thesis, Technische Universiteit Eindhoven (2014)
2. Armas-Cervantes, A., Baldan, P., Dumas, M., García-Bañuelos, L.: Behavioral comparison of process models based on canonically reduced event structures. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) BPM 2014. LNCS, vol. 8659, pp. 267–282. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10172-9_17
3. Bloemen, V., van de Pol, J., van der Aalst, W.M.P.: Symbolically aligning observed and modelled behaviour. In: 18th International Conference on Application of Concurrency to System Design, ACS D, Bratislava, Slovakia, 25–29 June, pp. 50–59 (2018)
4. Bloemen, V., van Zelst, S.J., van der Aalst, W.M.P., van Dongen, B.F., van de Pol, J.: Maximizing synchronization for aligning observed and modelled behaviour. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 233–249. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_14
5. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance Checking - Relating Processes and Models. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-99414-7>
6. de Leoni, M., Marrella, A.: Aligning real process executions and prescriptive process models through automated planning. *Expert Syst. Appl.* **82**, 162–183 (2017)
7. Esparza, J., Römer, S., Vogler, W.: An improvement of McMillan’s unfolding algorithm. *Form. Methods Syst. Des.* **20**(3), 285–310 (2002)
8. Khomenko, V., Koutny, M.: Towards an efficient algorithm for unfolding Petri nets. In: Larsen, K.G., Nielsen, M. (eds.) CONCUR 2001. LNCS, vol. 2154, pp. 366–380. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44685-0_25
9. Khomenko, V., Koutny, M., Vogler, W.: Canonical prefixes of Petri net unfoldings. *Acta Inf.* **40**(2), 95–118 (2003)
10. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Scalable process discovery and conformance checking. *Softw. Syst. Model.* **17**(2), 599–631 (2018)

11. Munoz-Gama, J., Carmona, J., Van Der Aalst, W.M.P.: Single-entry single-exit decomposed conformance checking. *Inf. Syst.* **46**, 102–122 (2014)
12. (Jorge) Munoz-Gama, J.: Conformance checking in the large (BPM 2013) (2013)
13. Reißner, D., Conforti, R., Dumas, M., La Rosa, M., Armas-Cervantes, A.: Scalable conformance checking of business processes. In: Panetto, H., et al. (eds.) OTM 2017. LNCS, vol. 10573, pp. 607–627. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69462-7_38
14. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* **33**(1), 64–95 (2008)
15. Taymouri, F.: Light methods for conformance checking of business processes. Ph.D. thesis, Universitat Politècnica de Catalunya (2018)
16. Taymouri, F., Carmona, J.: Model and event log reductions to boost the computation of alignments. In: Ceravolo, P., Guetl, C., Rinderle-Ma, S. (eds.) SIMPDA 2016. LNBIP, vol. 307, pp. 1–21. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74161-1_1
17. Taymouri, F., Carmona, J.: A recursive paradigm for aligning observed behavior of large structured process models. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 197–214. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_12
18. Torras, C.: Relaxation and neural learning: points of convergence and divergence. *J. Parallel Distrib. Comput.* **6**, 217–244 (1989)
19. van der Aalst, W.M.P.: Decomposing Petri nets for process mining: a generic approach. *Distrib. Parallel Databases* **31**(4), 471–507 (2013)
20. Dongen, B.F.: Efficiently computing alignments. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 197–214. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_12
21. van Dongen, B., Carmona, J., Chatain, T., Taymouri, F.: Aligning modeled and observed behavior: a compromise between computation complexity and quality. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 94–109. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_7
22. Weidlich, M., Elliger, F., Weske, M.: Generalised computation of behavioural profiles based on Petri-net unfoldings. In: Bravetti, M., Bultan, T. (eds.) WS-FM 2010. LNCS, vol. 6551, pp. 101–115. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19589-1_7
23. Weidlich, M., Mendling, J., Weske, M.: Efficient consistency measurement based on behavioral profiles of process models. *IEEE Trans. Softw. Eng.* **37**(3), 410–429 (2011)



Metaheuristic Optimization for Automated Business Process Discovery

Adriano Augusto^{1,2}(✉), Marlon Dumas², and Marcello La Rosa¹

¹ University of Melbourne, Melbourne, Australia
{a.augusto,marcello.larosa}@unimelb.edu.au

² University of Tartu, Tartu, Estonia
marlon.dumas@ut.ee

Abstract. The problem of automated discovery of process models from event logs has been intensely investigated in the past two decades, leading to a range of approaches that strike various trade-offs between accuracy, model complexity, and execution time. A few studies have suggested that the accuracy of automated process discovery approaches can be enhanced by using metaheuristic optimization. However, these studies have remained at the level of proposals without validation on real-life logs or they have only considered one metaheuristic in isolation. In this setting, this paper studies the following question: To what extent can the accuracy of automated process discovery approaches be improved by applying different optimization metaheuristics? To address this question, the paper proposes an approach to enhance automated process discovery approaches with metaheuristic optimization. The approach is instantiated to define an extension of a state-of-the-art automated process discovery approach, namely Split Miner. The paper compares the accuracy gains yielded by four optimization metaheuristics relative to each other and relative to state-of-the-art baselines, on a benchmark comprising 20 real-life logs. The results show that metaheuristic optimization improves the accuracy of Split Miner in a majority of cases, at the cost of execution times in the order of minutes, versus seconds for the base algorithm.

1 Introduction

The problem of automatically discovering business process models from event logs has been intensely studied in the past two decades. Research in this field has led to a wide range of Automated Process Discovery Approaches (APDAs) that strike various trade-offs between accuracy, model complexity, and execution time [7].

A few studies have suggested that the accuracy of APDAs can be enhanced by applying optimization metaheuristics. Early studies in this direction considered population-based metaheuristics, chiefly genetic algorithms [10, 14]. These heuristics are computationally heavy, requiring execution times in the order of

hours to converge when applied to real-life logs [7]. Another work has considered single-solution-based metaheuristics such as simulated annealing [15,21], which are less computationally demanding. However, these latter studies have remained at the level of proposals without validation on real-life logs and comparison of trade-offs between alternative heuristics.

In this setting, this paper studies the following question: *to what extent can the accuracy of APDAs be improved by applying single-solution-based metaheuristics?* To address this question, we propose a flexible approach to enhance APDAs by applying different optimization metaheuristics. The core idea is to perturb the intermediate representation of event logs used by the majority of the available APDAs, namely the Directly-follows Graph (DFG). The paper specifically considers perturbations that add or remove edges with the aim of improving fitness or precision, and in a way that allows the underlying APDA to discover a process model from the perturbed DFG. An instantiation of our approach is defined for a state-of-the-art APDA, namely Split Miner.

Using a benchmark of 20 real-life logs, the paper compares the accuracy gains yielded by four optimization metaheuristics relative to each other and relative to state-of-the-art APDAs. The experimental evaluation also considers the impact of metaheuristic optimization on model complexity measures as well as on execution times.

The next section gives an overview of APDAs and optimization metaheuristics. Section 3 presents the proposed metaheuristic optimization approach. Section 4 reports on the empirical evaluation and Sect. 5 draws conclusions and future work directions.

2 Background and Related Work

In this section, we give an overview of existing approaches to automated process discovery, followed by an introduction to optimization metaheuristics in general, and their application to automated process discovery in particular.

2.1 Automated Process Discovery

The execution of business processes is often recorded in the form of *event logs*. An event log is a collection of event records produced by individual instances (i.e. cases) of the process. The goal of automated process discovery is to generate a process model that captures the behavior observed in or implied by an *event log*. To assess the goodness of a discovered process model, four quality dimensions are used [23]: fitness, precision, generalization, and complexity. *Fitness* (a.k.a. recall) measures the amount of behavior observed in the log that is captured by the model. A perfectly fitting process model is one that recognizes every trace in the log. *Precision* measures the amount of behavior captured in the process model that is observed in the log. A perfectly precise model is one that recognizes only traces that are observed in the log. *Generalization* measures to what extent the process model captures behavior that, despite not being observed in the log,

is implied by it. Finally, *complexity* measures the understandability of a process model, and it is typically measured via size and structural measures. In this paper, we focus on fitness, precision, and F-score (the harmonic mean of fitness and precision).

A recent comparison of state-of-the-art APDAs [7] showed that an approach capable of consistently discovering models with the best fitness-precision trade-off is currently missing. The same study showed, however, that we can obtain consistently good trade-offs by hyperparameter-optimizing some of the existing APDAs based on DFGs – Inductive Miner [19], Structured Heuristics Miner [6], Fodina [24], and Split Miner [8]. These algorithms have a hyperparameter to tune the amount of filtering applied when constructing the DFG. Optimizing this and other hyperparameters via greedy search [7], local search strategies [11], or sensitivity analysis techniques [20], can greatly improve the accuracy of the discovered process models. Accordingly, in the evaluation reported later we use a hyperparameter-optimized version of Split Miner as one of the baselines.

2.2 Optimization Metaheuristics

The term *optimization metaheuristics* refers to a parameterized algorithm, which can be instantiated to address a wide range of optimization problems. Metaheuristics are usually classified into two broad categories [9]: (i) *single-solution-based metaheuristics*, or S-metaheuristics, which explore the solution space one solution at a time starting from a single initial solution of the problem; and (ii) *population-based metaheuristics*, or P-metaheuristics, which explored a population of solutions generated by mutating, combining, and/or improving previously computed solutions. Single-solution based metaheuristics tend to converge faster towards an optimal solution (either local or global) than P-metaheuristics, since the latter by dealing with a set of solutions require more time to assess and improve the quality of each single solution. P-metaheuristics are more computationally heavy but they are more likely to escape local optima. An exhaustive discussion on all available metaheuristics is beyond the scope of this paper, in the following we focus only on the S-metaheuristics that we explore in our approach: iterated local search, tabu search, and simulated annealing.

Iterated Local Search [22] starts from a (random) solution and explores the neighbouring solutions (i.e. solutions obtained by applying a perturbation) in search of a better one. When a better solution cannot be found, it perturbs the current solution and starts again. The perturbation is meant to avoid local optimal solutions. *Tabu Search* [16] is a memory-driven local search. Its initialization includes a (random) solution and three memories. The short-term memory keeps track of recent solutions and prohibits to revisit them. The intermediate-term memory contains criteria driving the search towards the best solutions. The long-term memory contains characteristics that have often been found in many visited solutions, to avoid revisiting similar solutions. Using these memories, the neighbourhood of the initial solution is explored and a new solution is selected accordingly. *Simulated Annealing* [17] is based on the concepts of Temperature (T , a parameter choose arbitrarily) and Energy (E , the objective function to

minimize). At each iteration the algorithm explores (some of) the neighbouring solutions and compares their energies with the one of the current solution. This latter is updated if the energy of a neighbour is lower, or with a probability that is function of T and the energies of the current and candidate solutions (usually $e^{-\frac{|E_1 - E_2|}{T}}$). The temperature drops over time, thus reducing the chance of updating the current solution with a higher-energy one. The algorithm ends when a criterion is met (e.g. energy below a threshold or $T = 0$).

2.3 Optimization Metaheuristics in Automated Process Discovery

Metaheuristic optimization has been considered in a few previous studies on automated process discovery. An early attempt to apply P-metaheuristics for automated process discovery was the Genetic Miner proposed by De Medeiros [14], subsequently overtaken by the Evolutionary Tree Miner [10]. In this latter approach, an evolutionary algorithm is used on top of process trees (i.e. a block-structured representation of a process model). Other applications of P-metaheuristics to automated process discovery are based on the imperialist competitive algorithms [3] and particle swarm optimization [12]. The main limitation P-metaheuristics in this context is that they are computationally heavy due to the cost of constructing a solution (i.e. process model) and evaluating its accuracy. This leads to execution times in the order of hours to converge to a solution, which on the end is comparable to that obtained by state-of-the-art algorithms that do not rely on optimization metaheuristics [7].

Only a handful of studies have considered the use of S-metaheuristics in this setting, specifically simulated annealing [15, 21]. However, these latter proposals are preliminary and have not been compared against state-of-the-art approaches on real-life logs.

3 Approach

This section outlines our approach for extending APDAs by means of S-metaheuristics (cf. Sect. 2). First, we give an overview of the approach and its components. Next, we discuss the adaptation of the metaheuristics to the problem of process discovery. Finally, we describe an instantiation of the approach for Split Miner.

3.1 Preliminaries

An APDA takes as input an event log. This log is transformed into an intermediate representation from which a model is derived. In many APDAs, the intermediate representation is the DFG, which is represented as a numerical matrix as formalized below.

Definition 1 [Event Log]. *Given a set of activities \mathcal{A} , an event log \mathcal{L} is a multiset of traces where a trace $t \in \mathcal{L}$ is a sequence of activities $t = \langle a_1, a_2, \dots, a_n \rangle$, with $a_i \in \mathcal{A}, 1 \leq i \leq n$.*

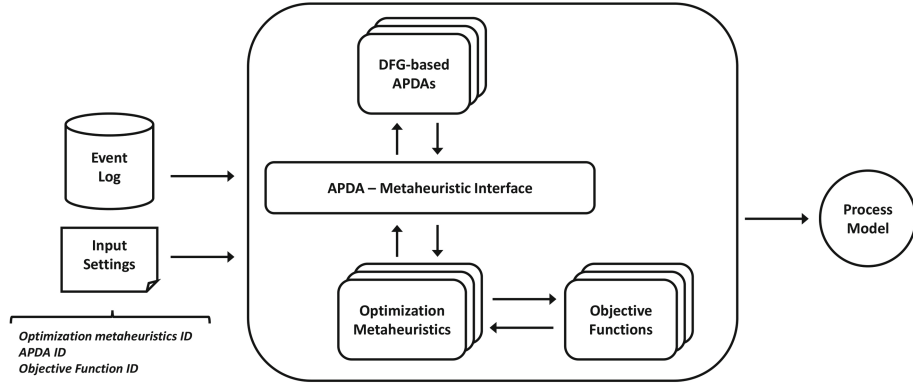


Fig. 1. Overview of our approach.

Definition 2 [Directly-Follows Graph (DFG)]. Given an event log \mathcal{L} , its Directly-Follows Graph (DFG) is a directed graph $\mathcal{G} = (N, E)$, where: N is the set of nodes, $N = \{a \in \mathcal{A} \mid \exists t \in \mathcal{L} \wedge a \in t\}$; and E is the set of edges $E = \{(x, y) \in N \times N \mid \exists t = \langle a_1, a_2, \dots, a_n \rangle, t \in \mathcal{L} \wedge a_i = x \wedge a_{i+1} = y \ [1 \leq i \leq n - 1]\}$.

Definition 3 [DFG-Matrix]. Given a DFG $\mathcal{G} = (N, E)$ and a function $\theta : N \rightarrow [1, |N|]$,¹ the DFG-Matrix is a squared matrix $X_{\mathcal{G}} \in [0, 1] \cap \mathbb{N}^{|N| \times |N|}$, where each cell $x_{i,j} = 1 \iff \exists (a_1, a_2) \in E \mid \theta(a_1) = i \wedge \theta(a_2) = j$, otherwise $x_{i,j} = 0$.

An APDA is said to be *DFG-based* if it first generates the DFG of the event log, then applies an algorithm to filter (e.g. removing activities) from the DFG, and finally converts the processed DFG into a process model.² Examples of DFG-based APDAs are Inductive Miner [18], Heuristics Miner [6, 25], Fodina [24], and Split Miner [8].

Different DFG-based APDAs may extract different DFGs from the same log. Also, a DFG-based APDA may discover different DFGs from the same log depending on its hyperparameter settings (e.g. the filtering threshold). The algorithm(s) used by a DFG-based APDA to discover the DFG from the event log and convert it into a process model may greatly affect the accuracy of an APDA. Accordingly, our approach focuses on optimizing the discovery of the DFG rather than its conversion into a process model.

3.2 Approach Overview

As shown in Fig. 1, our approach takes three inputs (in addition to the log): (i) the optimization metaheuristics; (ii) the objective function to be optimized (e.g.

¹ θ maps each node of the DFG to a natural number.

² Herein, when using the term DFG, we refer to the processed DFG (after filtering).

Algorithm 1. Optimization Approach

```

input   : Event Log  $\mathcal{L}$ , Metaheuristic  $\omega$ , Objective Function  $\mathcal{F}$ , DFG-based APDA  $\alpha$ 
1 CurrentDFG  $\mathcal{G}_c \leftarrow \text{DiscoverDFG}(\alpha, \mathcal{L})$ ;
2 BestModel  $\hat{m} \leftarrow \text{ConvertDFGtoProcessModel}(\alpha, \mathcal{G}_c)$ ;
3 CurrentScore  $s_c \leftarrow \text{AssessQuality}(\mathcal{F}, \hat{m})$ ;
4 BestScore  $\hat{s} \leftarrow s_c$ ;
5 while  $\text{CheckTerminationCriteria}() = \text{FALSE}$  do
6   Set  $V \leftarrow \text{GenerateNeighbours}(\mathcal{G}_c)$ ;
7   Map  $S \leftarrow \emptyset$ ;
8   Map  $M \leftarrow \emptyset$ ;
9   for  $\mathcal{G} \in V$  do
10    ProcessModel  $m \leftarrow \text{ConvertDFGtoProcessModel}(\alpha, \mathcal{G})$ ;
11    Score  $s \leftarrow \text{AssessQuality}(\mathcal{F}, m)$ ;
12    add  $(\mathcal{G}, s)$  to  $S$ ;
13    add  $(\mathcal{G}, m)$  to  $M$ ;
14    $\mathcal{G}_c \leftarrow \text{UpdateDFG}(\omega, S, \mathcal{G}_c, s_c)$ ;
15    $s_c \leftarrow \text{GetMapElement}(S, \mathcal{G}_c)$ ;
16   if  $\hat{s} < s_c$  then
17      $\hat{s} \leftarrow s_c$ ;
18      $\hat{m} \leftarrow \text{GetMapElement}(M, \mathcal{G}_c)$ ;
19 return  $\hat{m}$ ;
```

F-score); (iii) and the DFG-based APDA to be used for discovering a process model.

Algorithm 1 describes how our approach operates. First, the input event log is given to the APDA, which returns the discovered DFG and its corresponding process model (lines 1 and 2). This DFG becomes the current DFG and process model becomes the best process model (so far). The model's objective function score (e.g. F-score) is stored as the current score and the best score (lines 3 and 4). The current DFG is then given as input to function *GenerateNeighbours*, which applies changes to the current DFG to generate a set of neighbouring DFGs (line 6). These latter are given as input to the APDA, which returns the corresponding into process models. The process models are assessed by the objective function evaluators (line 9 to 13). When the metaheuristic receives the results from the evaluators (along with the current DFG and score), it chooses the new current DFG and updates the current score (lines 14 and 15). If the new current score is higher than the best score (line 16), it updates the best process model and the best score (lines 17 and 18). After the update, a new iteration starts, unless a termination criterion is met (e.g. a timeout, a maximum number of iterations, or a minimum threshold for the objective function). In this latter case, it outputs the best model found, i.e. the process model scoring the highest value for the objective function.

3.3 Adaptation of the Optimization Metaheuristics

To adapt Iterative Local Search (ILS), Tabu Search (TABU), and Simulated Annealing (SIMA) to the problem of automated process discovery, we need to define the following three concepts: (i) the problem solution space; (ii) a solution neighbourhood; (iii) the objective function. These design choices determine

how each of the metaheuristics navigates the solution space and escapes local minima, i.e. how to design the Algorithm 1 functions: *GenerateNeighbours* and *UpdateDFG*, resp. lines 6 and 14.

Solution Space. Being our goal the optimization of APDAs, we are forced to choose a solution space that fits well our context regardless the selected APDA. If we assume that the APDA is DFG-based (that is the case for the majority of the available APDAs), we can define the solution space as the set of all the DFG discoverable from the event log. Indeed, any DFG-based APDA can generate deterministically a process model from a DFG.

Solution Neighbourhood. Having defined the solution space as the set of all the DFG discoverable from the event log, we can refer to any element of this solution space as a DFG-Matrix. Given a DFG-Matrix, we define its neighbourhood as the set of all the matrices having one different cell value (i.e. DFGs having one more/less edge). In the following, every time we refer to DFG we assume it is represented as a DFG-Matrix.

Objective Function. It is possible to define the objective function as any function assessing one of the four quality dimensions for discovered process models (introduced in Sect. 2). However, being interested in optimizing the APDAs to discover the most accurate process model, in the remaining of this paper, we refer to the objective function as the F-score of fitness and precision: $\frac{2 \cdot \text{fit} \cdot \text{prec}}{\text{fit} + \text{prec}}$. Nonetheless, we remark that our approach can operate also with objective functions that take into account multiple quality dimensions striving for a trade-off, e.g. F-score and model complexity.

Having defined the solution space, a solution neighbourhood, and the objective function, we can turn our attention on how ILS, TABU, and SIMA navigate the solution space. ILS, TABU, and SIMA share similar traits in solving an optimization problem, especially when it comes to the navigation of the solution space. Given a problem and its solution space, any of these three S-metaheuristics starts from a (random) solution, discovers one or more neighbouring solutions, and assesses them with the objective function to find a solution better than the current. If a better solution is found, it is chosen as the new current solution and the metaheuristic performs a new neighbourhood exploration. If a better solution is not found, e.g. the current solution is locally optimal, the three metaheuristics follow different approaches to escape the local optimum and continue the solution space exploration. Algorithm 1 orchestrates and facilitates the parts of this procedure shared by the three metaheuristics. However, we must define the functions *GenerateNeighbours* (GNF) and *UpdateDFG* (UDF).

The GNF receives in input a solution of the solution space, i.e. a DFG, and it generates a set of neighbouring DFGs. By definition, GNF is independent from the metaheuristic and it can be as simple or as elaborate as we demand. An example of a simple GNF is a function that randomly selects neighbouring DFGs turning one cell of the input DFG-Matrix to 0 or to 1. Whilst, an example of an elaborate GNF is a function that accurately selects neighbouring DFGs

relying on the feedback received from the objective function assessing the input DFG, as we show in Sect. 3.4.

The UDF is at the core of our optimization, and it represents the metaheuristic itself. It receives in input the neighbouring DFGs, the current DFG, and the current score, and it selects among the neighbouring DFGs the one that should become the new current DFG. At this point, we can differentiate two cases: (i) among the input neighbouring DFGs there is at least one having a higher objective function score than the current; (ii) none of the input neighbouring DFGs has a higher objective function score than the current. In the first case, UDF always outputs the DFG having the highest score (regardless the selected metaheuristic). In the second case, the current DFG may be a local optimum, and each metaheuristic escapes it with a different strategy.

Iterative Local Search applies the simplest strategy, it perturbs the current DFG. The perturbation is meant to alter the DFG in such a way to escape the local optimum, e.g. randomly adding and removing multiple edges from the current DFG. The perturbed DFG is the output of the UDF.

Tabu Search relies on its three memories to escape a local optimum. The short-term memory (a.k.a. Tabu-list), containing DFG that must not be explored further. The intermediate-term memory, containing DFGs that should lead to better results and, therefore, should be explored in the near future. The long-term memory, containing DFGs (with characteristics) that have been seen multiple times and, therefore, not to explore in the near future. TABU updates the three memories each time the UDF is executed. Given the set of neighbouring DFGs and their respective objective function scores (see Algorithm 1, map S), TABU adds each DFG to a different memory. DFGs worsening the objective function score are added to the Tabu-list. DFGs improving the objective function score, yet less than another neighbouring DFG, are added to the intermediate-term memory. DFGs that do not improve the objective function score are added to the long-term memory. Also, the current DFG is added to the Tabu-list, being it already explored. When TABU does not find a better DFG in the neighbourhood of the current DFG, it returns the latest DFG added to the intermediate-term memory. If the intermediate-term memory is empty, TABU returns the latest DFG added to the long-term memory. If both these memories are empty, TABU requires a new (random) DFG from the APDA, and outputs its DFG.

Simulated Annealing avoids getting stuck in a local optimum by allowing the selection of DFGs worsening the objective function score. In doing so, SIMA explores areas of the solution space that other S-metaheuristics do not. When a better DFG is not found in the neighbourhood of the current DFG, SIMA analyses one neighbouring DFG at a time. If this latter does not worsen the objective function score, SIMA outputs it. Instead, if the neighbouring DFG worsens the objective function score, SIMA outputs it with a probability of $e^{-\frac{|s_n - s_c|}{T}}$, where s_n and s_c are the objective function scores of (respectively) the neighbouring DFG and the current DFG, and the temperature T is an integer that converges to zero as a linear function of the maximum number of iterations.

The temperature is fundamental to avoid updating the current DFG with a worse one if there would be no time to recover from the worsening (i.e. too few iterations left for continuing the exploration of the solution space from the worse DFG).

3.4 Instantiation for Split Miner

To assess our approach, we define an instantiation of it for Split Miner – a DFG-based APDA that performs favourably relative to other state-of-the-art APDAs [7]. To instantiate our approach for a concrete APDA, we need to implement an interface that allows the metaheuristics to interact with the APDA (as discussed above). The interface should provide four functions: *DiscoverDFG* and *ConvertDFGtoProcessModel* (see Algorithm 1), the *Restart Function* (RF) for TABU, and the *Perturbation Function* (PF) for ILS. The first two functions come with the DFG-based APDA, in our case Split Miner. Note that, the output of *DiscoverDFG* of Split Miner varies according to the hyperparameters settings.³ To discover the initial DFG (Algorithm 1, line 1), Split Miner uses its default parameters. We removed the randomness for discovering the initial DFG because most of the times, the DFG discovered by Split Miner with default parameters is already a good solution [8], and starting the solution space exploration from this latter can reduce the total exploration time.

Function RF is very similar to *DiscoverDFG*, since it requires the APDA to output a DFG, the only difference is that RF must output a different DFG every time it is executed. We adapted the *DiscoverDFG* of Split Miner to output the DFG discovered with default parameters the first time it is executed, and for the following executions a DFG discovered with random parameters.

Finally, function PF can be provided either by the APDA (via the interface) or by the metaheuristic. However, PF can be more effective when not generalised by the metaheuristic, allowing the APDA to apply different perturbations to the DFGs, taking into account how the APDA converts the DFG to a process model.

We invoke Split Miner’s concurrency oracle to extract the possible parallelism relations in the log using a randomly chosen parallelism threshold. For each new parallel relation discovered (not present in the current solution), two edges are removed from the DFG, whilst, for each deprecated parallel relation, two edges are added to the DFG. Alternatively, it is possible to set $PF = RF$, so that instead of perturbing the current DFG, a new random DFG is generated. This variant of the ILS is called Repetitive Local Search (RLS). In the evaluation reported below, we use both ILS and its variant RLS.

We use the F-score as the objective function, which is computed from the fitness and precision. Among the existing measures of fitness and precision we selected the Markovian fitness and precision defined in [5] (boolean function variant, order $k = 5$). The rationale for this choice is that these measures of fitness

³ Split Miner has two hyperparameters: the noise filtering threshold, used to drop infrequent edges in the DFG, and the parallelism threshold, used to determine which potential parallel relations between activities are used when discovering the process model from the DFG.

Algorithm 2. Generate Neighbours Function (GNF)

```

input   : CurrentDFG  $\mathcal{G}_c$ , CurrentMarkovianScore  $s_c$ , Integer  $size_n$ 
1 if  $getFitnessScore(s_c) > getPrecisionScore(s_c)$  then
2   | Set  $E_m \leftarrow getEdgesForImprovingPrecision(s_c)$ ;
3 else
4   | Set  $E_m \leftarrow getEdgesForImprovingFitness(s_c)$ ;
5 Set  $N \leftarrow \emptyset$ ;
6 while  $E_m \neq \emptyset \wedge |N| \neq size_n$  do
7   Edge  $e \leftarrow getRandomElement(E_m)$ ;
8   NeighbouringDFG  $\mathcal{G}_n \leftarrow copyDFG(\mathcal{G}_c)$ ;
9   if  $getFitnessScore(s_c) > getPrecisionScore(s_c)$  then
10  | if  $canRemoveEdge(\mathcal{G}_n, e)$  then add  $\mathcal{G}_n$  to  $N$ ;
11  | else
12  |   addEdge( $\mathcal{G}_n, e$ );
13  |   add  $\mathcal{G}_n$  to  $N$ ;
14 return  $N$ ;

```

and precision are the fastest to compute among state-of-the-art measures [4, 5]. Furthermore, the Markovian fitness (precision) provides a feedback that tells us what edges could be added to (removed from) the DFG to improve the fitness (precision). This feedback allows us to design an effective GNF. In the instantiation of our approach for Split Miner, the objective function's output is a data structure composed of: the Markovian fitness and precision of the model, the F-score, and the mismatches between the model and the event log identified during the computation of the Markovian fitness and precision, i.e. the sets of the edges that could be added (removed) to improve the fitness (precision).

Given this objective function's output, our GNF is described in Algorithm 2. The function receives as input the current DFG (\mathcal{G}_c), its objective function score (the data structure s_c), and the number of neighbours to generate ($size_n$). If fitness is greater than precision, we retrieve (from s_c) the set of edges (E_m) that could be removed from \mathcal{G}_c to improve its precision (line 2). Conversely, if precision is greater than fitness, we retrieve (from s_c) the set of edges (E_m) that could be added to \mathcal{G}_c to improve its fitness (line 4). The reasoning behind this design choice is that, given that our objective function is the F-score, it is preferable to increase the lower of the two measures (precision or fitness). i.e. if the fitness is lower, we increase fitness, and conversely if the precision is lower. Once we have E_m , we select randomly one edge from it, we generate a copy of the current DFG (\mathcal{G}_n), and we either remove or add the randomly selected edge according to the accuracy measure we want to improve (precision or fitness), see lines 7 to 13. If the removal of an edge generates a disconnected \mathcal{G}_n , we do not add this latter to the neighbours set (N), line 10. We keep iterating over E_m until the set is empty (i.e. no mismatching edges are left) or N reaches its max size (i.e. $size_n$). We then return N .

The algorithm ends when the maximum execution time is reached or and the maximum number of iterations it reached (in the experiments below, we set them by default to 5 min and 50 iterations).

4 Evaluation

We implemented our approach as a Java command-line application⁴ using Split Miner as the underlying automated process discovery approach and Markovian accuracy F-score as the objective function (cf. Sect. 3.4). We compared the quality of the models discovered by applying each of the optimization metaheuristics mentioned against those discovered by four baselines: (i) Split Miner; (ii) Split Miner with hyper-parameter optimization; (iii) Evolutionary Tree Miner; and (iv) Inductive Miner.

The experiments were performed on an Intel Core i5-6200U@2.30 GHz with 16 GB RAM running Windows 10 Pro (64-bit) and JVM 8 with 14 GB RAM (10 GB Stack and 4 GB Heap). The approach’s implementation, the batch tests, the results, and all the models discovered during the experiments are available for reproducibility purposes at <https://doi.org/10.6084/m9.figshare.7824671.v1>.

4.1 Dataset

For our evaluation we used the dataset of the benchmark of automated process discovery approaches in [7], which to the best of our knowledge is the most recent benchmark on this topic. This dataset includes twelve public logs and eight private logs. The public logs originate from the 4TU Centre for Research Data, and include the *BPI Challenge* (BPIC) logs (2012-17),⁵ the *Road Traffic Fines Management Process* (RTFMP) log⁶ and the *SEPSIS* log⁷. These logs record executions of business processes from a variety of domains, e.g. health-care, finance, government and IT service management. In seven logs (BPIC14, the BPIC15 collection and BPIC17), the filtering technique in [13] was applied to remove infrequent behavior; this step was necessary to maintain consistency with the benchmark dataset. The eight proprietary logs are sourced from several companies in the education, insurance, IT service management and IP management domains.

Table 1 reports the characteristics of the logs. As seen in the table, the dataset comprises simple logs (e.g. BPIC13_{cp}) and very complex ones (e.g. SEPSIS, PRT2) in terms of percentage of distinct traces, and both small logs (e.g. BPIC13_{cp} and SEPSIS) and large ones (e.g. BPIC17 and PRT9) in terms of total number of events.

⁴ Available under the label “Metaheuristically Optimized Split Miner” at <http://apromore.org/platform/tools>.

⁵ <https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f>,
<https://doi.org/10.4121/uuid:a7ce5c55-03a7-4583-b855-98b86e1a2b07>,
<https://doi.org/10.4121/uuid:c3e5d162-0cfd-4bb0-bd82-af5268819c35>,
<https://doi.org/10.4121/uuid:31a308ef-c844-48da-948c-305d167a0ec1>,
<https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b>.

⁶ <https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5>.

⁷ <https://doi.org/10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460>.

Table 1. Descriptive statistics of the real-life logs (public and proprietary).

Log	BPIC12	BPIC13 _{cp}	BPIC13 _{inc}	BPIC14 _f	BPIC15 _{1f}	BPIC15 _{2f}	BPIC15 _{3f}	BPIC15 _{4f}	BPIC15 _{5f}
Total Traces	13,087	1,487	7,554	41,353	902	681	1,369	860	975
Dist. Traces(%)	33.4	12.3	20	36.1	32.7	61.7	60.3	52.4	45.7
Total Events	262,200	6,660	65,533	369,485	21,656	24,678	43,786	29,403	30,030
Dist. Events	36	7	13	9	70	82	62	65	74
Tr. length	(min)	3	1	1	3	5	4	5	4
	(avg)	20	4	9	9	24	36	32	34
	(max)	175	35	123	167	50	63	54	61

Log	BPIC17 _f	RTFMP	SEPSIS	PRT1	PRT2	PRT3	PRT4	PRT6	PRT7	PRT9	PRT10
Total Traces	21,861	150,370	1,050	12,720	1,182	1,600	20,000	744	2,000	787,657	43,514
Dist. Traces(%)	40.1	0.2	80.6	8.1	97.5	19.9	29.7	22.4	6.4	0.01	0.01
Total Events	714,198	561,470	15,214	75,353	46,282	13,720	166,282	6,011	16,353	1,808,706	78,864
Dist. Events	41	11	16	9	9	15	11	9	13	8	19
Tr. length	(min)	11	2	3	2	12	6	6	7	8	1
	(avg)	33	4	14	5	39	8	8	8	2	1
	(max)	113	2	185	64	276	9	36	21	11	58

4.2 Experimental Setup

For each log in our dataset, we discovered eight process models: four using the metaheuristics presented in Sect. 3 (RLS, ILS, TABU and SIMA) and four baselines. The baselines include the Evolutionary Tree Miner (ETM) [10], Inductive Miner infrequent variant (IM) [18], and Split Miner (SM) [8], all with default parameters settings. ETM was allowed to run with a 4-h timeout. All comparisons with ETM are meant as comparisons of accuracy (fitness, precision, F-score) and not as execution time comparisons, as the computational heaviness of ETM has already been shown in previous work [7, 10]. The fourth baseline (HPO_{sm}) is a hyperparameter-optimized version of the SM algorithm, where we varied the two hyperparameters of SM (noise filtering and parallelism filtering threshold) across their full range with steps of 0.01 (from 0.01 to 1.00), and retaining the model with the highest Markovian F-score.

ETM, IM and SM were selected as baselines because they had the highest accuracy in a recent benchmark comparison of APDAs [7]. We also selected HPO_{sm} to compare the effects of optimization metaheuristics versus hyperparameter optimization.

For each of the discovered models we measured accuracy, complexity and discovery time. For the accuracy, we adopted two different sets of measures: one based on *alignments*, computing fitness and precision with the approaches proposed in [1, 2] (alignment-based accuracy); and one based on *Markovian abstractions*, computing fitness and precision with the approaches proposed in [4, 5] (Markovian accuracy). For assessing the complexity of the models we relied on *size* (number of nodes of the model), Control-Flow Complexity (CFC) (the amount of branching caused by split gateways in the model), and Structuredness (the percentage of nodes located directly inside a well-structured single-entry single-exit fragment).

4.3 Results

Tables 2 and 3 show the results of our comparative evaluation. Each row reports the quality of each discovered process model in terms of accuracy (both alignment-based and Markovian) and complexity, as well as the discovery time.

Due to space, we held out from the tables four logs: BPIC13_{cp}, BPIC13_{inc}, BPIC17, and PRT9. For these logs, none of the metaheuristics could improve the accuracy of the model already discovered by SM. This is due to the high fitness score achieved by SM in these logs. By design, our metaheuristics try to improve the precision by removing edges, but in these four cases, no edge could be removed without compromising the structure of the model (i.e. the model would become disconnected).

For the remaining 16 logs, all the metaheuristics improved consistently the Markovian F-score w.r.t. SM. Also, all the metaheuristics performed better than HPO_{sm}, except in two cases (BPIC12 and PRT1). Overall, the most effective optimization metaheuristic was ILS, which delivered the highest Markovian F-score nine times out of 16, followed by SIMA (eight times), RLS and TABU (six times each). Compared to ETM, the four metaheuristics achieved better Markovian F-scores in 15 out of 16 cases, and better alignment F-scores 14 times out of 16, while compared to IM, all the optimization metaheuristics achieved better Markovian F-scores in 15 cases out of 16, and better alignment F-scores across the whole dataset.

Despite the fact that the objective function of the metaheuristics was the Markovian F-score, all four metaheuristics optimized in half of the cases the alignment-based F-score. This is due to the fact that any improvement on the Markovian fitness translates into an improvement on the alignment-based fitness, though the same does not hold for the precision. This result highlights the (partial) correlation between the alignment-based and the Markovian accuracies, already reported in previous studies [4, 5]. Analysing the complexity of the models, we note that most of the times (nine cases out of 16) the F-score improvement achieved by the metaheuristics comes at the cost of size and CFC. This is expected, since SM tends to discover models with higher precision than fitness [7]. What happens is that to improve the F-score, new behavior is added to the model in the form of new edges (note that new nodes are never added). Adding new edges leads to new gateways and consequently to increasing size and CFC. On the other hand, when the precision is lower than fitness and the metaheuristic aims to increase the value of this measure to improve the overall F-score, the result is the opposite: the model complexity reduces as edges are removed. This is the case of the RTFMP and PRT10 logs. As an example of the two possible scenarios, Fig. 2 shows the models discovered by SIMA and SM from the BPIC14_f log (where the model discovered by SIMA is more complex than that obtained with SM), while Fig. 3 shows the models discovered by SIMA and SM from the RTFMP log (where the model discovered by SIMA is simpler). Comparing the results obtained by the metaheuristics with HPO_{sm}, we can see that our approach allows us to discover models that cannot be discovered simply by tuning the hyperparameters of SM. This relates to the solution space

Table 2. Comparative evaluation results for the public logs.

Event Log	Discovery Approach	Align. Acc.			Markov. Acc. ($k = 5$)			Complexity			Exec. Time(s)
		Fitness	Prec.	F-score	Fitness	Prec.	F-score	Size	CFC	Struct.	
BPIC12	ETM	0.440	0.820	0.573	0.536	0.462	0.496	67	16	1.00	14,400
	IM	0.990	0.502	0.666	0.280	0.002	0.005	59	37	1.00	6.6
	SM	0.963	0.520	0.675	0.818	0.139	0.238	51	41	0.69	3.2
	HPO _{sm}	0.781	0.796	0.788	0.575	0.277	0.374	40	17	0.58	4295.8
	RLS _{sm}	0.921	0.671	0.776	0.586	0.247	0.348	49	31	0.90	159.3
	ILS _{sm}	0.921	0.671	0.776	0.586	0.247	0.348	49	31	0.90	159.4
	TABU _{sm}	0.921	0.671	0.776	0.586	0.247	0.348	49	31	0.90	140.7
	SIMA _{sm}	0.921	0.671	0.776	0.586	0.247	0.348	49	31	0.90	151.1
BPIC14 _f	ETM	0.610	1.000	0.758	0.009	0.313	0.017	23	9	1.00	14,400
	IM	0.890	0.646	0.749	0.501	0.346	0.409	31	18	1.00	3.4
	SM	0.772	0.881	0.823	0.150	1.000	0.262	20	14	1.00	0.8
	HPO _{sm}	0.852	0.857	0.855	0.449	1.000	0.619	22	16	0.59	575.8
	RLS _{sm}	1.000	0.771	0.871	1.000	0.985	0.992	28	34	0.54	139.0
	ILS _{sm}	1.000	0.771	0.871	1.000	0.985	0.992	28	34	0.54	151.3
	TABU _{sm}	0.955	0.775	0.855	0.856	0.999	0.922	26	31	0.69	154.7
	SIMA _{sm}	1.000	0.771	0.871	1.000	0.985	0.992	28	34	0.54	140.3
BPIC15 _{1f}	ETM	0.560	0.940	0.702	0.235	0.284	0.257	67	19	1.00	14,400
	IM	0.970	0.566	0.715	0.665	0.001	0.002	164	108	1.00	1.1
	SM	0.899	0.871	0.885	0.701	0.726	0.713	111	45	0.51	0.7
	HPO _{sm}	0.962	0.833	0.893	0.804	0.670	0.731	117	55	0.45	1242.3
	RLS _{sm}	0.925	0.839	0.880	0.774	0.803	0.788	124	63	0.39	163.6
	ILS _{sm}	0.925	0.839	0.880	0.774	0.803	0.788	124	63	0.39	166.8
	TABU _{sm}	0.948	0.843	0.892	0.774	0.805	0.789	125	64	0.33	187.2
	SIMA _{sm}	0.920	0.839	0.878	0.772	0.807	0.789	125	63	0.43	160.4
BPIC15 _{2f}	ETM	0.620	0.910	0.738	0.301	0.389	0.339	95	32	1.00	14,400
	IM	0.948	0.556	0.701	0.523	0.002	0.004	193	123	1.00	1.7
	SM	0.783	0.877	0.828	0.514	0.596	0.552	129	49	0.36	0.6
	HPO _{sm}	0.808	0.851	0.829	0.561	0.582	0.572	133	56	0.30	1398.9
	RLS _{sm}	0.870	0.797	0.832	0.667	0.670	0.668	156	86	0.20	158.3
	ILS _{sm}	0.869	0.795	0.830	0.663	0.680	0.671	157	86	0.20	157.6
	TABU _{sm}	0.870	0.794	0.830	0.665	0.667	0.666	150	83	0.23	176.8
	SIMA _{sm}	0.871	0.775	0.820	0.677	0.662	0.669	159	93	0.26	167.4
BPIC15 _{3f}	ETM	0.680	0.880	0.767	0.238	0.172	0.199	84	29	1.00	14,400
	IM	0.950	0.554	0.700	0.480	0.002	0.003	159	108	1.00	1.3
	SM	0.774	0.925	0.843	0.436	0.764	0.555	96	35	0.49	0.5
	HPO _{sm}	0.783	0.910	0.842	0.477	0.691	0.564	99	39	0.56	9230.4
	RLS _{sm}	0.812	0.903	0.855	0.504	0.775	0.611	110	53	0.35	151.5
	ILS _{sm}	0.833	0.868	0.850	0.533	0.775	0.631	120	66	0.23	153.8
	TABU _{sm}	0.832	0.852	0.842	0.558	0.690	0.617	121	64	0.23	173.4
	SIMA _{sm}	0.827	0.839	0.833	0.565	0.694	0.623	123	71	0.18	159.4
BPIC15 _{4f}	ETM	0.650	0.930	0.765	0.351	0.292	0.319	83	28	1.00	14,400
	IM	0.955	0.585	0.726	0.567	0.001	0.002	162	111	1.00	2.4
	SM	0.762	0.886	0.820	0.516	0.615	0.562	101	37	0.27	0.5
	HPO _{sm}	0.785	0.860	0.821	0.558	0.578	0.568	103	40	0.27	736.4
	RLS _{sm}	0.825	0.854	0.839	0.634	0.672	0.652	114	57	0.21	146.9
	ILS _{sm}	0.853	0.807	0.829	0.649	0.657	0.653	117	64	0.27	147.8
	TABU _{sm}	0.811	0.794	0.803	0.642	0.661	0.651	115	61	0.24	161.7
	SIMA _{sm}	0.847	0.812	0.829	0.624	0.649	0.636	117	61	0.18	148.2
BPIC15 _{5f}	ETM	0.570	0.940	0.710	0.365	0.504	0.423	88	18	1.00	14,400
	IM	0.937	0.179	0.301	0.242	0.000	0.000	134	95	1.00	2.5
	SM	0.806	0.915	0.857	0.555	0.598	0.576	110	38	0.34	0.6
	HPO _{sm}	0.789	0.941	0.858	0.529	0.655	0.585	102	30	0.33	972.3
	RLS _{sm}	0.868	0.813	0.840	0.737	0.731	0.734	137	78	0.14	159.3
	ILS _{sm}	0.868	0.813	0.840	0.737	0.731	0.734	137	78	0.14	153.8
	TABU _{sm}	0.885	0.818	0.850	0.739	0.746	0.743	137	79	0.14	173.3
	SIMA _{sm}	0.867	0.811	0.838	0.734	0.727	0.731	137	78	0.16	154.3
RTFMP	ETM	0.990	0.920	0.954	0.981	0.010	0.019	57	32	1.00	14,400
	IM	0.980	0.700	0.817	0.934	0.046	0.087	34	20	1.00	13.9
	SM	0.996	0.958	0.977	0.959	0.311	0.470	22	17	0.46	2.9
	HPO _{sm}	0.887	1.000	0.940	0.685	0.696	0.690	20	9	0.35	2452.7
	RLS _{sm}	0.988	1.000	0.994	0.899	0.794	0.843	22	14	0.46	142.8
	ILS _{sm}	0.988	1.000	0.994	0.899	0.794	0.843	22	14	0.46	143.8
	TABU _{sm}	0.988	1.000	0.994	0.899	0.794	0.843	22	14	0.46	114.8
	SIMA _{sm}	0.986	1.000	0.993	0.875	0.893	0.884	23	15	0.39	131.0
SEPSIS	ETM	0.830	0.660	0.735	0.696	0.096	0.169	108	101	1.00	14,400
	IM	0.991	0.445	0.614	0.741	0.012	0.024	50	32	1.00	1.3
	SM	0.764	0.706	0.734	0.349	0.484	0.406	32	23	0.94	0.4
	HPO _{sm}	0.925	0.588	0.719	0.755	0.293	0.423	33	34	0.39	28,846
	RLS _{sm}	0.839	0.630	0.720	0.508	0.430	0.466	35	29	0.77	145.4
	ILS _{sm}	0.812	0.625	0.706	0.455	0.436	0.445	35	28	0.86	157.1
	TABU _{sm}	0.839	0.630	0.720	0.508	0.430	0.466	35	29	0.77	137.0
	SIMA _{sm}	0.806	0.613	0.696	0.477	0.445	0.460	35	30	0.77	137.2

Table 3. Comparative evaluation results for the proprietary logs.

Event Log	Discovery Method	Align. Acc.			Markov. Acc. ($k = 5$)			Complexity			Exec. Time(s)
		Fitness	Prec.	F-score	Fitness	Prec.	F-score	Size	CFC	Struct.	
PRT1	ETM	0.990	0.811	0.892	0.977	0.213	0.350	23	12	1.00	14,400
	IM	0.902	0.673	0.771	0.232	0.051	0.084	20	9	1.00	3.8
	SM	0.976	0.974	0.975	0.730	0.669	0.698	20	14	1.00	0.4
	HPO _{sm}	0.999	0.948	0.972	0.989	0.620	0.762	19	14	0.53	298.3
	RLS _{sm}	0.976	0.974	0.975	0.730	0.669	0.698	20	14	1.00	155.3
	ILS _{sm}	0.976	0.974	0.975	0.730	0.669	0.698	20	14	1.00	153.2
	TABU _{sm}	0.976	0.974	0.975	0.730	0.669	0.698	20	14	1.00	10.3
	SIMA _{sm}	0.983	0.964	0.974	0.814	0.722	0.765	20	15	1.00	132.6
PRT2	ETM	0.572	0.943	0.712	0.105	0.788	0.186	86	21	1.00	14,400
	IM	ex	ex	ex	0.329	0.179	0.232	45	33	1.00	2.3
	SM	0.795	0.581	0.671	0.457	0.913	0.609	29	23	1.00	0.3
	HPO _{sm}	0.826	0.675	0.743	0.501	0.830	0.625	21	13	0.67	406.4
	RLS _{sm}	0.886	0.421	0.571	0.629	0.751	0.685	29	34	1.00	141.4
	ILS _{sm}	0.890	0.405	0.557	0.645	0.736	0.688	29	35	1.00	172.3
	TABU _{sm}	0.866	0.425	0.570	0.600	0.782	0.679	29	33	1.00	143.1
	SIMA _{sm}	0.886	0.424	0.574	0.629	0.751	0.685	29	34	1.00	139.7
PRT3	ETM	0.979	0.858	0.915	0.858	0.313	0.459	51	37	1.00	14,400
	IM	0.975	0.680	0.801	0.874	0.481	0.621	37	20	1.00	0.9
	SM	0.882	0.887	0.885	0.381	0.189	0.252	31	23	0.58	0.4
	HPO _{sm}	0.890	0.899	0.895	0.461	0.518	0.488	26	14	0.81	290.2
	RLS _{sm}	0.945	0.902	0.923	0.591	0.517	0.551	31	23	0.55	138.4
	ILS _{sm}	0.945	0.902	0.923	0.591	0.517	0.551	31	23	0.55	144.2
	TABU _{sm}	0.944	0.902	0.922	0.589	0.519	0.552	30	20	0.60	134.7
	SIMA _{sm}	0.945	0.902	0.923	0.591	0.517	0.551	31	23	0.55	133.7
PRT4	ETM	0.844	0.851	0.847	0.629	0.950	0.757	64	28	1.00	14,400
	IM	0.927	0.753	0.831	0.615	0.952	0.747	27	13	1.00	1.1
	SM	0.884	1.000	0.938	0.483	1.000	0.652	25	15	0.96	0.5
	HPO _{sm}	0.973	0.930	0.951	0.929	0.989	0.958	26	24	0.31	867.5
	RLS _{sm}	0.997	0.903	0.948	0.993	0.990	0.992	26	28	0.92	140.1
	ILS _{sm}	0.997	0.903	0.948	0.993	0.990	0.992	26	28	0.92	152.3
	TABU _{sm}	0.955	0.914	0.934	0.883	0.988	0.932	26	26	0.77	138.6
	SIMA _{sm}	0.997	0.903	0.948	0.993	0.990	0.992	26	28	0.92	136.9
PRT6	ETM	0.980	0.796	0.878	0.890	0.611	0.725	41	16	1.00	14,400
	IM	0.989	0.822	0.898	0.946	0.444	0.604	23	10	1.00	2.9
	SM	0.937	1.000	0.967	0.542	1.000	0.703	15	4	1.00	0.3
	HPO _{sm}	0.937	1.000	0.967	0.542	1.000	0.703	15	4	1.00	105.1
	RLS _{sm}	0.984	0.928	0.955	0.840	0.818	0.829	22	14	0.41	141.1
	ILS _{sm}	0.984	0.928	0.955	0.840	0.818	0.829	22	14	0.41	144.2
	TABU _{sm}	0.984	0.928	0.955	0.840	0.818	0.829	22	14	0.41	124.9
	SIMA _{sm}	0.984	0.928	0.955	0.840	0.818	0.829	22	14	0.41	131.2
PRT7	ETM	0.900	0.810	0.853	0.969	0.217	0.355	60	29	1.00	14,400
	IM	1.000	0.726	0.841	1.000	0.543	0.704	29	13	1.00	1.3
	SM	0.914	0.999	0.954	0.650	1.000	0.788	29	10	0.48	0.6
	HPO _{sm}	0.944	1.000	0.971	0.772	1.000	0.871	22	9	0.64	173.1
	RLS _{sm}	0.993	1.000	0.996	0.933	1.000	0.965	23	11	0.78	139.2
	ILS _{sm}	0.993	1.000	0.996	0.933	1.000	0.965	23	11	0.78	142.9
	TABU _{sm}	0.993	1.000	0.996	0.933	1.000	0.965	23	11	0.78	134.0
	SIMA _{sm}	0.993	1.000	0.996	0.933	1.000	0.965	23	11	0.78	131.9
PRT10	ETM	1.000	0.627	0.771	0.748	0.001	0.003	61	45	1.00	14,400
	IM	0.964	0.790	0.868	0.945	0.001	0.001	41	29	1.00	4.6
	SM	0.970	0.943	0.956	0.905	0.206	0.335	45	47	0.84	0.5
	HPO _{sm}	0.936	0.943	0.939	0.810	0.243	0.374	30	22	0.73	1214.3
	RLS _{sm}	0.917	0.989	0.952	0.741	0.305	0.432	44	41	0.86	153.0
	ILS _{sm}	0.917	0.989	0.952	0.741	0.305	0.432	44	41	0.86	155.4
	TABU _{sm}	0.917	0.989	0.952	0.741	0.305	0.432	44	41	0.86	117.6
	SIMA _{sm}	0.917	0.989	0.952	0.741	0.305	0.432	44	41	0.86	136.7

exploration. Indeed, while HPO_{sm} can only explore a limited number of solutions (DFGs), i.e. those that can be generated by the underlying APDA, SM in this case, by varying its hyperparameters, the metaheuristics go beyond the solution space of HPO_{sm} by exploring new DFGs in a pseudo-random manner.

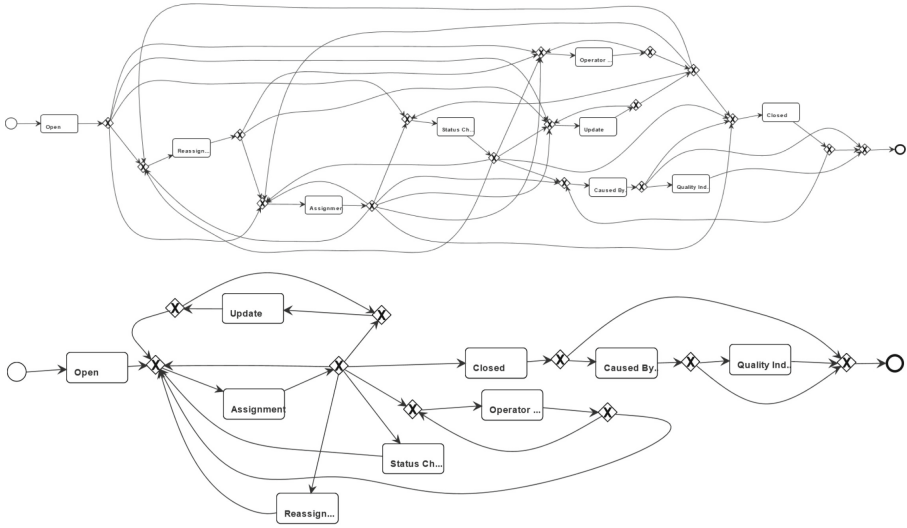


Fig. 2. BPIC14_f models discovered with SIMA (above) and SM (below).

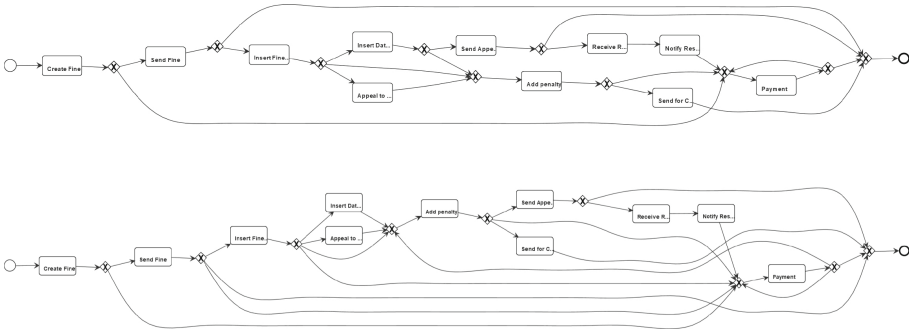


Fig. 3. BPIC14_f models discovered with SIMA (above) and SM (below).

In terms of execution times, the four metaheuristics perform similarly, having an average discovery time close to 150 s. While this is considerably higher than the execution time of SM (~1s on average), it is much lower than HPO_{sm} and ETM, while consistently achieving higher accuracy.

5 Conclusion

This paper showed that the use of S-metaheuristics is a promising approach to enhance the accuracy of DFG-based APDAs. The outlined approach takes advantage of the DFG’s simplicity to define efficient perturbation functions that improve fitness or precision while preserving structural properties required to ensure model correctness.

The evaluation showed that the metaheuristic extensions of Split Miner achieve considerably higher accuracy for a clear majority of logs in the benchmark, particularly when using fine-grained measures of fitness and precision based on Markovian abstractions, but also when using measures based on trace alignment. These accuracy gains come at the expense of slightly higher model size and structural complexity. The results also show that the choice of S-metaheuristics (among the four considered in this paper) does not visibly affect accuracy nor model complexity. The metaheuristic extensions do come with a penalty in terms of execution times. The execution times of the metaheuristic-enhanced versions of Split Miner are $\sim 2\text{--}3$ min versus ~ 1 s for the base miner. Interestingly, the S-metaheuristics improve accuracy even with respect a hyperparameter-optimized version of Split Miner, while achieving considerably lower execution times. This means that the metaheuristic extensions of Split Miner explore solutions that cannot be constructed by varying the filtering and parallelism thresholds.

The study reported here is limited to one APDA (Split Miner). A possible direction for future work is to define and evaluate extensions of this approach for other DFG-based APDAs such as Fodina and Inductive Miner. Also, the approach focuses on improving F-score, while it could be applied to optimize other objective functions (e.g. combinations of F-score and model complexity) or to perform Pareto-front optimization, i.e. finding Pareto-optimal solutions with respect to multiple quality measures. Finally, this study only considered four S-metaheuristics. There is room for investigating other metaheuristics or other variants of simulated annealing, e.g. using different cooling schedules. Finally, the paper only considered one baseline approach that uses a P-metaheuristics (ETM). A more detailed comparison of tradeoffs between S-metaheuristics and P-metaheuristics in this setting is another avenue for future work.

Acknowledgements. We thank Raffaele Conforti for his input to an earlier version of this paper. This research is partly funded by the Australian Research Council (DP180102839) and the Estonian Research Council (IUT20-55).

References

1. Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B., van der Aalst, W.: Measuring precision of modeled behavior. *ISEB* **13**(1), 37–67 (2015)
2. Adriansyah, A., van Dongen, B., van der Aalst, W.: Conformance checking using cost-based fitness analysis. In: *EDOC*. IEEE (2011)
3. Alizadeh, S., Norani, A.: ICMA: a new efficient algorithm for process model discovery. *Appl. Intell.* **48**(11), 4497–4514 (2018)
4. Augusto, A., Armas-Cervantes, A., Conforti, R., Dumas, M., La Rosa, M., Reissner, D.: Abstract-and-compare: a family of scalable precision measures for automated process discovery. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *BPM 2018*. LNCS, vol. 11080, pp. 158–175. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_10

5. Augusto, A., Armas Cervantes, A., Conforti, R., Dumas, M., La Rosa, M., Reissner, D.: Measuring fitness and precision of automatically discovered process models: a principled and scalable approach. Technical report, University of Melbourne (2019)
6. Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Bruno, G.: Automated discovery of structured process models from event logs: the discover-and-structure approach. *DKE* **117**, 373–392 (2017)
7. Augusto, A., et al.: Automated discovery of process models from event logs: review and benchmark. *IEEE TKDE* **31**(4), 686–705 (2019)
8. Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Polyvyanyy, A.: Split miner: automated discovery of accurate and simple business process models from event logs. *KAIS* **59**, 251–284 (2018)
9. Boussaïd, I., Lepagnot, J., Siarry, P.: A survey on optimization metaheuristics. *Inf. Sci.* **237**, 82–117 (2013)
10. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: On the role of fitness, precision, generalization and simplicity in process discovery. In: Meersman, R., et al. (eds.) *OTM 2012*. LNCS, vol. 7565, pp. 305–322. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33606-5_19
11. Burattin, A., Sperduti, A.: Automatic determination of parameters' values for heuristics miner++. In: *IEEE Congress on Evolutionary Computation* (2010)
12. Chifu, V.R., Pop, C.B., Salomie, I., Balla, I., Paven, R.: Hybrid particle swarm optimization method for process mining. In: *ICCP*. IEEE (2012)
13. Conforti, R., La Rosa, M., ter Hofstede, A.: Filtering out infrequent behavior from business process event logs. *IEEE TKDE* **29**(2), 300–314 (2017)
14. de Medeiros, A.K.A.: Genetic process mining. Ph.D. thesis, Eindhoven University of Technology (2006)
15. Gao, D., Liu, Q.: An improved simulated annealing algorithm for process mining. In: *CSCWD*. IEEE (2009)
16. Glover, F.: Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **13**(5), 533–549 (1986)
17. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
18. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs containing infrequent behaviour. In: Lohmann, N., Song, M., Wohed, P. (eds.) *BPM 2013*. LNBIP, vol. 171, pp. 66–78. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06257-0_6
19. Leemans, S., Fahland, D., van der Aalst, W.: Scalable process discovery and conformance checking. *Softw. Syst. Model.* **17**, 599–631 (2016)
20. Ribeiro, J., Carmona Vargas, J.: A method for assessing parameter impact on control-flow discovery algorithms. In: *Algorithms and Theories for the Analysis of Event Data* (2015)
21. Song, W., Liu, S., Liu, Q.: Business process mining based on simulated annealing. In: *ICYCS*. IEEE (2008)
22. Stützle, T.: Local search algorithms for combinatorial problems. Ph.D. thesis, Darmstadt University of Technology (1998)
23. van der Aalst, W.: *Process Mining - Data Science in Action*. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
24. vanden Broucke, S., De Weerd, J.: Fodina: a robust and flexible heuristic process discovery technique. *DSS* **100**, 109–118 (2017)
25. Weijters, A., Ribeiro, J.: Flexible heuristics miner (FHM). In: *CIDM*. IEEE (2011)



Learning Accurate LSTM Models of Business Processes

Manuel Camargo^{1,2}✉^{ib}, Marlon Dumas¹^{ib}, and Oscar González-Rojas²^{ib}

¹ University of Tartu, Tartu, Estonia

{manuel.camargo,marlon.dumas}@ut.ee

² Universidad de los Andes, Bogotá, Colombia

o-gonza1@uniandes.edu.co

Abstract. Deep learning techniques have recently found applications in the field of predictive business process monitoring. These techniques allow us to predict, among other things, what will be the next events in a case, when will they occur, and which resources will trigger them. They also allow us to generate entire execution traces of a business process, or even entire event logs, which opens up the possibility of using such models for process simulation. This paper addresses the question of how to use deep learning techniques to train accurate models of business process behavior from event logs. The paper proposes an approach to train recurrent neural networks with Long-Short-Term Memory (LSTM) architecture in order to predict sequences of next events, their timestamp, and their associated resource pools. An experimental evaluation on real-life event logs shows that the proposed approach outperforms previously proposed LSTM architectures targeted at this problem.

Keywords: Process mining · Deep learning · Long-Short-Term Memory

1 Introduction

Models of business process behavior trained with deep learning techniques have recently found several applications in the fields of predictive process monitoring [2, 7, 13]. Such models allow us to move from predicting boolean, categorical, or numerical performance properties, to predicting what will be the next event in a case, when will it occur, and which resource will trigger it. They also allow us to predict the most likely remaining path of an ongoing case and even to generate entire execution traces of a business process (or entire event logs), which opens up the possibility of using such models for process simulation. Yet another application of such models can be found in the field of anomaly detection [8].

This paper addresses the question of how to use deep learning techniques to train accurate models from business process event logs. This question has been previously addressed in the context of predictive process monitoring by using Recurrent Neural Networks (RNNs) with Long-Short-Term Memory (LSTM)

architecture. Specifically, Evermann et al. [2] proposed an approach to generate the most likely remaining sequence of events (suffix) starting from a prefix of an ongoing case. However, this architecture cannot handle numerical variables and hence it cannot generate sequences of timestamped events. This inability to predict timestamps and durations is also shared by the approach of Lin et al. [6]. An alternative approach by Tax et al. [13] can predict timestamps but it does not use the embedded dimension of the LSTM network, which forces it to one-hot-encode categorical variables. In particular, it one-hot-encodes the type of each event (i.e. the activity to which the event refers). As a result, its accuracy deteriorates as the number of event types increases. As shown later in this paper, this choice leads to poor accuracy when applied to real-life event logs with a couple of dozen event types.

The paper addresses the limitations of the above approaches by proposing new pre- and post-processing methods and architectures for building and using generative models from event logs using LSTM neural networks. Specifically, the paper proposes an approach to learn models that can generate traces (or suffixes of traces starting from a given prefix) consisting of triplets (event type, role, time-stamp). The proposed approach combines the advantages of Tax et al. [13] and Evermann et al. [2] by making use of the embedded dimension while supporting both categorical and numerical attributes in the event log. The paper considers three architectures corresponding to different combinations of shared and specialized layers in the neural network.

The paper reports on two experimental evaluations. The first one compares alternative instantiations of the proposed approach corresponding to different architectures, pre-processing, and post-processing choices. The goal of this evaluation is to derive guidelines as to which design choices are preferable depending on the characteristics of the log. The second evaluation compares the accuracy of the proposed approach relative to the three baselines mentioned above.

The next section provides an overview of RNNs and LSTMs and discusses related work on the use of deep learning techniques in the field of process mining and predictive process monitoring. Section 3 introduces the proposed approach, while Sect. 4 presents its evaluation. Finally, Sect. 5 summarizes the contributions and findings and outlines future work.

2 Background and Related Work

2.1 RNN and LSTM Networks

Deep Learning is a sub-field of machine learning concerned with the construction and use of networks composed of multiple interconnected layers of neurons (perceptrons), which perform non-linear transformations of data [4]. The main goal of these transformations is train the network to “learn” the behaviors/patterns observed in the data. Theoretically, the more layers of neurons there are in the network, the more it becomes possible to detect higher-level patterns in the data thanks to the composition of complex functions [5].

Recurrent Neural Networks (RNN) contain cyclical connections that have been specially designed for the prediction of sequential data [10]. In this type of data, the state of an observation depends on the state of its predecessor. So the RNNs use a part of the processed output (h) of the preceding unit of processing (a cell) for the processing of a new input (X). Figure 1 presents the basic RNN cell structure. Even though, RNNs have a good performance when predicting sequences with short-term temporary dependencies, they fail to account for long-term dependencies. Long Short-Term Memory (LSTM) networks address this problem. In LSTM networks apart from the use of part of the previous output for a new processing, a long-term memory is implemented. In the long term memory, the information flows from cell to cell with minimal variation, keeping certain aspects constant during the processing of all inputs. This constant input allows to remain the coherence of the predictions in long periods of time.

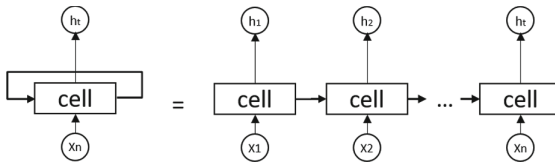


Fig. 1. RNN basic structure

2.2 Related Work

Tax et al. [13] use LSTM networks to predict the type of the next event of an ongoing process case and the time until the next event (its timestamp). In this approach, each event is mapped to a feature vector by encoding the event type using one-hot encoding and supplementing it with features related to the event's occurrence time, such the time of the day, the time since the previous event, and the accumulated duration since the start of the case. The weights in the network are set so as to minimize the cross-entropy between the ground-truth one-hot encoding of the next event and the predicted one-hot encoding as well as the Mean Absolute Error (MAE) between the ground truth time until the next event and the predicted time. The network architecture consists of a shared LSTM layer that feeds two independent LSTM layers specialized in predicting the next event and the other in predicting times. By repeatedly predicting the next event in a case and its timestamp, the authors also use their approach to predict the remaining sequence of events until case completion and the remaining cycle time. The experiments show that the LSTM approach outperforms automata-based approaches for predicting the remaining of sequence of events and the remaining time [1, 9]. In this approach the embedded dimension in LSTMs is not used to capture the event type, but instead the event type is one-hot encoded. This design choice is suitable when the number of event types is low, but detrimental for larger numbers of event types as shown later in this paper.

Evermann et al. [2] also apply LSTM networks to predict the type of the next event of a case. Unlike [13], this approach uses the embedded dimension of LSTMs to reduce the input's size and to include additional attributes such as the resource associated to each event. The network's architecture comprises two LSTM hidden layers. An empirical evaluation shows that this approach sometimes outperforms the approach of [13] at the task of predicting the next event. However, the approach focuses on predicting event types. It cannot handle numerical variables and hence it cannot predict the next event's timestamp. In this paper, we combine the idea of using the embedded dimension from [2] with the idea of interleaving shared and specialized layers from [13] to design prediction architectures that can handle large numbers of event types.

Lin et al. [6] propose an RNN-based approach, namely MM-Pred, for predicting the next event and the suffix of an ongoing case. This approach uses both the control-flow information (event type) and the case data (event attributes). The proposed architecture is composed of encoders, modulators and decoders. Encoders and decoders use LSTM networks to transform the attributes of each event into and from hidden representations. The modulator component infers a variable-length alignment weight vector, in which each weight represents the relevance of the attribute for predicting the future events and attributes. This work suffers from the same limitation as [2]: It does not support the prediction of attributes with numerical domains, including timestamps and durations.

In [7] the authors propose another approach to predict the next event using a multi-stage deep learning approach. In this approach, each event is first mapped to feature vector. Next, transformations are applied to reduce the input's dimensionality, e.g. by extracting n-grams, applying a hash function, and passing the input through two auto-encoder layers. The transformed input is then processed by a feed-forward neural network responsible for the next-event prediction. Again, this approach suffers from the same limitation as [2], namely that it does not handle numerical variables and hence it cannot predict timestamps or durations.

In [8] the authors propose a neural network architecture called BINet for real-time anomaly detection in business process executions. The core of this approach is a GRU neural network trained to predict the next event and its attributes. The approach is designed to assign a likelihood score to each event in a trace, which is then used to detect anomalies. This approach shows that generative models of process behavior can also be used for anomaly detection. In this paper, we do not consider this possible application. Instead, we focus on training models to produce sequences of timestamped events with associated roles.

In [12], the authors compare the performance of several techniques for predicting the next element in a sequence using real-life datasets. Specifically, the authors consider generative Markov models (including all-k markov models, AKOM), RNN models, and automata-based models, and compare them in terms of precision and interpretability. The results that the AKOM model yields the highest accuracy (outperforming an RNN architecture in some cases) while automata-based models have higher interpretability. This latter study addresses

the problem of predicting the next event's type, but it does not consider the problem of simultaneously predicting the next event and its timestamp as we do in this paper.

3 Approach

This section describes the method we propose to build predictive models from business process event logs. This method uses LSTM networks to predict sequences of next events, their timestamp, and their associated resource pools. Three LSTM architectures are proposed that seek to improve the learning of the network in relation to the different events logs characteristics. These architectures can accurately reproduce the behavior observed in the log. Figure 2 summarizes the phases and steps for building predictive models with our method.

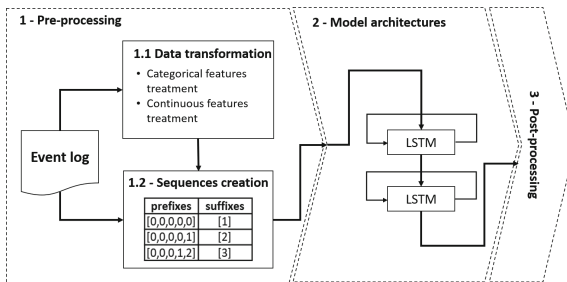


Fig. 2. Phases and steps for building predictive models

3.1 Pre-processing Phase

Data Transformation. According with the attributes nature (i.e. categorical or continuous) specific pre-processing tasks were carried looking for the improvement of the data quality for feeding the LSTM models.

Our main concern in the case of the *categorical attributes* was its transformation into numerical values to be interpreted by the LSTM network without increase the attributes dimensionality. In contrast with approaches that use one-hot encoding (i.e. process flow), which is valid to manage a reduced number of attributes and categories, our model uses activities and resources as categorical attributes. The inclusion of multiple categorical attributes looks for using more information about the process behaviour to improve the prediction accuracy. However, this multiplicity could also increment the number of potential categories exponentially. To deal with this problem, we propose the grouping of resources into roles and the use of embedded dimensions.

On the one hand, the grouping of resources into roles was performed using the algorithm described by Song and Van der Aalst [11]. This algorithm seeks to

discover resource pools (called roles in [11]) based on the definition of activity execution profiles for each resource and the creation of a correlation matrix of similarity of those profiles. The use of this algorithm allowed us to reduce the number of categories of this attribute, but keeping enough information to help the LSTM network to make more clear the differences between events.

On the other hand, the use of embedded dimensions helps in the control of the exponential attributes growth while provides more detailed information about the associations between attributes. To exemplify its advantages let's take the event log BPI 2012¹, which has 36 activities and 5 roles. If we use one-hot encoding to represent each unique pair activity-role in the event log, 180 new attributes composed by 179 zeroes are needed. This huge increment in the dimensionality is mostly composed by useless information. In contrast, only 4 dimensions are needed to encode the log if using embedded dimensions to map the categories into a n-dimensional space, in which each coordinate corresponds to a unique category. In this dimensional space the distances between points represents the how close is one activity performed by one role in relation with the same activity performed by other role. This additional information can help the network to understand the associations between events and differentiate them among similar ones. An independent network was trained to coordinate the embedded dimensions. The training network was fed with positive and negative examples of association between attributes, allowing the network to identify and locate near attributes with similar characteristics. The number of embedded dimensions was determined as the fourth root of the number of categories just to avoid a possible collision between them, according to a common recommendation used in the NLP community². The generated values were exported and reused in all the experiments as non-trainable parameters, which allowed not to increase the complexity of the models. The Fig. 3a presents the architecture of the network used for training the embedded layers, and the Fig. 3b shows a representation of the generated 4d space reduced to a 3d space for activities.

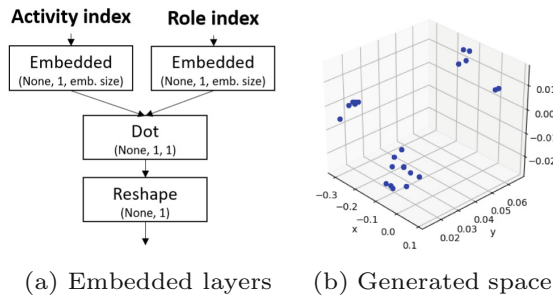


Fig. 3. Embedding network architecture and results

¹ <https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f>.

² https://www.tensorflow.org/guide/feature_columns.

In the case of *continuous attributes*, our major concern was the scaling of the values in a $[0, 1]$ range to be interpreted by the predictive models. Our model uses the relative time between activities as categorical input, calculated as the time elapsed between the complete time of one event and the complete time of the previous one. The relative time is easier to interpret by the models and is useful to calculate the timestamp of the events in a trace. However, due to the nature of each event log the relative time may have a high variability. This high variability can hide useful information about the process behaviour such as time bottlenecks or anomalous behaviours that can be hidden if the attribute scaling is performed without care. If the relative times present low variability, the use of log-normalization could also distort the perception of data. Therefore both techniques were evaluated to determine which best fits to the characteristics of the relative times. Figure 4 illustrates the results of scaling the relative times in the event log BPI 2012. In particular, the use of log-normalization makes variations in relative times clearly observable.

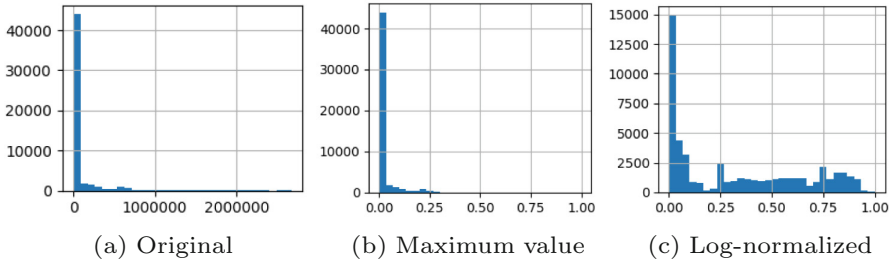


Fig. 4. Scaling of relative times over the maximum value and log-normalization

Sequences Creation. We decided to extract n-grams of fixed sizes of each event log trace to create the input sequences and expected events to train the predictive network. N-grams allow to control the temporal dimensionality of the input, and bring clear patterns of sub-sequences describing the execution order of activities, roles or relative times, regardless of the length of the traces. One n-gram is extracted each time-step of the process execution, and is done for each attribute on an independent way, this meant that for our models we count with 3 independent inputs: activities, roles and relative times. Table 1 presents five n-grams extracted from the case id 174770 of the BPI 2012 event log. The numbers in the activities, roles, and times correspond to the indexes and scaled values in the data transformation step.

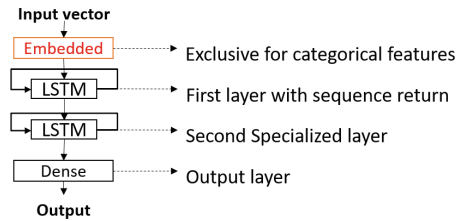
3.2 Model Structure Definition Phase

LSTM networks were used as the core of our predictive models since they are a well-known and proven technology to handle sequences, which are the nature

Table 1. N-grams for case number 174770 of the BPI 2012 event log

Time step	Activities	Roles	Relative times
0	[0 0 0 0]	[0 0 0 0]	[0. 0. 0. 0.]
1	[0 0 0 0 10]	[0 0 0 0 5]	[0. 0. 0. 0. 0.]
2	[0 0 0 10 7]	[0 0 0 5 5]	[0. 0. 0. 0. 4.73e-05]
3	[0 0 10 7 18]	[0 0 5 5 1]	[0. 0. 0. 4.73e-05 5.51e-01]
4	[0 10 7 18 5]	[0 5 5 1 1]	[0. 0. 4.73e-05 5.51e-01 1]
5	[10 7 18 5 18]	[5 5 1 1 1]	[0. 4.73e-05 5.51e-01 1 7.48e-04]

of a business process event log. Figure 5 illustrates the basic architecture of our network consisted of an input layer for each attribute, two stacked LSTM layers and a dense output layer. The first LSTM layer is in charge of provide a sequence output rather than a single value output to feed the second LSTM layer. Additionally, the categorical attributes have an embedded layer for their coding.

**Fig. 5.** Baseline architecture

Likewise, three variants of the baseline architecture were tested as is shown in the Fig. 6. The hypothesis behind these approaches is that sharing information between the layers can help to differentiate execution patterns. However, these changes could interfere with the identification of patterns in a log with high variability in relative times or in structure, generating noise in learning.

The specialized architecture (see Fig. 6a) does not share any information, in fact can be understood as three independent models. The shared categorical architecture (see Fig. 6b), concatenates the inputs related with activities and roles, and shares the first LSTM layer. Is expected that this architecture avoids the possible noises introduced by sharing information between attributes of different nature (i.e. categorical or continuous). The full shared architecture (see Fig. 6c), concatenates all the inputs and completely shares the first LSTM layer. In the evaluation section, the possibility of an architecture fits better than other in accordance with the nature of each event log is explored.

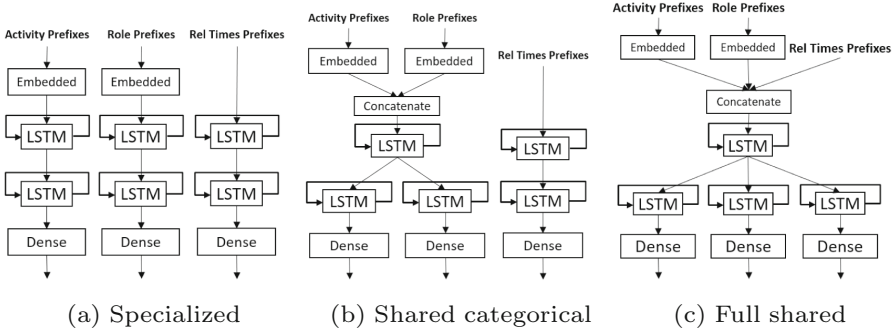


Fig. 6. Tested architectures

3.3 Post-processing Phase

Our technique is capable of generate complete traces of business processes starting from a zero prefix size. The way of doing this is by the use of continuous feedback of the model with each new generated event, until the generation of a finalization event (hallucination). This technique has been used by previous approaches, however, we explore the use of arguments of the maxima (arg-max) and random choice as techniques for the category selection of the next predicted event. Arg-max is the technique commonly used to select the next category of a prediction, and consists in selecting the one that has the highest predicted probability. In theory this technique should work well for specific prediction tasks, such as the most likely category of the next event, given an incomplete case. However, if the model is used in a generative way, it could be biased and tends to generate always the same kind of sequences, that is, the most probable ones. To avoid the this, we use the random selection of a new category following the predicted probability distribution. This attribute allows us to generate a greater number of different traces, by not getting stuck in the higher probabilities. This technique also allows us to reveals what the neural network has actually learned from the dynamics observed in the event log. Of course, the introduction of a random element, forces us to perform multiple repetitions of the experiment to find the convergence in the measurements. Both approaches were taken into account in the evaluation of results about the reproduction of the observed current state of business processes.

4 Evaluation

This section describes two experimental evaluations. The first experiment compares different instantiations of the three proposed architectures in terms of pre-processing and post-processing choices. The second experiment compares the proposed approach to the three baselines discussed in Sect. 2.2 for the tasks of next event, suffix, and remaining time prediction.

4.1 Comparison of LSTM Architectures and Processing Options

Datasets. For this experiment, we use nine real-life event logs from different domains and with diverse characteristics:

- The Helpdesk³ event log contains records from a ticketing management process of the helpdesk of an Italian software company.
- The two event-logs within BPI 2012⁴ are related to a loan application process from a German financial institution. This process is composed by three sub-processes from which we used the W sub-process in order to allow the comparison with the existing approaches [2, 12].
- The event log within BPI 2013⁵ is related to a Volvo’s IT incident and problem management. We used the complete cases to learn generative models.
- The five event-logs within BPI 2015⁶ contain data on building permit applications provided by five Dutch municipalities during a period of four years. The original event log was subdivided in five parts (one per each municipality). All the event logs were specified at a sub-processes level including more than 345 activities. Therefore, it was pre-processed to be managed at a phases level by following with the steps described in [14].

The sequence flow (SF) of each event log was classified as simple, medium, and complex according with its composition in terms of number of traces, events, activities and length of the sequences. In the same way, the time variability (TV) was classified as stable or variable according with the relation between the mean and max duration of each event log (see Table 2).

Table 2. Event logs description

Event log	Num. traces	Num. events	Num. activities	Avg. activities per trace	Max. activities per trace	Mean duration	Max. duration	SF	TV
Helpdesk	4580	21348	14	4.6	15	40.9 days	59.2 days	Simple	Stedy
BPI 2012	13087	262200	36	20	175	8.6 days	137.5 days	Complex	Stedy
BPI 2012 W	9658	170107	7	17.6	156	8.8 days	137.5 days	Complex	Stedy
BPI 2013	1487	6660	7	4.47	35	179.2 days	6 years, 64 days	Simple	Irregular
BPI 2015-1	1199	27409	38	22.8	61	95.9 days	4 years, 26 days	Medium	Irregular
BPI 2015-2	832	25344	44	30.4	78	160.3 days	2 years, 341 days	Medium	Irregular
BPI 2015-3	1409	31574	40	22.4	69	62.2 days	4 years, 52 days	Medium	Irregular
BPI 2015-4	1053	27679	43	26.2	83	116.9 days	2 years, 196 days	Medium	Irregular
BPI 2015-5	1156	36.234	41	31.3	109	98 days	3 years, 248 days	Medium	Irregular

Experimental Setup. This experiment compares different instantiations of our approach in terms of their ability to learn execution patterns and to reliably

³ <https://doi.org/10.17632/39bp3vv62t.1>.

⁴ <https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f>.

⁵ <https://doi.org/10.4121/uuid:a7ce5c55-03a7-4583-b855-98b86e1a2b07>.

⁶ <https://doi.org/10.4121/uuid:31a308ef-c844-48da-948c-305d167a0ec1>.

reproduce the behavior registered in the event log. Accordingly, we use the LSTM models to generate full event logs starting from size zero prefixes, and we then compare the generated traces against those in the original log.

We used two metrics to assess the similarity of the generated event logs. The Demerau-Levinstein (DL) algorithm measures the distance between sequences in terms of the number of editions necessary for one string character to be equal to another. This algorithm penalizes each time actions such as insertion, deletion, substitution, and transposition are carried out. Their measurements are commonly scaled by using the maximum size between the two sequences that are compared. Therefore, we use its inverse to measure the similarity between a generated sequence of activities or roles and a sequence observed in the actual event log. Then, a higher value implies a higher similarity among the sequences.

We trivially lift the DL measure (which applies to pairs of strings or traces), to measure the difference between two event logs by pairing each generated trace with the most similar trace (w.r.t. DL distance) of the ground-truth log. Once the pairs (generated trace, ground-truth trace) are formed, we calculate the mean DL between them. The Mean Absolute Error (MAE) metric is used to measure the error in predicting time-stamps. This measure is calculated by taking the absolute value of the distance between an observation and the predicted value, and then calculating the average value of these magnitudes. We use this metric to evaluate the distance between the generated relative time and those observed time, for each pair (generated trace, ground-truth trace).

We used cross validation by splitting the event logs into two folds: 70% for training and 30% for validation. The first fold was used as input to train 2000 models (approximately 220 models per event log). These models were configured with different pre-processing techniques and architectures. The configurations' values were selected randomly from the full search space of 972 combinations.

Then, new event logs of complete events are generated with each trained model (cf. techniques for the selection of the next activity described in the Sect. 3). Fifteen logs of each configuration were generated and their results averaged. More than 32000 generated event logs were evaluated.

Results and Interpretation. Table 3 summarizes the similarity results of the event logs generated from different model instantiations. The *Pre-processing*, *Model definition* and *Post-processing* columns describe the configuration used in each phase for building the evaluated models. The *DL act* and *DL roles* columns measure the similarity in the predicted categorical attributes. The *MAE* column corresponds to the mean absolute error of the cycle time of the predicted traces.

These results indicate that using this approach it is possible to train models that learn and reliably reproduce the observed behavior patterns of the original logs. Additionally, the results suggest that for the LSTM models is more difficult to learn sequences with a greater vocabulary than longer sequences. To learn these patterns, a greater number of examples is required, as can be seen in the results of BPI2012 and BPI2015. Both logs have more than 30 activities, but there is a great difference in the amount of traces (see Table 2). The high degree of similarity of the BPI2012 also suggests that the use of embedded dimensions

Table 3. Similarity results in event logs for different configurations

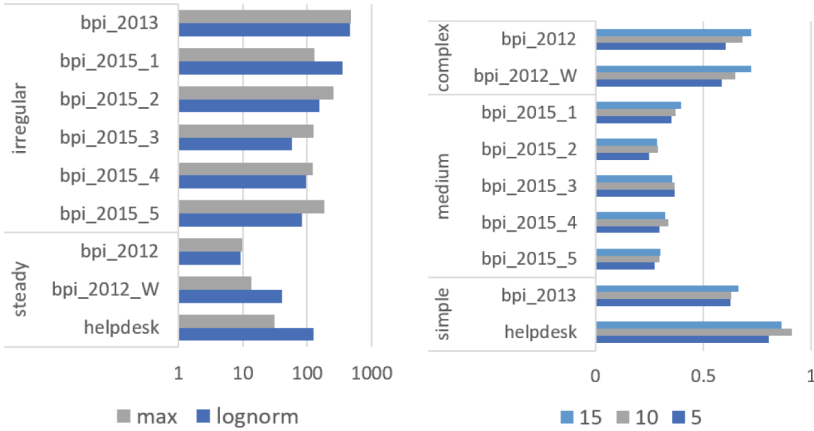
Event log	Pre-processing		Model definition	Post-processing	DL act.	DL roles	MAE (days)
	Scaling	N-gram size	Architecture	Selection method			
BPI 2012	Max	15	Specialized	Random	0.8929	0.7888	9
	Max	15	Shared cat.	Random	0.885	0.8998	9
	Lognorm	15	Concatenated	Random	0.8426	0.856	4
BPI 2012 W	Max	15	Specialized	Random	0.8742	0.8245	11.8
	Max	15	Concatenated	Random	0.7902	0.8552	7.3
	Max	10	Concatenated	Random	0.7855	0.8329	5.9
BPI 2013	Lognorm	10	Joint	Arg max	0.5442	0.698	242.6
	Max	15	Shared cat.	Random	0.7209	0.8139	471.5
	Lognorm	15	Shared cat.	Random	0.4416	0.8475	472.5
BPI 2015-1	Max	10	Concatenated	Random	0.4397	0.8048	76.6
	Lognorm	10	Specialized	Random	0.4228	0.8498	79.3
	Lognorm	10	Concatenated	Arg max	0.3642	0.5922	40.1
BPI 2015-2	Lognorm	10	Shared cat.	Arg max	0.3737	0.6228	159.4
	Max	15	Concatenated	Random	0.3462	0.8612	158.3
	Max	10	Shared cat.	Arg max	0.0431	0.1691	89
BPI 2015-3	Lognorm	10	Concatenated	Random	0.4616	0.8501	53.2
	Lognorm	5	Concatenated	Random	0.4456	0.8729	54.4
	Lognorm	15	Concatenated	Arg max	0.4255	0.7786	39.6
BPI 2015-4	Lognorm	5	Concatenated	Arg max	0.4034	0.7188	96
	Lognorm	5	Specialized	Random	0.3609	0.8248	98.8
	Max	5	Shared cat.	Arg max	0.0581	0.0968	71.1
BPI 2015-5	Lognorm	15	Specialized	Random	0.3633	0.8653	84.1
	Max	5	Shared cat.	Random	0.3323	0.9019	82.5
	Lognorm	10	Concatenated	Arg max	0.3228	0.6547	49.6
Helpdesk	Max	5	Shared cat.	Random	0.9568	0.9869	42.1
	Max	5	Joint	Arg max	0.5773	0.7368	7.3

to handle a high number of event types improves the results, so long as the number of examples is enough to learn the underlying patterns.

In relation with the architectural components evaluated in this experiment, we analyze them according to the phases to build generative models: preprocessing, model structure and hyper-parameters selection, and prediction.

Regarding the *pre-processing phase*, Fig. 7a illustrates how logs with little time variability present better results using max value as scaling technique. In contrast, logs that have an irregular structure have lower MAE using log-normalization. Additionally, Fig. 7b presents the results of DL similarity in the use of n-grams of different sizes, in relation to the structure of event logs. We can observe that the use of longer n-grams has better results for logs with longer traces, showing a stable increasing trend. In contrast, it is not clear a trend for the event logs with medium and simple structures. Therefore, the use of long n-grams should be reserved to logs with very long traces.

Regarding the *model structure definition phase*, Fig. 8 illustrates that the concatenated architecture has the lowest overall similarity. In contrast, the model architecture that only shares information between categorical attributes has the median best performance. However, it is not very distant from the specialized architecture, albeit a wider spread. This implies that sharing information



(a) Scaling of relative times results (b) N-gram size selection

Fig. 7. Preprocessing phase components comparison

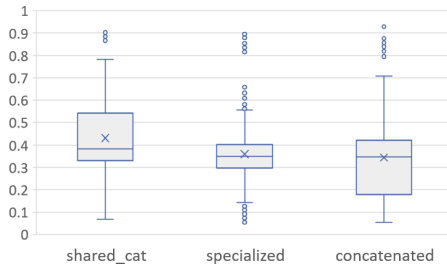


Fig. 8. Shared layers' overall similarity

between attributes of different nature can generate noise in the patterns that the network is processing, thus, hindering the learning process.

Regarding the *prediction phase*, Fig. 9 shows how random choice outperforms arg-max in all the event logs. This behaviour is even more clear in the event logs with longer and complex traces. The results suggest that random choice is advisable for assess the learning process in spite of the event log structure.

4.2 Comparison Against Baselines

Experimental Setup. The aim of this experiment is to assess the relative performance of our approach at the task of predicting the next event, the remaining sequence of events (i.e. suffixes), and the remaining time, for trace prefixes of varying lengths. For *next event prediction*, we feed each model with trace prefixes of increasing length, from 1 up to the length of each trace. For each prefix, we predict the next event and we measure the accuracy (percentage of correct

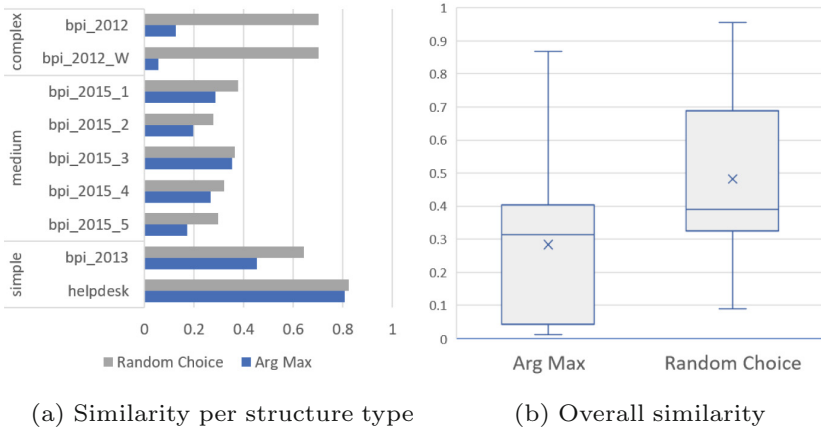


Fig. 9. Comparison of next-event selection methods

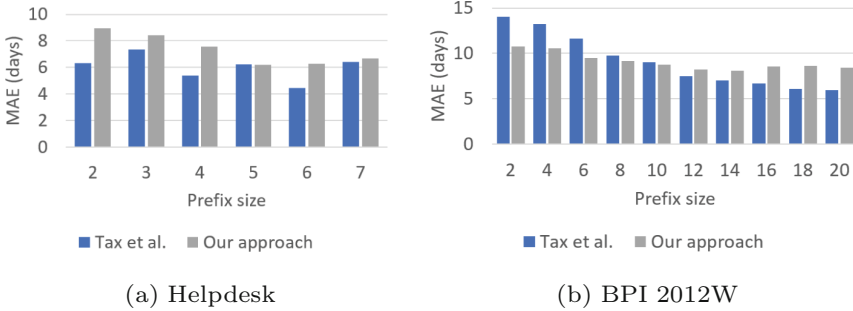
predictions). For suffix and remaining time prediction, we also feed the models with prefixes of increasing lengths. However, this time, we allow the models to hallucinate until the end of the case is reached. The remaining time is then computed by subtracting the timestamp of the last event in the prefix from the timestamp of the last hallucinated event. As in [13], we use DL as a measure of similarity for suffix prediction and MAE for remaining time prediction. For next event and suffix prediction, we use [2, 6, 13] as baselines while for remaining time prediction, we only use [13], since [2, 6] cannot handle this prediction task. We only use the Helpdesk, BPI2012W and BPI2012 event logs, because these are the only logs for which results are reported in [2, 6, 13]. The results reported for [2] for the Helpdesk and BPI2012 event logs correspond to the re-implementation of this technique reported in [6].

Results and Interpretation. Table 4 summarizes the average accuracy for the next-event prediction task and the average similarity between the predicted suffixes and the actual suffixes. For the task of next-event prediction, our approach performs similar to that of Evermann et al. and Tax et al. while slightly outperforming them for the BPI2012W event log. However, it underperforms the approach by Lin et al. For the task of suffix prediction, our approach outperforms all baselines including that of Lin et al. These results suggest that the measures adopted for the dimensionality control of the categorical attributes, allows our approach to achieve consistently good performance even for long sequences.

Figure 10 presents the MAE for remaining cycle time prediction. Even though the objective of our technique is not to predict the remaining time, it achieves similar performance at this task relative to Tax et al. – slightly underperforming it in one log, and slightly outperforming it for long suffixes in the other log.

Table 4. Next event and suffix prediction results

Implementation	Next event accuracy			Suffix prediction distance		
	Helpdesk	BPI 2012W	BPI 2012	Helpdesk	BPI 2012W	BPI 2012
Our approach	0.789	<u>0.778</u>	0.786	<u>0.917</u>	<u>0.525</u>	<u>0.632</u>
Tax et al.	0.712	0.760		0.767	0.353	
Everman et al.	0.798	0.623	0.780	0.742	0.297	0.110
Lin et al.	<u>0.916</u>		<u>0.974</u>	0.874		0.281

**Fig. 10.** Results of remaining cycle-time MAE in days

5 Conclusion and Future Work

This paper outlined an approach to train LSTM networks to predict the type of the next event in a case, its timestamp, and the role associated to the event. By iteratively predicting the next event, the approach can also predict the remaining sequence of events of a case (the suffix) and it can also generate entire traces from scratch. The approach consists of a pre-processing phase (scaling and n-gram encoding), an LSTM training phase, and a post-processing phase (selection of the predicted next event among the likely ones). The paper compared several options for each of these phases with respect to the task of generating full traces that closely match the traces in the original log. The evaluation shows that the use of longer n-grams gives higher accuracy, log-normalization is a suitable scaling method for logs with high variability, and randomly selecting the next event using the probabilities produced by the LSTM leads to a wider variety of traces and higher accuracy, relative to always choosing the most likely next event. The paper also showed that the proposed approach outperforms existing LSTM-based approaches for predicting the remaining sequence of events and their timestamps starting from a given prefix of a trace.

We foresee that the proposed approach could be used as a tool for business process simulation. Indeed, in its essence, a process simulator is a generative model that produces sets of traces consisting of event types, resources, and timestamps, from which it calculates performance measures such as waiting times, cycle times, and resource utilization. While process simulators rely on

interpretable process models (e.g. BPMN models), any model that can generate traces of events, where each event consists of an event type (activity label), a timestamp, and a resource, can in principle be used to simulate a process. A key challenge to use LSTM networks for process simulation is how to capture “what-if” scenarios (e.g. the effect of removing a task or removing a resource). To this end, we plan to apply techniques to guide the generation of event sequences from LSTM models using constraints along the lines of [3].

Reproducibility. The source code, event logs and example models can be downloaded from <https://github.com/AdaptiveBProcess/GenerativeLSTM.git>.

Acknowledgments. This research is funded by the Estonian Research Council (IUT20-55) and the European Research Council (Project PIX).

References

1. Breuker, D., Matzner, M., Delfmann, P., Becker, J.: Comprehensible predictive models for business processes. *MIS Q.* **40**(4), 1009–1034 (2016)
2. Evermann, J., Rehse, J.R., Fettke, P.: Predicting process behaviour using deep learning. *Decis. Support Syst.* **100**, 129–140 (2017)
3. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Petrucci, G., Yeshchenko, A.: An eye into the future: leveraging a-priori knowledge in predictive business process monitoring. In: Carmona, J., Engels, G., Kumar, A. (eds.) *BPM 2017*. LNCS, vol. 10445, pp. 252–268. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_15
4. Hao, X., Zhang, G., Ma, S.: Deep learning. *Int. J. Semant. Comput.* **10**(03), 417–439 (2016). <https://doi.org/10.1142/S1793351X16500045>
5. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
6. Lin, L., Wen, L., Wang, J.: MM-Pred: a deep predictive model for multi-attribute event sequence. In: *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 118–126. Society for Industrial and Applied Mathematics (2019). <https://doi.org/10.1137/1.9781611975673.14>
7. Mehdiyev, N., Evermann, J., Fettke, P.: A multi-stage deep learning approach for business process event prediction. In: *19th Conference on Business Informatics (CBI)*, vol. 01, pp. 119–128. IEEE (2017). <https://doi.org/10.1109/CBI.2017.46>
8. 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
9. Polato, M., Sperduti, A., Burattin, A., Leoni, M.D.: Time and activity sequence prediction of business process instances. *Computing* **100**(9), 1005–1031 (2018)
10. Schmidhuber, J.: Deep Learning in neural networks: an overview. *Neural Netw.* **61**, 85–117 (2015). <https://doi.org/10.1016/j.neunet.2014.09.003>
11. Song, M., van der Aalst, W.M.: Towards comprehensive support for organizational mining. *Decis. Support Syst.* **46**(1), 300–317 (2008). <https://doi.org/10.1016/j.dss.2008.07.002>
12. Tax, N., Teinemaa, I., van Zelst, S.J.: An interdisciplinary comparison of sequence modeling methods for next-element prediction. *CoRR* abs/1811.00062 (2018)

13. 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
14. Teinemaa, I., Leontjeva, A., Masing, K.O.: BPIC 2015: diagnostics of building permit application process in Dutch municipalities. BPI Challenge Report 72 (2015)

Management



Trust-Aware Process Design

Michael Rosemann^(✉)

Queensland University of Technology, Brisbane, Australia
m.rosemann@qut.edu.au

Abstract. Longitudinal studies point to the global erosion of trust in institutions and their business processes. As a result, the provision of trusted processes has become a new design criterion that exceeds the traditional Business Process Management (BPM) goals of time, cost, and quality, and also goes beyond security and privacy concerns. The notion of trust, however, has rarely been studied in the context of BPM. This paper initiates the conceptualization of trust in BPM by providing two new artefacts, i.e. a four-stage model for the design of trusted processes and a related meta model. Both have been derived from relevant theories and existing, general trust conceptualizations. Two exploratory case studies and secondary data have facilitated the identification of an initial set of application scenarios and trust requirements.

Keywords: Trust management · Uncertainty · Process design · Trustworthiness

1 Introduction

The recent Edelman Trust Barometer reports show a ‘world of distrust’ in institutions. The trust index for businesses, government, non-government organizations, and media remains low, with countries such as the USA having seen the highest decline in trust ever recorded in 2018 [1, 2]. Business processes are essential institutional artifacts and, as such, are not only severely exposed to this evaporation of trust, but badly managed processes are also a root cause of the drop in institutional trust [3]. Incidents related to organizations such as Facebook, Samsung, or Volkswagen and the spill-over effects into related industries demonstrate the need to make trust an explicit design concern. In addition to reactively dealing with the trust crisis (so called trust repair), there is also evidence that trust managed well has a positive correlation with economic performance and customer loyalty [4].

Trust is required when uncertainty exists within a business process. It is the result of a subjective assessment of this uncertainty (belief), and only if sufficient confidence exists will the process be initiated (action) [5]. For example, a customer might require a process to be completed in a certain timeframe for the decision to order. However, there is only a promised, but no certain, delivery date. Depending on the customer’s general propensity to trust [6, 7] and her believe that the process will be completed in the required timeframe, she will take the action and trigger the sales process by placing an order. In order to build the trust required, the organization could: (a) reduce process uncertainties where possible; (b) offer an insurance in case the process will be late;

and/or (c) provide additional, trust-building information (e.g., “98% of all processes are completed within 2 days”).

Unfortunately, the Business Process Management (BPM) discipline so far has little to offer when it comes to the conceptualization, design, or execution of trust-aware business processes. Trust as an artifact has not made it into process lifecycle models, BPM meta models, or BPM maturity models. There are three reasons for this. First, BPM has emerged in the age of institutional trust. Organizations, and their processes, were largely trusted and competed on economic criteria. This explains why the time–cost–quality triangle still dominates the default set of process goals [8]. As a consequence, a process might be of high transactional excellence (i.e., performs well in terms of time, cost, and quality), but the lack of trust in the process prevents it from being adopted. Second, though substantial BPM research has been invested into compliance and, more recently, the trustworthiness of a process, this research only addresses internal, objective criteria. Trust, however, is an external, subjective assessment. It is a perceptual construct and assessed differently in the context of the same process by different stakeholders depending on their individual tolerance for unpredictability (so called uncertainty avoidance). In other words, BPM research has largely assumed rational behavior and focused on evidence (“the process has a defined Six Sigma score”) - not confidence (“the customer believes that the process will deliver in time”). Third, the digital economy has empowered and connected citizens and with it provided new forms of decentralized trust in addition to the previous reliance on institutional trust. This can be seen in platforms and marketplaces where social networks have enabled users to connect quickly with members of their trusted network when executing a process. Trust networks allow access to other users or a group of users as an additional source of trust beyond the typical interaction with an institution only. Furthermore, technological solutions such as Blockchain facilitate trusted processes by providing an infrastructure for secure, local transactions within processes [9]. Embedding these new forms of non-institutional trust into BPM artifacts has rarely been the focus of research so far.

The rising importance of trusted processes, new sources of social and digital trust, and the related gap in the body of BPM knowledge motivated this research and its underlying research question, “*How can trust be embedded in the design of business processes?*” The aim is to provide an additional trust layer to guide process analysts in the design of trusted business processes.

In order to address this research question, relevant trust research and underlying theories (e.g., uncertainty reduction, information asymmetry) were studied and deployed in the context of trusted processes (as opposed to the common focus on trust in organizations or people). The insights gained shaped the methodology proposed here. We derived illustrative processes with trust requirements from two exploratory case studies: a retail bank and a consulate providing services to its nation’s citizens. In addition, secondary data has been used to identify exemplary processes with trust requirements.

This paper is structured as follows. Section 2 introduces the relevant trust-related terminology and reflects on its coverage in the context of BPM. Section 3 presents a four-stage model for the design of trusted processes, before Sect. 4 consolidates and

interrelates these elements in a meta model. Section 5 summarizes the paper and discusses its limitations and possible future research directions.

2 Trust, Trust Concerns, and Trustworthiness

2.1 Trust

The notion of trust has been comprehensively discussed and researched for decades in a variety of scientific disciplines (e.g., sociology, psychology, economics, game theory). Soellner et al. [10] summarized the status of Information Systems research in the area of trust highlighting the different types of relationships between people, technology and organizations in which trust matters. In comparison, this paper will take a narrow view and not reflect on the diverse connotations of trust in a social context. Instead, it zooms into the requirements of trust in a corporate context, with a new focus on trusted business processes. Other perspectives such as the trust an organization has in its customer (e.g., in the context of a loan application), trust as it relates to the employees involved in a process or trust in the overall purpose or integrity of an organization are out of scope within this paper.

Botsman [11] defines trust as “*a confident relationship with the unknown*”. Therefore, trust only becomes relevant if there is uncertainty. Despite ongoing attempts to create fully automated, compliant, reliable processes, there remains a residual uncertainty within nearly all processes. In fact, 100% process certainty will be in most cases economically or technically impossible and, consequently, trust will nearly always be a success factor for process adoption, unless it is a mandated process (e.g., tax declaration). Botsman’s definition allows decomposing trust into the two elements uncertainty and confidence. Uncertainty itself can be further broken down into process uncertainty, i.e. the likelihood that a process does not deliver as promised, and vulnerability, i.e. the uncertainty as it relates to the implications in case a process indeed fails to deliver.

Trust might be needed because of an information overload (e.g., which book to buy?) or a scarcity of information (e.g., will the company deliver the item in time?). Trust is a social construct; that is, different stakeholders trust the expected performance of a process differently depending on their propensity to trust, their previous experiences with the process and contextual factors [6]. A trust judgment might be made intuitively and spontaneously or could be the result of a conscious analytical reflection. Trust can be personalized (e.g., trusting a process case worker) or generalized (e.g., trust in an organization or a specific business process).

In summary, and aligned with McAlister [12], we define trust in a process as the judgment of confident reliance on this process based on positive expectations of its future behavior. A trusted business process gives stakeholders confidence to place their faith into this process in light of an outcome that is uncertain to a degree. Thus, trust consists of a belief (confidence in a future process) and an action (commitment to initiate the process). A business process can be regarded as the trustee, and an external stakeholder who relies on the process is the trustor.

Trust has a close relationship with quality and risk—two artifacts that are intensively discussed in the context of BPM. First, *quality* describes the performance of the process itself and its outcome [13]. Related research and a plethora of widely used BPM methodologies (e.g., Lean Six Sigma) have been dedicated to understanding and improving process qualities. Quality and trust have a close relationship when it comes to reliability as a popular process quality indicator. Quality and trust are also both ‘in the eye of the beholder.’ However, while quality management is a mature process-aware discipline with engineering-like methods and tools, trust management is far less advanced. Furthermore, quality is often a promise made by an organization to its customer. Whether the customer believes that the company can fulfill this promise (i.e., whether the customer trusts the process) is out of scope. Finally, a process might be of a known sub-quality standard. As there is no uncertainty about it, it would have no trust requirements.

Second, *risk* is the probability or, more broadly, the threat of an occurrence with negative implications for a process [14]. Unlike trust, risk can often be quantified and is not prone to subjective assessments (e.g., its frequency may be derived from log files). Many types of risks along a process, however, are either not relevant to an external customer (e.g., the risk to underestimate the cost of a process) or not known to the customer. Trust materializes in the subjective decision to proceed with a process in light of a risk. This includes perceiving risks where they do not exist (e.g., the inability to assess the quality of a proven technology). Not all risk-taking behavior requires trust.

The dedicated body of knowledge on trust in the context of BPM is very limited. For example, Greenberg et al. [15] studied the role of trust in the governance of outsourcing business processes. In their research the authors integrate transaction cost economy with types of trust as discussed in the Information Systems literature and differentiate contact, contract, and control as the three main stages of a process deserving a trust investigation. Berner et al. [16] discuss the notion of trusted process information in their investigation of ‘Process Visibility.’ ‘Trusted’ as an attribute of process visibility is defined here as the degree to which the business process information is perceived to be valid, reliable and objective and a positive attitude is embraced towards the source.

2.2 Trust Concerns

The design of trusted processes requires an understanding of related trust requirements. These are known as *trust concerns*, which capture those issues potentially preventing customers from engaging with a process. Trust concerns can be elicited either directly in the form of interviews or indirectly be derived by experienced process designers [17]. Mohammadi and Heisel [17] provide a comprehensive set of trust concern identification patterns. However, these are rather generic (e.g., “I am concerned about the correct functionality of the services”) and not tailored to the elements of a business process.

In our explorative case study work with two organizations (financial service, consulate) we identified a diverse set of trust concerns including issues such as:

- *Quality of a process activity*: Will the forecasted balance of my account be correct?
- *Resource expertise*: Does the person dealing with my case have the specific expertise needed?
- *Resource goal*: Can I trust the organization to incentivize their broker so that they recommend the right product to me?
- *Resource availability*: Will I get my appointment at 9.30 am, or do I have to consider waiting for a while?
- *Data*: Will you respect the privacy of my data that I provide along the process?
- *Process success rate*: Will my application to re-naturalize be successful?

As it can be seen, trust concerns are stated from an end-user perspective. This is different to statements related to risk-aware business processes which are typically made from a process-provider view.

In order to consolidate trust concerns, *trust profiles* can be used to assign users to distinct trust persona [7]. A *trust persona* is a group of users with comparable trust concerns. For example, following the diffusion of innovation model [18], one could differentiate between the very early adopters (innovators) who have a high level of trust in a new process versus skeptical, late movers who only trust a process after they have seen sufficient evidence of its performance.

Note that in addition to trust concerns, it is important to also recognize the existence of *trust opportunities*, i.e. proactively creating customer touchpoints that facilitate the development of trust. Examples for trust opportunities are Amazon re-confirming the desire to order, if a customer orders the same book for the second time or if a telecommunication company proactively downgrades a data plan to appropriately map it to the actual data consumption of its customer [19]. For the purpose of this paper, however, trust opportunities are out of scope.

2.3 Trustworthiness

“Organizations that weave trustworthiness signals into all elements of their [...] processes, over time, earn reputations of trust with their stakeholders” [20]. While trust is in the eye of the customer, organizations have various options to improve the likelihood that their processes will be trusted. They can design trustworthy processes as a response to the articulated trust concerns; that is, processes that deserve to be trusted. Trustworthiness goes beyond security concerns [21] and is discussed in Computer Science as an attribute of technical systems, including characteristics such as privacy, reliability, availability, performance, usability, etc. With regards to the human and organizational capital involved in a process, trustworthiness includes features such as honesty, competence, commitment, benevolence, intentionality, or integrity (for an overview based on a literature review of attributes of trustworthiness see Chong et al. [22]).

Mohammadi and Heisel’s [23] work on integrating trustworthiness requirements in business process models using BPMN is an important contribution of relevance for this research. Starting with identified trust concerns of the end user, specific trustworthiness

properties are derived, and detailed recommendations are made for how to model these. However, there is no discussion on how to actually build trust.

3 A Four-Stage Model for the Design of a Trusted Process

The BPM discipline has a tradition of developing dedicated perspectives to include emerging requirements and to separate concerns. These optional perspectives are complementing the core scope and methods of established process lifecycle models focused on the control flow and immediate artifacts such as data and resources. This type of research has previously included proposals to support perspectives such as risk-awareness [24], cost-awareness [25], creativity-awareness [26], privacy-awareness [27], or context-awareness [28] of business processes and is summed up in the notion of x-aware BPM [29]. However, to the best of our knowledge, there is no BPM-related research that guides academics and professionals regarding the design of *trust-aware* business processes.

In accordance with the way the previously mentioned additional perspectives have extended BPM approaches, this paper adds a trust layer to the body of BPM knowledge consisting of a methodology and a meta model (Sect. 4) to formalize the integration. The methodology for trust-aware process design comprises four stages. The first stage covers the positioning of trust within the context of business processes whereas the following three stages, inspired by Botsman's trust definition above, address uncertainty, vulnerability and confidence as the core elements of trust.

- (1) *Identify moments of trust*: Identify the steps in a process where external stakeholders make a decision that requires trust.
- (2) *Reduce uncertainty*: Address operational, behavioral and perceptual issues so that the overall process uncertainty is reduced.
- (3) *Reduce vulnerability*: Reduce the vulnerability of the customer in case the process does indeed not perform as expected.
- (4) *Build confidence*: Create a positive bias despite uncertainty and vulnerability.

3.1 Identify Moments of Trust

Trust is needed along a process when the external party is about to make a decision and uncertainty exists. For example, a customer might have proceeded in an online sales process to the point where items have been selected, put in the shopping basket, and a decision in the form of hitting the 'order now' button needs to be made. In this moment, the customer will have to trust that the items can be shipped by the desired delivery date, assuming that a late arrival would make these items unusable. This touchpoint when trust matters is called the *moment of trust*.

A conceptualization of moments of trust, therefore, is required to capture the decision point and the uncertainty that the customer is concerned about. Similar to the modeling of risks [24], the latter requires to be selective as there could be a plethora of uncertainties at a moment of trust (e.g., Will my data be protected if I share these? Will the product be of the desired quality? Has the company complied with all standards?).

As a moment of trust locates one or many *trust concerns* within a process, the severity of these concerns can be used as a measure of priority. As trust concerns are assigned to *trust persona*, the relevance of a moment of trust for different external stakeholders can be assessed. The proposed notion for moments of trust is as follows:

```
Name (moment of trust) [Object of uncertainty [process|
activity|resource|data]: Name (trust concern1), Name (trust
concernn) ]
```

The following schematic (Fig. 1) shows an example from one of our exploratory case studies of how moments of trust, trust concerns, and related trust persona could be embedded in a process. In this case, applicants for re-naturalized citizenship have two process-related trust concerns at the beginning of the process: (1) Applicants are concerned regarding the success rate of the process (Is it worthwhile to put in all the effort and to pay the application fee?) and (2) Will the process be finished in less than a year? This moment of trust is only relevant for applicants who live outside the country, as there is an awareness that this fact has a negative impact on the success rate and increases the processing time. At the next step, all applicants have a shared data-related trust concern with regards to data security. As this is of relevance for all users, no specific trust persona is associated.

Beyond what has been visualized here, one could envisage that an increased relevance and importance of trust might motivate organizations to assign dedicated *trust owners* (similar to the notion of risk owners) to specific trust concerns or moments of trust as part of their overall trust governance. A translation to specific notations (e.g., BPMN) has not been defined yet but is straightforward.

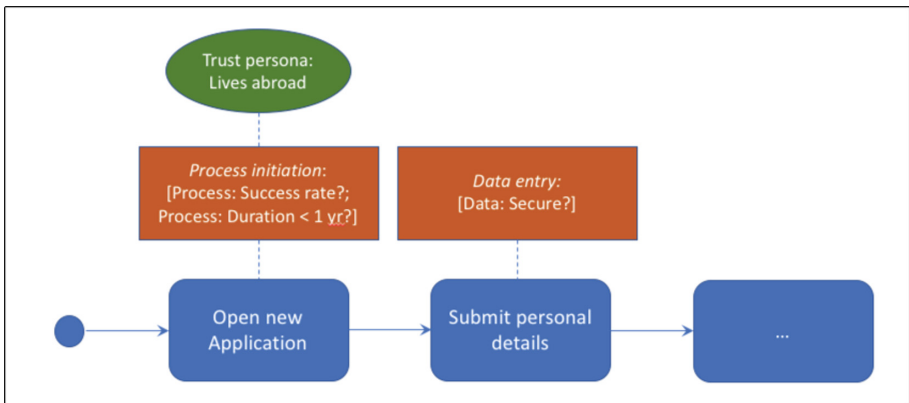


Fig. 1. Example of *moments of trust* and *trust persona* within a business process

3.2 Reduce Process Uncertainty

Trust is only needed when the trustor is uncertain about the performance of the trustee (in this case, the process). If uncertainty is the root cause of a trust concern, reducing this root cause is the most effective way to manage trust. The less uncertainty there is in a process, the higher is the trust in the process.

Process uncertainty comes in three forms, (1) operational uncertainty and (2) behavioral uncertainty which both create actual process *variation*, and (3) perceived uncertainty which is grounded in the lack of process *visibility*.

Reducing Variation (Operational Uncertainty). BPM has a track record of addressing systemic process uncertainty. Uncertainty in a process can be the result of an uncertain activity (e.g., high variation in processing time for this activity), an uncertain resource (e.g., varying levels of qualification), uncertain data (e.g., securing data privacy), or overall process uncertainty (e.g., processing time).

As a common BPM approach, Six Sigma is dedicated to reducing variation; that is, maximizing certainty. The higher the Six Sigma score, the lower the deviation of the outcomes and, as a result, the higher the process certainty. For example, a high Six Sigma score for an airport security process would mean that passengers do not have to rely on a trust-based judgment of the likely processing time, but can expect a predictable throughput time. In the ideal form, uncertainty completely disappears, and a process would show law-like performance. For example, if artificial intelligence (AI)-based interpretations of x-rays reach a level of 100% accuracy for certain types of diagnoses, this process activity converts from an uncertain, human-dependent task to an activity with certainty that equals the predictability of results on a calculator.

Using the case above (Fig. 1), the process-related trust concern regarding the processing time could be addressed by scalable resources. For example, accounting firms providing tax advice tend to scale up their workforce using liquid workforces to ensure they can deliver on their promise (e.g., a 24-h turn-around time).

Reducing Variation (Behavioral Uncertainty). Systems, processes, and business rules often provide freedom for staff in terms of how to perform their allocated tasks in detail. In these cases, staff are not guided by procedural rules but by standards reflecting the organizational commitment to integrity, sincerity, and honesty. Behavioral uncertainty occurs when the customer is unsure if staff will act according to these higher-order standards. For example, a mortgage broker might be more motivated by the commission than the customer value associated with a financial product. This cause of process variation can be addressed in two ways. First, a clear articulation of values and related cultural change management can lead to a higher commitment to relevant ethical standards. Second, behavioral uncertainty can be eliminated via automation with the assumption that the related systemic uncertainty is easier to control (e.g., robo-advice in the context of a lending process).

Increasing Visibility (Perceived Uncertainty). In addition to actual certainty, perceived uncertainty can compromise the trust position of a business process. In this case, a customer does not trust a process, not because the process is in fact uncertain but because the customer believes that the process is uncertain. Perceived uncertainty is

grounded in information asymmetry [30] and best addressed by increasing the visibility of a process to overcome this asymmetry. For example, Volkswagen's glass factory makes the final stages of the manufacturing process visible to the future car owner, and open kitchens in restaurants make the cooking process visible to diners. These concepts increase the line of visibility along a process and eliminate perceived uncertainty as a factor for insufficient trust. Similarly, the ability to track delivery processes (open process monitoring), as practiced by many logistics or food delivery companies nowadays, increases the customer's knowledge of the actual arrival time and, as such, reduces uncertainty over time.

In addition to making the actual process status and its progress visible to the customer, tools such as predictive process monitoring and process forecasting can be used to increase the visibility; that is, reduce the uncertainty of future process events. For example, Uber publishes the average estimated time of arrival of its drivers as a real-time data feed at various airports.

3.3 Reduce Vulnerability

Not all forms of uncertainty within a business process can be eliminated. This could be due to technical reasons (e.g., a robotic vision algorithm has only a certain quality to detect items), the impossibility of anticipating all exceptions, reliance on human judgement, external factors (e.g., weather, traffic, suppliers), or other reasons. In this third step of trust-aware process design the focus is on reducing the vulnerability; that is, the negative implication in case the process does not deliver as promised (e.g., a delivery is late). Vulnerability can be seen as the potential cost to a trustor in case the trustee does not perform as expected. The higher this cost, the more trust is required to proceed.

In order to mitigate these costs, different types of vulnerability and corresponding compensations need to be differentiated. Three examples covering time, product, and price vulnerability are provided in the following to clarify this:

- (1) *The costs of a process being late.* For example, in 2018 the German Bundesbahn processed 2.7 million claims processes by travelers seeking reimbursements for their late trains. Such an arrangement requires an additional process capability.
- (2) *The outcome of a process might not satisfy the customer's expectations.* For example, many online retailers offer a free-of-charge return (e.g., Adidas' change-of-mind returns policy). This requires process-rollback capabilities.
- (3) *The value-for-money equation is dissatisfactory.* For example, a consulting company could re-configure its pricing algorithm in its invoicing process so that the fee is based on the value provided as opposed to being time-based. This form of vulnerability management requires a modification of a task within a process.

3.4 Building Confidence

Unfortunately, the bulk of BPM research stops at the design of trustworthy processes (e.g., Mohammadi and Heisel [17]). The conversion of a trustworthy process into a trusted process and the exploration of additional sources of trust, however, has been so

far largely out of scope. Trustworthy processes become trusted processes if the customer has confidence that the process will deliver as promised. Das and Teng [31] also show that trust complements control as a confidence builder. The relationship between confidence and trust, and how the latter goes beyond the former, is comprehensively discussed by Mayer et al. [6, p. 713].

Reducing process uncertainty and vulnerability already has an indirect positive effect on confidence. However, there are also ways to directly increase customer confidence in a business process. The fourth and final stage of process-aware trust management therefore deals with mechanisms helping with the emergence of confidence.

Once the moments of trust and related trust concerns have been identified, it needs to be explored how a stakeholder's confidence in the relationship with the identified uncertainty and vulnerability can be increased. For this, it is proposed to identify alternative *sources of trust*. A source of trust provides confidence-building information on which a decision can be made. Depending on the characteristics of the process and the profile of the trust persona—that is, individual preferences and attitudes—certain sources of trust might be more important than others. In the following, six alternative sources of trust will be discussed. For each of these, it will be briefly shown how to use process-aware information systems for these sources of trust.

Democratic Trust: Trust the Majority. A quantitative source of trust is the articulation of confidence-boosting information covering the behavior or experiences of the majority of users at the moment of trust within a process. For example, in our explorative cases we found that some citizens proactively ask questions such as ‘how many applications have been successful?’ when considering whether to lodge specialized applications such as a request to re-naturalize as a citizen. The provision of such a figure, if high enough, could act as a confidence-boosting piece of information. A similar example of a trust-building piece of information would be a metric stating the (high) number of customers (in percentage) who have not returned products after purchasing a specific product (unseen) online. A variant of democratic trust would be the upfront provision of aggregated process information. For example, Uber uses the average arrival times of their drivers (e.g., 4 min) as a trust-building mechanism for the service latency¹ of their personal mobility process. Thus, democratic trust means trust in numbers and is nurtured via a reference to the positive experiences of the majority of process customers. Democratic trust is popular in platforms using rating systems.

A process-aware information system (PAIS) could embed such sources of trust by consolidating and publishing previous process data (assuming they have confidence-building values). Similar to the idea that data usage is seen as a proxy for data quality, process usage could be seen as an indicator for process quality. This is similar to solutions that are used as part of product recommendation services (e.g., Amazon store).

¹ The service latency of a process is the time between the triggering event and the first action of the process.

More advanced systems would be able to consider the trust persona and offer only persona-specific information. For example, process success rates would be offered depending on the applicant's demographic class.

Democratic trust is of relevance in processes with limited decision complexity (limited number of alternatives) and with a high volume of users (so adequate majority scores can be sourced).

Local Trust: Trust in My Personal Network. A user might not trust the process or published frequencies (democratic trust, i.e., wisdom of the crowds). However, there might be trusted people whose (positive) experiences with the process could be confidence-building (i.e., wisdom of friends). This requires finding a way to transfer the trust into these contacts towards trust in the process.

Trust networks capture relationships with trusted people. An existing trusted relationship can be activated as a source of trust in the context of a business process. A trust network can either be explicit—that is, a user expresses in a trust statement (e.g., as a score between 0–1) the extent to which they trust another user—or it can be implicit—that is, trust is inferred from a relationship between two people. Examples of the latter are connections in social networks such as Facebook or LinkedIn. The assumption is that the mutual willingness to create a connection is grounded in mutual trust. If the user can be motivated to share their trust network at the moment of trust in a process, trusted users can be identified. For example, Airbnb Social Connections allow connecting Airbnb with Facebook during the booking process to see if someone from the potential guest's personal network is a personal friend of the host or has reviewed the host. Such a connection could be seen as trust-building for the person using the Airbnb booking process and reduces uncertainties regarding an essential resource in this process; that is, the host.

A concern with this source of trust is that the trust network might reflect a filter bubble; that is, like-minded sources of trust and not objective sources are activated. Unlike democratic trust, this source of trust requires: (1) involvement of the user who has to be incentivized to share their trust network, (2) a track record of members of this trust network who have previously engaged with the process, and (3) the ability to identify these users. The quality of local trust depends on the size and accessibility of the trust network. Thus, local trust is more relevant for community processes (e.g., retail, entertainment, or personal services such as UrbanSitter) with limited privacy concerns as opposed to specialized processes (e.g., health care).

Global Trust: Trust in Respected Users. In more advanced decision-making processes that go beyond simple yes or no decisions, majority statements (of a general population or sourced from a local trust network) might not be sufficient. Instead, what is needed is the confidence that can only be derived from a person with a high commonly accepted reputational value (e.g., intensive user of a specific process). This can be found, for example, in Amazon's book recommendations, Wikipedia's selection of editors, or LinkedIn endorsements.

This source of trust requires access to process users with a high reputational standing. The user could either leave a track of 'endorsed processes' or could be proactively contacted as a source of trust. An example for the former would be the training processes of amateur athletes who follow the training processes of recognized

athletes. An example of the latter is the case when during the process of hiring a graduate the recruiting company contacts the academic experts who taught the relevant candidate.

As can be seen, global trust matters for processes when higher levels of expertise are needed to select a process variant (or a resource within a process). In these cases, the reputational standing and the specific expertise matter more than the personal connection (local trust) or the view of the majority of process users (democratic trust).

Specific Trust: Trust in People Like Me. In some cases, trust can only be derived from users who share a number of essential attributes with the user who is about to engage with a process. For example, the platform Patientslikeme.com is a network of more than 600,000 users that allows people with health issues to identify other patients with similar demographics and symptoms who have successfully overcome the identified health problem. A study of the behavior of a specific class of patients (e.g., 90% of them took a certain medication) can be a relevant source of trust for users when deciding about their very own therapy process. In these processes, confidence needs to be built within an individual user who could not derive trust from their trust network (as no member of this network has experience or familiarity with this medical issue). ‘People like me’ is a trust-building mechanism that, according to the Edelman Trust Barometer, ranks third highest in terms of credibility [2].

In such a scenario, the decision-maker derives the trust required from an endorsement or the process behavior of a person with shared health circumstances or attributes. This type of reasoning underlies predictive process monitoring solutions that identify similar tokens from the process log file. In the context of trust management, this would require: (1) identifying similar tokens, and (2) recommending process paths based on the positivity of the outcome. For example, what therapeutical process did those patients (like me) choose that led to the desired outcome; that is, they overcame the medical issue?

A PAIS will only be able to provide such data if rich insights about the process users do exist. Certain attributes might be derived based on the way the process is executed. Similar to democratic trust, the provider of the process becomes a broker to a new trusted source as opposed to local trust in the form of a trust network where the user has to bring trusted connections to the process.

These previous four forms of trust (democratic, local, global, specific) can be classified as social trust; that is, they derive trust via some form of social interaction. The next two forms of trust are based on forms of organizational trust and grounded in the process provider.

Institutional Trust: Trust in the Organization. This is the traditional source of trust. However, as indicated at the start of this paper, it is also the source of trust most in danger, as institutional trust has been on the decline [1]. Institutional trust in the context of BPM means that there is a flow-on effect of the (dis)trust into the institution to the (dis)trust in a business process provided by this organization.

Building institutional trust is largely grounded in building trustworthy processes by ways of demonstrating compliance, commitment to security and privacy, sufficiently qualified and appropriately incentivized employees, etc. Trust-building beyond these ‘hard’ process facts could be finding ways of how the trust in the organization spills

over to trust in the process. Well-known for this is the capability maturity model. Here an increased organizational maturity score is a proxy for process reliance.

The inclusion of institutional trust in a PAIS could come in the form of proactive statements that highlight the credibility of the various components of the organization (e.g., highlight maturity levels, qualifications of resources involved, ethical standards, etc.). Institutional trust may be the only form of trust for entire new processes in which no historical process data or data for social trust is available yet. Processes that become trusted because of institutional forms of trust are processes that benefit from the additional assurance that the organization can provide assets to the process, especially as a way to mitigate vulnerability (e.g., lending process in a retail bank).

Robotic Trust: Trust in Machines. The sixth and last confidence-building element in a business process is related to reliable technology as a source of trust. Robotic trust becomes relevant when the technology used within the business process has developed a level of maturity that it is commonly trusted. Examples are the use of calculators in invoicing processes or navigation systems in delivery processes. Recent digital technologies (e.g., AI, Blockchain) take robotic trust to new levels and have enabled, for example, decentralized processes such as Powerledger's peer-to-peer energy trading process. It could be easily envisaged that very soon patients will trust an AI algorithm more than the viewpoint of a radiologist within a healthcare process.

Technologies such as robotic process automation are contributing to an increased reliability of a process and thus can be seen as a new form of robotic trust. In such processes, the customer derives the confidence required from the technology used more than from the trust in the corporation providing the process.

4 A Meta Model for Trust-Aware Process Design

In order to formalize the notion of trust-aware process design, a meta model summarizing and interrelating the constructs presented above is provided.

Trust only matters when a *process* has *uncertainties*; that is, the user of the process cannot be sure that the process behaves as promised. Uncertainties in a process can be rooted in *activities*, *resources*, or *data*, matter at a *moment of trust*, and lead to *trust concerns*. The difference in the assessment of process uncertainties by different users is captured in a taxonomy of *trust persona* based on their *trust profiles*.

Organizations have three ways to address these trust concerns: reduce uncertainties, reduce vulnerability, and increase confidence.

First, they need to address process uncertainties which manifest in three alternative forms: (1) they can reduce *actual systemic uncertainties* by addressing and ultimately reducing operational process variations (e.g., via Six Sigma); (2) organizations can address *actual behavioral uncertainties* by ensuring that the staff involved in the process are performing their tasks according to values such as confidentiality, integrity, availability, security, privacy, performance, and others; and (3) organizations need to address *perceived uncertainties* by making the process more visible as a way to overcome this type of uncertainty.

Second, organizations can mitigate uncertainties grounded in the *vulnerability* of an eventually non-performing process. This requires revising processes in a way that the costs of a process not performing to a customer’s expectations are mitigated. Here we differentiate mitigating vulnerability as it relates to the *process time*, the *process outcome*, or the *process cost*. Both uncertainty management and vulnerability management lead to a *trustworthy process*.

Third, organizations can provide *sources of trust* in order to further boost the user’s confidence in the process. Depending on the process characteristics, the related decisions to be made, and the trust persona, these sources will be of different relevance. As such, we distinguish between *democratic trust*, *local trust*, *global trust*, *specific trust*, *institutional trust*, and *robotic trust* as sources of trust.

The extent to which a trust persona ends up trusting a process is a result of how well the three elements of trust management address the identified trust concerns. The summarizing meta model, showing the three clusters of user, uncertainty, and trust management, is presented in Fig. 2.

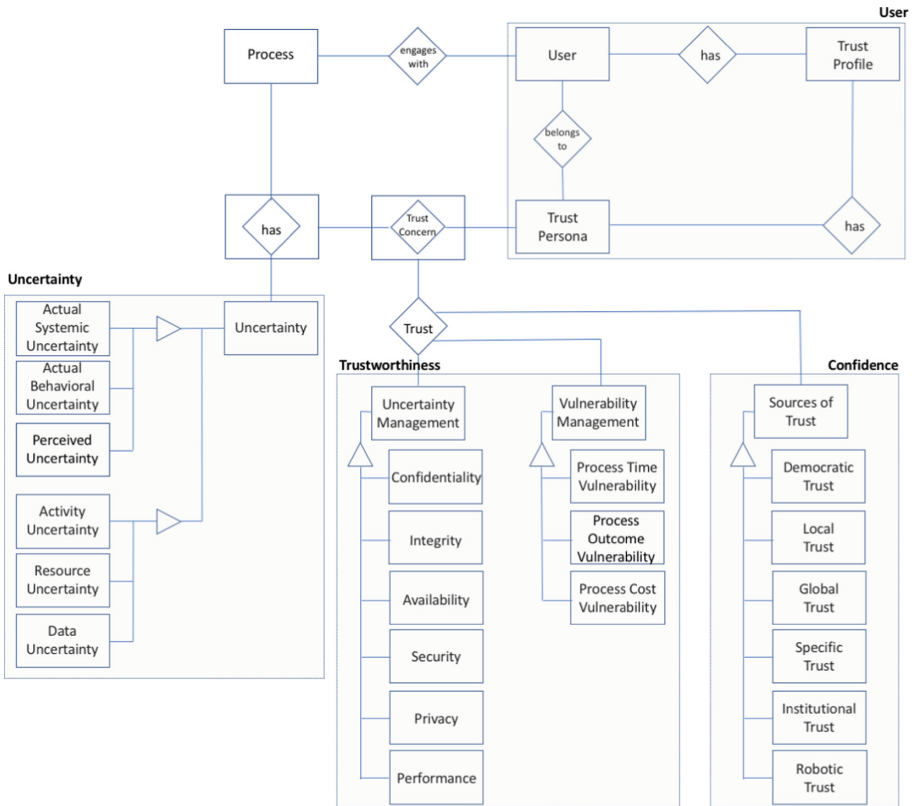


Fig. 2. Meta model for trust-aware process design

5 Conclusions and Future Work

Various cross-sectorial incidents have triggered a significant decline in the trust stakeholders have in institutions and their constituting elements, including their executives, organizational structures, resources, and business processes. At the same time, new forms of social and digital trust (e.g., Blockchain) are emerging and provide new opportunities to increase an organization's trust position. The BPM discipline, however, has not made trust a dedicated concern within its widely used artifacts, such as process lifecycle models, maturity models, or meta models.

Motivated by this demand for and the absence of knowledge in this field, this paper summarized the related body of knowledge before proposing a four-stage model of trust-aware process design. Related literature incl. underlying theories of relevance has been presented, the core trust concepts have been introduced, and ways to identify moments of trust, to reduce process uncertainty and vulnerability and to increase confidence in the process have been discussed. A consolidating meta model formally interrelates these concepts and connects the presented trust constructs as an additional layer of concern to the essential notion of a process. Real-life examples derived from two exploratory cases studies and secondary data have shown the relevance and potential applicability of this model.

This paper is conceptual in nature and, as a result, has to come with a set of *limitations*. First, further trust theories in disciplines such as sociology or economics could have been identified and improved the theoretical foundation of the trust constructs presented here. Second, a more thorough empirical investigation in the tradition of Action Design Research could have led to an iterative, empirically grounded development of the model. Third, and finally, the meta model presented is only a first step towards a more formalized articulation of trust-aware process design (including proposals for trust notations in the context of business process modelling).

As a first contribution to the domain of trust-aware process design, this paper could not cover a number of related topics that need to be addressed in *future research* to arrive at a more complete understanding and conceptualization of trust-aware processes. This includes, among others, rigorously interrelating trust with the known stages of the process lifecycle, addressing issues with regards to the actual measurement of trust and identifying relevant capabilities and including these in BPM maturity models. Substantial empirical work is required to assess the validity, reliability and overall relevance of the proposed model. Future empirical work could also lead to an initial set of trust profiles and typical trust concerns as well as ultimately lead to trust patterns; that is, process building blocks for the design of trust-aware business processes. Each proposed source of trust could attract further work, in particular in terms of its impact on users' trust, their implementation in PAIS and their alignment with process profiles; that is, what type of process requires what source(s) of trust? Researchers with a passion for BPM governance might reflect on the potential roles of trust designers and trust owners or how trust can be embedded in enterprise architectures.

References

1. Edelman Trust Barometer. Global Report. The Battle for Truth (2018). <https://www.edelman.com/research/2018-edelman-trust-barometer>. Accessed 2 June 2019
2. Edelman Trust Barometer. Global Report: Trust at Work, p. 33 (2019). <https://www.edelman.com/trust-barometer>. Accessed 2 June 2019
3. Gillespie, N., Siebert, S.: Organizational trust repair. In: Searle, R.H., Nienaber, A.-M.I., Sitkin, S.B. (eds.) *The Routledge Companion to Trust*, pp. 284–301. Taylor Francis (2018)
4. Kamers, R.: The role of trust in the B2B buying process. In: *Proceedings of 6th IBA Bachelor Thesis Conference*, Enschede (2015)
5. Sztompka, P.: *Trust: A Sociological Theory*. Cambridge University Press, Cambridge (2000)
6. Mayer, R.C., Davis, J.H., Schoorman, F.D.: An integrative model of organizational trust. *Acad. Manag. Rev.* **20**(3), 709–734 (1995)
7. Garbarino, E., Jonson, M.S.: The different roles of satisfaction, trust, and commitment in customer relationships. *J. Mark.* **63**(2), 70–87 (1999)
8. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.: *Fundamentals of Business Process Management*, 2nd edn. Springer, Berlin (2018). <https://doi.org/10.1007/978-3-662-56509-4>
9. Mendling, J., et al.: Blockchains for business process management – challenges and opportunities. *ACM Trans. Manag. Inf. Syst.* **9**(1) (2018)
10. Soellner, M., Benbasat, I., Gefen, D., Leimeister, M., Pavlou, P.A.: Trust: an MIS quarterly research curation (2016). <https://www.misqresearchcurations.org/mis-quarterly-research-curations>. Accessed 2 June 2019
11. Botsman, R.: *Who Can You Trust? How Technology Brought Us Together and Why It Might Drive Us Apart*. Public Affairs, New York (2017)
12. McAllister, D.J.: Affect and cognition-based trust as foundations for interpersonal cooperation in organizations. *Acad. Manag. J.* **38**(1), 24–59 (1995)
13. Heravizadeh, M.: *Quality-aware business process management*. Ph.D. thesis, QUT Brisbane, November 2009
14. Suriadi, S., et al.: Current research in risk-aware business process management: overview, comparison, and gap analysis. *Commun. AIS* **34**(1), 933–984 (2014)
15. Greenberg, P.S., Greenberg, R.H., Antonucci, Y.L.: The role of trust in the governance of business process outsourcing relationships: a transaction cost economics approach. *Bus. Process Manag. J.* **14**(5), 593–608 (2008)
16. Berner, M., Graupner, E., Maedche, A., Mueller, B.: Process visibility – towards a conceptualization and research themes. In: *Proceedings of the 33rd International Conference on Information Systems (ICIS 2012)*, Orlando (2012)
17. Gol Mohammadi, N., Heisel, M.: Enhancing business process models with trustworthiness requirements. In: Habib, S.M.M., Vassileva, J., Mauw, S., Mühlhäuser, M. (eds.) *IFIPTM 2016. IAICT*, vol. 473, pp. 33–51. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41354-9_3
18. Rodgers, E.M.: *Diffusion of Innovation*, 5th edn. Free Press, New York (2003)
19. Peppers, D., Rogers, M.: *Extreme Trust. Turning Proactive Honesty and Flawless Execution into Long-Term Profits*. Penguin, New York (2016)
20. Hurley, R.F., Gillespie, N., Ferrin, D.L., Dietz, G.: Designing trustworthy organizations. *Sloan Manag. Rev.* **54**, 74–82 (2013)
21. Kunze, N., Schmidt, A.U., Velikova, Z., Rudolph, C.: Trust in business processes. In: *Proceedings of the 9th International Conference for Young Computer Scientists (ICYCS 2008)*, Zhangjiajie, China (2008)

22. Chong, B., Yang, Z., Wong, M.C.S.: Asymmetrical impact of trustworthiness attributes on trust perceived value and purchase intention: a conceptual framework for cross-cultural study on consumer perception of online auction. In: Proceedings of the 5th International Conference on Electronic Commerce (ICEC 2003), Pittsburgh, Pennsylvania, USA (2003)
23. Mohammadi, N.G., Heisel, M.: Patterns for the identification of trust concerns and specification of trustworthiness requirements. In: Proceedings of the 21st European Conference on Pattern Languages of Programs, p. 31. ACM (2016)
24. Rosemann, M., zur Muehlen, M.: Integrating risks into business process models. In: Proceedings of the 16th Australasian Conference on Information Systems (ACIS 2005), Manly (2005)
25. Wynn, M.T., Low, W.Z., ter Hofstede, A.M., Nauta, W.: A framework for cost-aware process management: cost reporting and cost prediction. *J. Univ. Comput. Sci.* **20**(3), 406–430 (2014)
26. Seidel, S.: Toward a theory of managing creativity-intensive processes: a creative industries study. *IseB* **9**(4), 407–446 (2011)
27. Alhaqbani, B.S., Adams, M.J., Fidge, C.J., ter Hofstede, A.H.M.: Privacy-aware workflow management. In: Glykas, M. (ed.) *Business Process Management. SCI*, vol. 444, pp. 111–128. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-28409-0_5
28. Rosemann, M., Recker, J.: Context-aware process design: exploring the extrinsic drivers for process flexibility. In: Proceedings of the International Workshop on Business Process Modelling, Development and Support (BPMDS 2006), pp. 149–158 (2006)
29. Rosemann, M.: Proposals for future BPM research directions. In: Ouyang, C., Jung, J.-Y. (eds.) *AP-BPM 2014. LNBIP*, vol. 181, pp. 1–15. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08222-6_1
30. Akerlof, G.A.: The market for “lemons”: quality uncertainty and the market mechanism. *Q. J. Econ.* **84**(3), 488–500 (1970)
31. Das, T.K., Teng, B.-S.: Between trust and control: developing confidence in partner cooperation in alliances. *Acad. Manag. Rev.* **23**(3), 491–512 (1998)



Mining Process Mining Practices: An Exploratory Characterization of Information Needs in Process Analytics

Christopher Klinkmüller¹(✉), Richard Müller², and Ingo Weber¹

¹ Data61, CSIRO, Eveleigh, NSW, Australia
{christopher.klinkmuller,ingo.weber}@data61.csiro.au

² Leipzig University, Leipzig, Germany
rmueller@wifa.uni-leipzig.de

Abstract. Many business process management activities benefit from the investigation of event data. Thus, research, foremost in the field of process mining, has focused on developing appropriate analysis techniques, visual idioms, methodologies, and tools. Despite the enormous effort, the analysis process itself can still be fragmented and inconvenient: analysts often apply various tools and ad-hoc scripts to satisfy information needs. Therefore, our goal is to better understand the specific information needs of process analysts. To this end, we characterize and examine domain problems, data, analysis methods, and visualization techniques associated with visual representations in 71 analysis reports. We focus on the representations, as they are of central importance for understanding and conveying information derived from event data. Our contribution lies in the explication of the current state of practice, enabling the evaluation of existing as well as the creation of new approaches and tools against the background of actual, practical needs.

Keywords: Process mining · Visual analytics · Qualitative content analysis

1 Introduction

Many activities in phases of the business process management life-cycle, including process discovery, analysis and monitoring [4], benefit from the investigation of event logs that were generated during the execution of a business process. Such event data can be used to answer questions like “Does the process behave as expected?” or “Are there any bottlenecks that negatively impact process performance?”. Commonly, those high-level *domain problems* are too complex to be straightforwardly answered by applying a single analysis technique, and thus analysts divide them into more fine-grain questions, leading to lower-level *information needs* that can be satisfied through the application of analysis techniques. While this divide-and-conquer strategy enables experts to iteratively form a mental picture of the business process, analysts also “[...] often do not know what

they do not know” [19, p.43]. Consequently, the information needs are rarely predetermined, but arise from insights gained during the analysis process [7].

Research, predominantly in the field of process mining, has developed a plethora of approaches, e.g. [9, 17, 18] that enable analysts to satisfy specific types of information needs. Commercial and academic tools (like Apromore, Celonis, Disco, Everflow, Lana, myInvenio, ProM, QPR, TimelinePI, etc.) offer bundles of readily available analysis techniques. Moreover, project methodologies such as [3, 21, 23] provide universal, problem-independent guidelines for the application of such techniques in process mining projects. Due to the maturity of those research outcomes, they are increasingly adopted in real-world analysis projects, enabling us to examine those projects and elicit insights into the analysts’ work practices. So far, reviews of such projects have focused on categorizing re-occurring problems [1, 20], but lack insights into strategies that analysts choose to find answers to the domain problems. Yet, such insights would provide a foundation for further refining and enhancing the available approaches and tools.

On this basis, we aim to refine our understanding of the *relationship between the domain problems and the information needs* that arise in analysis projects. To this end, we conduct a systematic study as per [6] and analyze a corpus of 71 project reports that resulted from the problem-driven analysis of real-world event data in the context of the annual business process intelligence challenge (BPIC). While the significance of such studies was in general highlighted in [12, 13], our particular contributions to process mining, visual process analytics, and business process management are twofold. First, the schema that we use to examine work practices can serve as a general reference point for assessing existing or for ideating advanced analysis approaches. Second, we take a first step towards a shared and refined understanding of work practices in process mining projects and present a consolidated overview of such practices from a large number of analysis projects. In future work, researchers can rely on these insights to orient the design of techniques towards actual, practical needs. We also hope that our work stimulates further analysis of work practices.

Specific findings from our study show that discovery of control flow is often conducted by analysts to establish a basic understanding of the business process, whereas other problems like the investigation of the time, case or organizational perspectives constitute the actual goal of the project. Moreover, for discovery analysts heavily utilize process mining algorithms to obtain descriptive process models, indicating that the low-level analysis techniques match the domain problem well. By contrast, for other domain problems analysts rely on general-purpose techniques or tables, pointing to situations where the analysis techniques do not match the domain problems. We also derive a set of eight frequent work practice patterns to provide direction for future work.

Following, we describe our methodology including the analyzed material and discuss limitations of our study in Sect. 2. In Sect. 3, we outline the annotation schema used to systematically describe the information needs and domain problems. In Sect. 4 we present the insights from our analysis. We conclude with a summary of related work in Sect. 5 and of our findings in Sect. 6.

2 Research Methodology

In this work, we adopted a qualitative research approach, which is suitable in situations like ours where a deeper understanding of a phenomenon is developed by investigating information material [16]. To this end, we followed guidelines for qualitative content analysis [11] and applied the analysis process depicted in Fig. 1. Following, we outline each of the activities and discuss limitations.

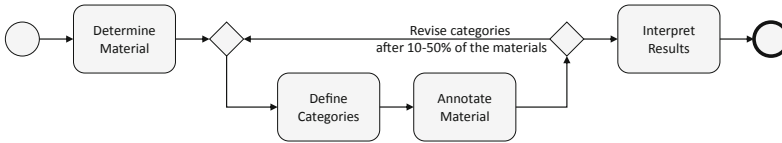


Fig. 1. The qualitative content analysis process (cf. [11])

2.1 Step 1 - *Determine Material*

As source material we used all BPIC reports available to date. The annual BPI Challenge has been organized in conjunction with the international workshop on business process intelligence¹ since 2011. Every year the challenge publishes a dataset containing real-world event logs. The dataset is provided by an organization from industry or government which asked questions related to the underlying business process (except for the first year). Upon publication of the dataset, the organizers invite analysts from academia and industry who are given a few months time to answer the questions by analyzing the dataset and to submit a report. Frequently, the analysts were invited to express any other interesting insights they obtained. Finally, a committee examines the reports and awards the best submissions. At the time of writing, eight BPIC editions were conducted and a total of 71 reports were published with 213 contributors co-authoring at least one report. The reports cover a broad range of scenarios and involve an extensive number of analysts, both from industry and academia, and therefore form a solid basis for obtaining insights into business process analysis practices.

In the study, we focused on analyzing the visual representations from those reports, including amongst others process models, charts, network diagrams, and tables. The reason is that those representations are the major means to convey information related to the underlying business process. Hence, we regard them to be representative of the low-level information needs that arose during the analysis project. Resulting from the application of specific analysis techniques they also provide an overview of those techniques' capabilities. Yet, not all representations were relevant to our study, as some do not reflect a low-level information need. For example, some representations are about the applied methodology, algorithms or tools, or the quality of a prediction model. We thus defined the

¹ <https://www.win.tue.nl/bpi/>, Accessed: 12/02/2019.

following inclusion criterion: *a visual representation must be generated from the provided event data and it must be used for explaining aspects of the underlying business process.* In total, we yielded a set of 2021 visual representations.

2.2 Steps 2 and 3 - *Define Categories and Annotate Material*

We next needed to describe the visual representations. As we wanted to analyze the descriptions and derive patterns of work practices from them, it was important that they rely on a consistent vocabulary. Thus, we followed guidelines for qualitative content analysis [11] and determined a set of categories that refer to the dimensions of the representations that we wanted to examine. The dimensions refer to the information need associated with the representations as well as the high-level questions that representations contribute to. Here, we abstract from the applied categories (details are provided in Sect. 3) and focus on the applied methodology. For each category, we then needed to define the set of codes which we used to encode the characteristics that the visual representations show with regard to the respective dimension. These sets must be *exhaustive* and *mutually exclusive* [8], so that (i) the codes cover all relevant aspects, (ii) all visual representations can be annotated appropriately, and (iii) the codes refer to distinct concepts, in order to guarantee that each representation can be described clearly and that there are no two ways of describing a visual representation.

We applied the following procedure to infer the category codes. First, we determined the categories and derived initial code sets from the literature. Then, we began to annotate the visual representations using these categories and codes. While the categories remained unchanged during the study, our code definitions occasionally underwent conceptual changes. That is, when we encountered representations that could not be described appropriately using the code set, we introduced new codes. Additionally, we sometimes experienced that our perception of a certain code changed during the annotation procedure. Due to those conceptual changes, we needed to consolidate the sets of category codes from time to time. Moreover, after a consolidation we revisited previous annotations to ensure consistency with the new schema. These updates occurred during the annotation of the first 50% of the visual representations. After that the schema was mature and could be applied without further changes. Finally, the questions posed in the challenge were annotated as well.

The annotation of visual representations itself was primarily conducted by one author of the paper, and the annotation of the challenge questions was done by another author independently. To ensure high quality of the annotations, we implemented the following procedures. First, the definition of the categories was frequently discussed by all authors. Second, the respective other authors of the paper conducted random sample checks to validate the annotations. Third, annotations that were challenging were discussed among all authors.

2.3 Step 4 - *Interpret Results*

Lastly, we derived descriptions of work practices from the annotations by summarizing and relating them, in order to identify trends in the work practices. In this context, we mostly analyzed the annotations by means of frequency distributions, and pattern mining. The results are presented in Sect. 4.

2.4 Limitations

To any study like ours, a number of limitations and threats to validity are inherent. We discuss the main factors and our approaches to mitigation below.

First, there could be personal bias: the annotation process relies on our subjective perception, and the interpretation was driven by insights relevant to us. We aimed to mitigate this issue as discussed above, but a residual risk remains.

Second, the representativeness of the data and results might be limited. Our source data stems from the BPI Challenge and might differ from process analytics practices in industry. This point is, to a degree, mitigated by the data and challenge questions stemming directly from real-world organizations, as well as by the large numbers of co-authors (>200) and visual representations (>2000).

Finally, the insights into work practices are restricted by the method of sourcing data from the *results* of these practices only. In particular, visual representations in the reports were exclusively two-dimensional and static; in contrast, analysts can interact with tools and data. Also, the reports cannot be assumed to show the full analysis process, e.g., for some information needs the analysts might not have found satisfactory results, and hence did not include any representations in the report. However, in the challenge setting with multiple teams addressing each question, this issue is partly mitigated: as long as *any* team has answered an information need, the data was included in our study. Next, visual representations were annotated based on the respective report's content and structure, which might not cover all influences that a representation had on the analysis process. Further, the choice of visual representations might be based on personal preference or tool access. To mitigate the risk of overemphasizing the visual aspects, we did not only focus on how data was presented, but we also investigated what and why data was analyzed (see Sect. 3).

While some of these limitations and threats could not be mitigated in the chosen study design, we believe the insights gained and described in the following to be of high relevance to advancing the fields of process mining and analytics.

3 The Annotation Schema: Categories and Codes

During the annotation, we focused on describing information needs and domain problems that are associated with the visual representations. According to [13], understanding these two aspects is a prerequisite for the development of data visualization tools. Hence, we defined the categories shown in Fig. 2.

The first category that we considered is the *domain problem*. It refers to the general question that was posed by the dataset provider or that the analysts

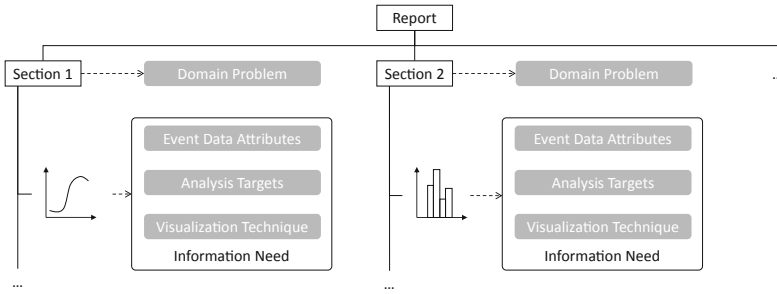


Fig. 2. Categories for the annotation of visual representations

found interesting to explore. The argumentation related to such a question is commonly not backed up by one, but by multiple visual representations. As a consequence, the first step in annotating the representations within a report was to identify the domain problems that this report examined. For each of the questions, we then introduced a conceptual section and assigned all visual representations that are related to the respective problem to that section. We also annotated the sections and thus by extension the representations with the code for the respective domain problem. The resulting conceptual document structure is oriented towards, but does *not* necessarily represent the structure of the report itself, as, e.g., some visual representations were listed in the appendix and referenced in the text, an executive summary outlined basic findings that were presented in more detail in separate sections, or the logical section structure was very fine-grained and divided visual representations by irrelevant aspects. Further, we only assigned representations to one section based on the context in which they were referenced. We hence might ignore their relevance to other sections. Yet, without further inquiry the assignment to other sections reflects our subjective interpretation, but unlikely the representations' actual influence.

We then annotated the visual representations, focusing on the information needs that are linked to them. To this end, we followed the guidelines from [13] that suggest to define a visual representation in terms of what, why, and how data is analyzed. First, we examined what part of the *event data* was used to generate the visual representation. Second, with regard to the why-dimension we focused on the *analysis target*. This category is related to the relationship in the data that is expressed by the visual representation. Finally, we captured how the data was represented by annotating the *visualization technique*. Note that some visual representations might serve multiple information needs; especially tables contained different types of data which needed to be distinguished. Consequently, we obtained 2085 information needs for the 2021 visual representations. In the following, we introduce the specific codes for each of the categories.

Domain Problem. The purpose of this category is to provide an abstract encoding for the specific domain problems that are investigated in the report. In this regard, we derived our initial set of five codes from the process mining

use cases [1] and the more general BPM use cases [20]. This set included the problems of *process discovery* where a process model describing the control flow is inferred from the data and of *conformance checking* which deals with verifying that the behavior in the event log adheres to a set of business rules, e.g., defined as a process model. While these two use cases focus on the control-flow perspective, there are three enhancement use cases which refer to other perspectives. Domain problems related to the *time perspective* deal with understanding the performance of the process such as throughput times, working times or waiting times. The *organizational perspective* focuses on the utilization of resources and their dependencies and the *case perspective* deals with the influence of other process attributes, e.g., related to the customer, on the behavior.

During the annotation process, we identified three additional domain problems. First, there are *prediction* problems where analysts aimed to create models that can forecast the development of process instances. This type is strongly related to the case perspective, as it is about comprehending the influences of attributes on the process behavior. However, given its explicit focus on prediction, we decided to capture it separately. Second, *drift detection* aims to recognize points in time at which the underlying behavior of a process changed and to provide details regarding this change. Finally, *familiarization* is an activity that helps experts to understand basic characteristics of the business process and the event data. While not necessarily related to a specific business question, we included it in our study due to its significance for the analysis process.

Event Data Attributes. This category refers to the parts of the data that the visual representation examines and is thus used to capture the attributes in the data that are investigated to satisfy the information need. The codes for this category are not based on a categorization from the literature, but were developed in the context of our study. A first set of codes refers to the entities that are examined in a visual representation. These entities include *cases* representing single process instances and *activity instances* within those cases representing the execution of a certain *activity*. An activity can belong to a *subprocess*. A case often processes an *item*, e.g., a claim, a product, or a diagnosis, and involves *external partners*, e.g., customers or suppliers, as well as *organizational entities* which perform activities or who oversee a case. Types of organizational entities include resources, departments, branches, and locations. Analysts are also interested in relationships between these entities. The *control flow* refers to constraints on the ordering of activities at the process level. The *conformance* to such a control flow definition can be examined at the individual or the aggregated case level. Similarly, *execution patterns* are related to whether a case shows a certain type of behavior or not. With regard to the organizational units, *responsibilities* are often investigated, i.e., the activities that resources work on. Additionally, analysts are interested in the *organizational hierarchy* to identify teams and they evaluate *work practices* which focus on combinations of resources that frequently work on the same cases. The last set of analysis attributes is related to timing. Here, *durations* are examined with regard to the individual or groups of cases as well as to resources and their performance. The data can also be clustered

or narrowed down by focusing on certain *time points*, such as years, months, weeks, weekdays, mornings, etc. In this context, the *execution status* of a case at a certain point is a specific derived attribute. Finally, *drift scores* provide information on how well the behavior in a case is aligned with the behavior in cases that were handled in a given time window.

Analysis Targets. There are different ways in which the attributes can be examined. In this regard, we capture the analysis targets. Here, the analysis targets specified in [14] served as a basis for our annotation. There are targets that refer to the entities within the dataset. In this context, *trends* describe overall characteristics of the entities, *outliers* are entities that do not adhere to these characteristics, and *features* are patterns that outline interesting structures within the data. Attribute-specific targets include those that are focused on single attributes: its *distribution* or its *extremes*, i.e., the minimum and maximum values. Relationships between attributes can be quantified based on their *correlation*, i.e., the degree to which their values are related. A *dependency* between attributes exists if the values of one attribute determine values of the other. Additionally, the *similarity* is a quantitative measure that is based on all values of an attribute. Finally, data might be represented as a graph to inspect its *topology*. We also recognized one additional target: *meta-information* is important for analysts to understand the attributes' meaning.

Visualization Technique. The last category refers to the visualization technique that is applied, to make the data interpretable. In this regard, we used the terminology from the data visualization catalogue² which specifies general-purpose techniques. The techniques applied in the reports are *bar chart* (including column charts and multi-set versions), *box and whisker plot*, *chloropleth map*, *chord diagram*, *heatmap*, *line graph*, *network diagrams*, *pie chart*, *radar chart*, *scatter plot*, *table*, *tree diagram*, *treemap*, *venn diagram* and *word cloud*. Detailed information on each of these techniques can be found in the catalog.

As can be expected, the source data included process-specific visualization techniques. Following our methodology, we added these to our vocabulary during annotation. Specifically, there are two types of specialized network diagrams. The *process model* depicts the control-flow of a process and the *social network* the relationships between organizational units. The *dotted chart* is a specific scatter plot used to visualize the correlation of attributes of activity instances such as timestamps, activities, resources, and cases. Finally, the *trace alignment* is a table-based technique that shows the sequences of activity instances for a set of cases and how their sequential ordering is aligned with a default ordering.

4 Analysis of Mining Practices

We now evaluate the information needs and domain problems. In particular, we describe patterns of mining practices that we detected based on our annotations. In Sect. 4.1, we provide an overview of all domain problems. We then use the

² <https://datavizcatalogue.com>.

Table 1. Distribution of the domain problems per year

	2011	2012	2013	2014	2015	2016	2017	2018	Avg.
Discovery	55.6%	28.4%	5.5%	4.8%	1.5%	0%	11%	7.3%	14.3%
Conformance	0%	3.4%	32.3%	0.9%	0%	0%	0.6%	0%	4.7%
Time pers.	0%	20.5%	0%	5.1%	19.5%	2.9%	23.5%	0%	8.9%
Org. pers.	8.3%	13.6%	3.1%	4.5%	37.9%	0%	8.7%	13%	11.2%
Case pers.	13.9%	6.3%	54.4%	60.7%	19.9%	80.3%	44.3%	24.6%	38.1%
Prediction	0%	1.1%	0%	3%	0%	0%	0.8%	1.5%	0.8%
Drift detection	0%	0%	0%	6.9%	8.8%	6.6%	0.3%	23.2%	5.7%
Familiarization	22.2%	26.7%	4.7%	14.1%	12.3%	10.2%	10.7%	30.4%	16.4%

insights to prioritize the domain problems and present a detailed analysis of the most important problems in Sect. 4.2.

4.1 Holistic View

Our first analysis focuses on the importance of the domain problems to the analysts. As an importance indicator we computed the absolute frequencies of information needs for each combination of domain problem and BPIC edition. For better comparability, we normalized the frequencies per edition, i.e., based on the total number of information needs within an edition. Table 1 shows these frequencies and their averages, per domain problem.

In the first edition in 2011, discovery was the dominating domain problem; it also was the problem that the analysts focused on the most in 2012, although the other domain problems started to receive increased attention. In the remaining editions the case perspective is the most frequently investigated problem. In this regard, 2018 is an exception where many information needs arose during familiarization and the case perspective ranked second. On average, the case perspective was the most important problem. A large share of the information needs also emerged during familiarization and discovery. Moreover, while conformance checking, prediction, and drift detection only played minor roles, the time and organizational perspectives were moderately important.

Next, we compared the importance of the domain problems assigned by the analysts to the importance assigned by the organizations that provided the datasets. To this end, we determined the problem frequencies based on the domain problems that we assigned to these questions. However, about 10% of the questions asked for any interesting insights beyond those addressed by the other questions without providing further direction; for these, we did not assign any problem. Additionally, familiarization was not present as a domain problem, as it is a task that analysts conduct to prepare for the examination of the domain problems. Similar to the reports, in the questions perspective-related problems ranked first, with the case perspective being associated with 29.8% of the questions, the organizational perspective with 14.9% and the time perspective with 10.7%. The group of conformance checking, drift detection and prediction were

Table 2. Correlation between visualization techniques and domain problems

	Discovery	Confor- mance	Time Pers.	Org. Pers.	Case Pers.	Prediction	Drift Detection	Familiar- ization
Bar Chart	6.4%	14.8%	15%	10.3%	14.3%	14.1%	14.2%	13.3%
Chord Diagram	0%	0%	0%	0%	1.7%	0%	0%	0.8%
Line Chart	2%	5.8%	7.7%	5.4%	11.5%	26.9%	6.4%	9%
Network Diagram	0%	0%	0.4%	3.1%	0.5%	0%	2.8%	1%
Pie Chart	0%	0.6%	0.4%	0.4%	2.3%	1.3%	0.7%	1.3%
Scatterplot	0%	0%	1.8%	1.8%	2.6%	10.3%	1.1%	2.1%
Tree	1.5%	0%	0.7%	1.8%	1.4%	0%	1.8%	1.3%
Other	1%	0%	0%	1.8%	1.6%	0%	1.1%	1.1%
General-purpose	10.9%	21.3%	25.9%	24.6%	36%	52.6%	28%	29.8%
Heatmap	0.5%	0%	0%	0.4%	1.6%	0%	0.7%	0.9%
Table	20.3%	41.3%	52.2%	41.1%	41.5%	34.6%	55%	42.3%
Tables	20.8%	41.3%	52.2%	41.5%	43.1%	34.6%	55.7%	43.2%
Dotted Chart	5%	0%	0%	6.3%	0.3%	0%	7.1%	2.1%
Process Model	60.4%	34.8%	21.2%	8%	15.6%	12.8%	9.2%	20.1%
Social Network	1%	0.6%	0.7%	19.6%	4.7%	0%	0%	4.3%
Trace Alignment	2%	1.9%	0%	0%	0.2%	0%	0%	0.4%
Process Mining	68.3%	37.4%	21.9%	33.9%	20.9%	12.8%	16.3%	27%

the subject of 5.3% to 10.7% of the questions. Interestingly, discovery was only posed as a domain problem by the organizations in three years and hence only 8% of the questions were related to it. We hypothesize that the mismatch between the importance of discovery for organizations and for analysts can be traced back to the relevance of discovery for establishing a basic understanding of the underlying business process. That is, in accordance with the L* life-cycle model [21] analysts rely on the insights from this activity for the investigation of the other domain problems. Consequently, for analysts discovery often played a role similar to familiarization and supported analysts in their preparation efforts.

To obtain first insights into the analysis process, we next investigated the use of visualization techniques with respect to each domain problem. We focused on the techniques, as we distinguished between general-purpose techniques, tables and those specific to process mining: dotted charts, process models, social networks, and trace alignments. Thus, the techniques provide a rough estimation for the application of process mining-specific analysis techniques. Note however that the general-purpose techniques might display event data attributes and analysis targets that were obtained from the application of process mining techniques. For each combination of domain problem and visualization technique, we computed the absolute frequencies with regard to the information needs, and

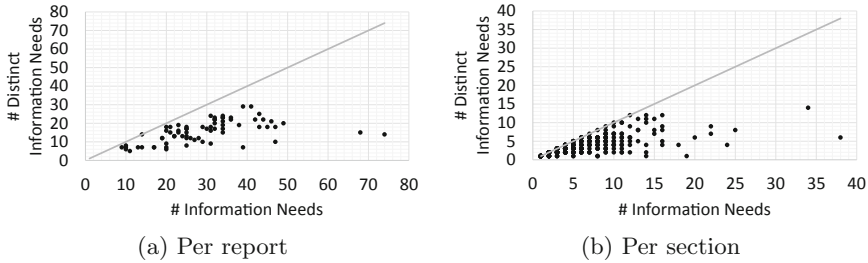


Fig. 3. Information needs in total and distinct information needs

normalized the frequencies with respect to the overall number of information needs per domain problem. Table 2 summarizes the results.

The process mining-specific techniques and especially the process models are the most important means for discovery, providing experts with important insights into the control-flow perspective. However, with regard to the other domain problems these techniques are less important. Indeed, process models are used across all problems and satisfy 17.4% of the information needs on average. Moreover, social networks play a key role for the organizational perspective. Yet, the majority of information is represented using general-purpose techniques and tables. Especially tables, as a flexible visualization technique suited for displaying high-dimensional data, are used very frequently and cover 41.6% of all information needs on average across all problems. The general-purpose techniques are applied to 28.6% of the information needs on average, with bar and line charts being the most widely adopted techniques.

The interpretation of these results must be treated with care, as they are insensitive to cases where general-purpose techniques and tables summarize the results of process mining analysis techniques. Nevertheless, the widespread use of general-purpose techniques and tables does indicate a lack of standardized approaches at the domain problem level. That is, while there are invaluable techniques that address issues at the level of information needs, there is limited support for analysts in orchestrating these techniques to understand specific domain problems. For example, discovering process models from logs is indispensable for understanding the control flow; however *discovery at the problem level* is addressed with a broader spectrum of representations than process models.

Lastly, we assessed the diversity of the analysts' information needs. To this end, we conducted the following analysis once for each report and once for each section. First, for a given section or report, we counted the information needs contained in it. Among those information needs we also determined the number of distinct information needs, i.e., where the annotations for visualization techniques, event data attributes, and analysis targets are identical. Figure 3 outlines the results. The grey line in the figure marks the equality between both measures, i.e., dots on the line are reports (a) or sections (b) where each information

need is unique. The trend in the figure shows that the analysts tend to reuse certain types of visual representations. There are two possible explanations for this observation. First, analysts might be interested in certain aspects and re-apply the same technique to analyze different snapshots of the data. Here, they might benefit from dashboard-like tools, enabling them to configure views that can dynamically be updated with different subsets of the data. Second, analysts might be familiar with only a few analysis techniques. In this case, advanced guidance approaches might help analysts to explore data from various perspectives. Yet, in order to arrive at a final conclusion further experimentation is warranted.

4.2 Details for Frequent Domain Problems

So far, we have looked at the importance of domain problems and general work practices. We now focus on the analysis of specific domain problems and the mining practices associated with them. In particular, we identify and describe frequent information needs. The explication of these needs constitutes important input for assessing and designing analysis techniques. In this regard, we focus on the two most frequent domain problems. First, we examine how analysts familiarize themselves with the data. Here, we also consider discovery problems, as our analysis revealed that discovery is often linked to the familiarization problem. Second, we focus on the case perspective as the most frequent problem.

Familiarization and Discovery. A first result stems directly from our annotation process, during which we inductively developed the codes describing the event data attributes. At the level of technique development the data model that is generally applied is a logical data model comprising *log*, *trace*, and *event* entities, relationships between them as well as a set of continuous and discrete

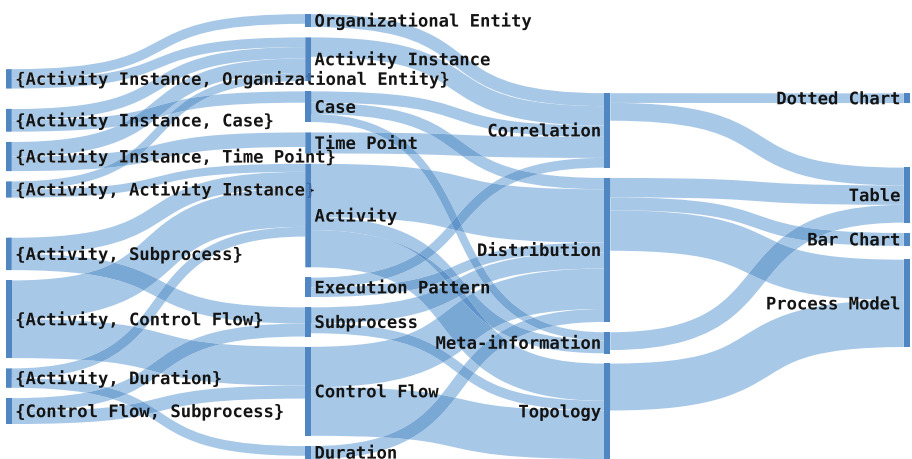


Fig. 4. Frequent analysis patterns related to familiarization and discovery

attributes describing the entities. While this level of abstraction ensures that the developed techniques are reusable, it is also free of semantics. Yet, analysts typically view the data from the conceptual standpoint and think about the data in terms of entities including activities, organizational entities, and items, as well as relationships between them including responsibilities, work practices, or the control flow dependencies. With regard to the development of analysis tools, it might thus be valuable to enable analysts to map the physical data model to a conceptual model and to conduct the analysis based on the conceptual model. Moreover, entities and attributes in this data model might be the result of a specific analysis, e.g., a social network visualization might be used to identify groups of resources within the hierarchy whose performance is later on investigated as well. Thus, tools could also support analysts in incorporating analytical results into the domain model.

To identify analysis patterns specific to familiarization and discovery, we extracted frequent pairs of annotated codes from the information needs associated with these two problems. We only considered pairs and codes that occurred in at least 5% of the information needs. Figure 4 summarizes these pairs using a parallel sets visualization. In this visualization there are four columns of nodes. Starting from the left, sets of event data attributes are depicted in the first column, event data attributes in the second, analysis targets in the third, and visualization techniques in the last. An edge depicts the frequency of a code pair or, in case of the sets of event data attributes, the frequency of attribute containment. Note that the size of the nodes is also proportional to the frequencies of the codes.

The figure shows four main types of analysis. First, process models are used to visualize the topology of the process or the control-flow, respectively. In this regard, the frequency of activities and their connections is displayed as well. Second, meta-information primarily regarding activity and case attributes is captured in tables. Third, the major category of information needs is related to understanding the distribution of cases, activities, execution patterns, and durations, and is visualized using bar charts, tables or other techniques. Fourth, analysts also investigate the correlation between a broad range of attributes including execution patterns, items, durations, time points and organizational entities. This type of information is displayed in tables, dotted charts or other types of general-purpose techniques. Additionally, Fig. 4 shows which data attributes were often examined in combination, e.g., activities and durations, activity instances and time points, etc.

Case Perspective. We repeated the above analysis for the case perspective and obtained the parallel sets visualization in Fig. 5. Here, we identified three main use cases. First, process models including the frequency of activities, their dependencies, or execution times are inspected. Process models are also used to identify execution patterns and to put them into context. Second, the distribution of subprocesses, activity instances, and execution patterns is represented using tables and various other types of general-purpose techniques. Finally, the third and main use case deals with examining the relationships between attributes. In

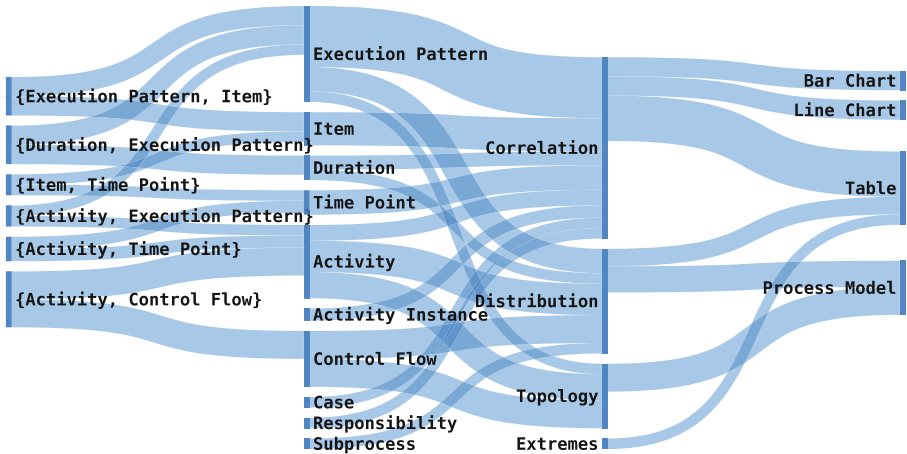


Fig. 5. Frequent analysis patterns related to the case perspective

this context, a large portion of information needs is linked to correlating execution patterns to items, durations, and responsibilities, amongst others. Here, bar charts, line charts, and tables are mainly utilized for visualization.

5 Related Work

There are two streams of research that are relevant to our study. First, there are analysis techniques and visual idioms which support analysts in the analysis of specific sub-questions. The development of visual idioms is subject to the field of *visual process analytics* and examples include the dotted chart which provides an overview of the events in an event log [18]; a technique to replay cases on top of process models [22]; or confusion matrices to compare process variants with respect to different perspectives [15]. The idioms often make use of *process mining* [21] techniques that extract knowledge from event logs, including, amongst others, the process' actual control flow (e.g., [2,9]) and its conformance to the intended behavior (e.g., [5,17]). In this paper, we focused on understanding how these techniques are applied in the context of process mining projects.

More relevant to our work are those works that focus on the work practices of analysts. On the one hand, there are methodologies for systematically approaching analysis projects, e.g., PM² [23], the L* life-cycle model [21], and the Process Diagnostics Method [3]. These methodologies comprise high-level processes including generic activities like data collection, data cleaning, and data analysis. Additionally, they provide anecdotal and exemplary evidence to outline their intended use. In contrast, we focus on explicating and analyzing the actual work practices based on empirical data. In this context, there are a few empirical studies that provide insights into the work practices. This includes catalogs of business process management [20] and process mining use case [1]. Additionally,

Martens and Verheul [10] categorized the techniques applied in the first four editions of the BPIC. Yet, these studies focus on the categorization of problems or techniques, but do not provide details insights into their relationship.

6 Findings and Recommendations

In this work, we presented a systematic study in which we examined the work practices in process mining projects based on reports that resulted from these projects. In our study, we observed that the most frequently examined problems are those referring to the analysis of perspectives other than the control-flow perspective, especially the case perspective. In this regard, our analysis revealed that the problems are largely explored via visualization techniques *not* specific to process mining, pointing to areas that might benefit more sophisticated analytical support. Additionally, the data revealed that discovery is a domain problem that organizations need to explore. Moreover, discovery is also often analyzed as part of the familiarization with the data in order to establish a basic understanding of the underlying process. Finally, we noticed that analysts rely on similar sets of visual representations when addressing different information needs. This indicates that analysts apply a work practice of defining an analysis technique and re-applying it to different data snapshots. We also presented a set of eight work practice patterns that can guide the development of advanced tools.

In future work, it would be interesting to extend the investigation of work practices by assessing the usefulness of a visual representation in the overall analysis process, as well as its contribution towards actually answering a domain question. Doing so would require interviews with analysts and business stakeholders as well as observations in laboratory settings; relying on the reports for these purposes would be too speculative.




References

1. Ailenei, I., Rozinat, A., Eckert, A., van der Aalst, W.M.P.: Definition and validation of process mining use cases. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBP, vol. 99, pp. 75–86. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_7
2. Augusto, A., Conforti, R., Dumas, M., La Rosa, M.: Split miner: discovering accurate and simple business process models from event logs. In: ICDM, pp. 1–10 (2017)
3. Bozkaya, M., Gabriels, J., van der Werf, J.: Process diagnostics: a method based on process mining. In: eKNOW, pp. 22–27 (2009)
4. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.: Fundamentals of Business Process Management. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-33143-5>
5. García-Bañuelos, L., van Beest, N., Dumas, M., La Rosa, M., Mertens, W.: Complete and interpretable conformance checking of business processes. IEEE Trans. Softw. Eng. **44**, 262–290 (2017)
6. Isenberg, P., Zuk, T., Collins, C., Carpendale, S.: Grounded evaluation of information visualizations. In: Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization, pp. 6:1–6:8 (2008)

7. Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., Melançon, G.: Visual analytics: definition, process, and challenges. In: Kerren, A., Stasko, J.T., Fekete, J.-D., North, C. (eds.) *Information Visualization*. LNCS, vol. 4950, pp. 154–175. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70956-5_7
8. Krippendorff, K.: *Content Analysis: An Introduction to Its Methodology*, 2nd edn. Sage Publications, Thousand Oaks (2004)
9. Leemans, S., Fahland, D., van der Aalst, W.: Discovering block-structured process models from event logs - a constructive approach. In: *Petri Nets*, pp. 311–329 (2013)
10. Martens, J., Verheul, P.: Social performance review of 5 Dutch municipalities: future fit cases for outsourcing? In: *BPI* (2015)
11. Mayring, P.: Qualitative content analysis. *Forum Qual. Soc. Res.* **1**(2) (2000). Article no. 20
12. Meyer, M., Sedlmair, M., Munzner, T.: The four-level nested model revisited: blocks and guidelines. In: *BELIV*, pp. 11:1–11:6 (2012)
13. Munzner, T.: A nested model for visualization design and validation. *IEEE Trans. Vis. Comput. Graph.* **15**(6), 921–928 (2009)
14. Munzner, T.: *Visualization Analysis and Design*. CRC Press, Boca Raton (2014)
15. Nguyen, H., Dumas, M., La Rosa, M., ter Hofstede, A.H.M.: Multi-perspective comparison of business process variants based on event logs. In: Trujillo, J.C., et al. (eds.) *ER 2018*. LNCS, vol. 11157, pp. 449–459. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_32
16. Recker, J.: *Scientific Research in Information Systems: A Beginner's Guide*. Springer, Berlin (2013). <https://doi.org/10.1007/978-3-642-30048-6>
17. Rozinat, A., van der Aalst, W.: Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* **33**(1), 64–95 (2008)
18. Song, M., van der Aalst, W.: Supporting process mining by showing events at a glance. In: *WITS 2007*, pp. 139–145 (2007)
19. Spence, R.: *Information Visualization - An Introduction*. Springer, Cham (2014). <https://doi.org/10.1007/978-3-319-07341-5>
20. van der Aalst, W.: Business process management: a comprehensive survey. *ISRN Softw. Eng.* **2013** (2013). Article no. 507984
21. van der Aalst, W.: *Process Mining: Data Science in Action*. Springer, Berlin (2016). <https://doi.org/10.1007/978-3-662-49851-4>
22. van der Aalst, W., de Leoni, M., ter Hofstede, A.: Process mining and visual analytics: breathing life into business process models. *BPM reports*, BPMcenter.org (2011)
23. van Eck, M.L., Lu, X., Leemans, S.J.J., van der Aalst, W.M.P.: PM²: a process mining project methodology. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) *CAiSE 2015*. LNCS, vol. 9097, pp. 297–313. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19069-3_19



Towards a Process Reference Model for Research Management: An Action Design Research Effort at an Australian University

Jeremy Gibson^(✉) , Kanika Goel , Janne Barnes,
and Wasana Bandara 

Queensland University of Technology, Brisbane, QLD 4000, Australia
{jeremy.gibson,k.goel,janne.barnes,
w.bandara}@qut.edu.au

Abstract. Increasing emphasis in the Higher Education sector for high impact research has generated a proliferation of activities aimed at supporting university research processes, commonly referred to as ‘research management’. While there has been considerable growth in this new field, it remains an elusive area, with a lacuna on what comprises good ‘research management’. A lack of common terminology and definition of the activities comprised within research management limits the capacity to provide efficient services, properly share learnings and consistently assess the effectiveness of this work.

This paper discusses the development of a research management reference model, through an Action Design Research (ADR) project conducted at a leading Australian university. The model defines 10 core domains (with areas of activities and processes within each) that constitutes the end-to-end research management process. The model was derived and validated across four ADR cycles of a detailed case study – which proved its potential value. Future research is planned to further validate the model in other universities, both within Australia and internationally.

Keywords: Research management · Reference model ·
Action Design Research · Case study

1 Introduction and Background

Research outcomes has become a core indicator of university performance, with its emphasis growing over the last two decades [5]. Many universities across the world have steadily increased their focus on research income and impact [5, 18]. In Australia alone, research income has increased nearly five-fold from 1997 to 2017 [10]. This has initiated an emerging new field of work, designed to support and oversee institutional-wide research activity, commonly referred to as ‘research management’. Kirkland et al. [18] defines research management (RM) as institutional activities, separate from the research process itself, that add value to the research activities supporting the relevant stakeholders. Ultimately, research management is about providing services that allow the researchers to do less administration and focus more on research. Research management *does* incur notable cost, which is often built-in to and funded by the research

income generated, hence, it is important that the researchers feel the benefit of these services. Recent studies (i.e. [5]) have found that universities with a higher research management index have higher research productivity, in both publications and competitive grants; indicative of the positive impacts it can bring. But the research management domain, in general is in its genesis yet.

Research management practices cover a wide spread of disciplines, including financial management and accounting, contract development, business-relationship management, training and capability building as well as the research specific fields of ethics oversight and research output management. This broad collection of disciplines is then set against the backdrop of ever-changing government legislations, the multitude of funding body requirements and internal strategy and oversight needs, resulting in complex, interdependent processes. While there are societies dedicated to guide and support research managers¹ and some frameworks that provide diverse guidance to research management at different units of analysis and levels of maturity [5, 27], there is still very little direction on what research management “best-practices” are. To date, no resource provides a holistic view on what research management entails, and null resources exist, on how to design, deliver and continuously manage the core services and underlying processes to support institution-wide research management.

Other sectors (such as Health [6], Manufacturing [2] and Finance [31]) have progressed well by applying process-centric approaches to generate efficient and effective service delivery. A process-based approach attempts to change the focus of stakeholders to a single flow of work, resulting in enhanced performance [14]. As stated by Hammer [15] (p. 7), “*Through process management, an enterprise can create high-performance processes, which operate with much lower costs, faster speeds, greater accuracy, reduced assets, and enhanced flexibility.*”

For the research management (RM) field, a first crucial (yet missing) step in this journey is to identify and define the involved ‘processes’, and this is not a trivial task, especially in complex and previously under-examined domains (like RM). While there are many learnings that can be taken from the field of project management, the domain requirements of research management (such as the funding models, legislative and ethical oversight and the segmentation between academic and professional staff) warrant a specific definition of processes. *Reference process models* have been created in diverse domains (e.g. [1, 17, 33]) as a means to address this gap. A reference model presents a synthesis of the most essential/best-practice processes of a domain, ordered in a systematic manner (in logical hierarchies, with standardized sets of actions along with their interdependencies) [7, 16, 24, 25]. A well-developed reference model can be used as a point of reference for diverse process centric purposes; it plays a vital role in the initial ‘process discovery’ [12] phase, and can be used as a source of guidance for process improvement [13].

Aiming to develop a reference model for research management of a university, this paper addresses the research question: **what are the processes fundamental for research management of a university?** An Action Design Research (ADR) is

¹ Two examples are the Society of Research Administrators International (<https://www.srainternational.org/>) and the Australasian Research Management Society (<https://researchmanagement.org.au/>).

conducted to identify and synthesize research management processes within a multi-tiered hierarchy, forming an evidence-based research management reference model.

2 Methodology

An Action Design Research (ADR) approach is adopted and deployed here, where the aim was to develop an empirically supported research management reference model which would be developed/re-specified and validated through multiple stages. Peterson and Lundberg [22] describe ADR as a means “to generate prescriptive design knowledge through learning from the intervention of building and evaluating an artefact in an organizational setting to address a problem”. ADR typically takes place within multiple cycles – each of which follows three-steps [28]: (1) problem formulation; (2) building, intervention and evaluation; and (3) reflection and learning; after which the final step is (4) formalization of learning. The stages are iterated through and built upon over the course of the work. This paper reports on the outcomes resulting after 4 ADR cycles (see Fig. 1), as applied within a single case study at the Queensland University of Technology (QUT), Brisbane, Australia.

2.1 Introducing the Case Study Context

QUT, the selected case study for this work, had undergone rapid growth as a research-intensive university in recent years. It has grown from just under Au \$15 million in research income in the year 2000 to over Au \$100 million in 2017 [10]. This rapid growth has seen a reciprocated increase in provisioning research support. Through this expansion, processes and systems around research management have proliferated and evolved to various degrees. Overlaps of some services are observed and legacy systems and underlying processes have continued to exist well after their expiry date.

A ‘Research Transformation’ project was kicked off in 2018, to evaluate the organizational model that would best support QUT’s research activity into the future. At the same time the Research Management Systems Upgrade Project, or RMSUP, was underway to replace QUT’s core research management system(s), which had been in place for over 20 years. Early phases of these projects raised the lack of a consistent way to speak about the different activities of research management and their interconnected nature. Some of the terminology used was very specific to certain areas and was creating issues and hindering process redesign². It was soon clear that a more holistic frame of reference was needed.

This described context made QUT a very suitable case study candidate for this research. Additionally, the researcher team had ready access to the case with the required support to run an ADR study.

² One example was between the Research Grants and the Commercial Research team. For the former the term ‘project’ reflected the entire lifecycle from the development of a grant application, whereas for the latter a ‘project’ was only considered to exist once a contract had been signed. This simple terminology difference had caused a large deviation between the two in both processes and how systems were used.

2.2 Developing the Research Management Reference Model: ADR Cycles Within the Case Study

Four ADR cycles were conducted, following the guidelines of Sein et al. [28], over a period of 6 months (see Fig. 1). The evolutionary model building efforts are presented here with the final model outlined in Sect. 3. The cycles began first within the ADR team (i.e. the research team members and others from the RMSUP team), where the knowledge of the team's 3 domain experts was drawn on, to obtain the first exploratory model. The next cycles were designed to engage other university-wide stakeholders to further build and evaluate the model. Over the course of these cycles, 15 experts from a variety of domains were engaged (see Table 1), with 5 other experts engaged informally. The primary means of engagement was through detailed structured walkthroughs, as this allowed for both direct feedback from the experts and observation of the effectiveness of the model as a standardized communication tool, a method used effectively in other ADR projects [22, 23]. All up, 10 such walkthroughs were conducted, involving the 15 participants, which were run by the RMSUP team, with researcher observations. They were designed to see how easily understood the reference model was, as well the completeness of the activities it detailed. Some initial framing of the model was delivered in the walkthroughs, but space was allowed to see what the experts' interpretations of terms were, to see if terminology needed to be changed. Each of these cycles are outlined further below.

Table 1. Detailed walkthrough summary

#	Attendees	Domain/Department of origin
1	Director, Office of Research Director, Office of Commercial Services	Strategic Management/Grants & Commercial Research
2	Manager, Research Development Unit	Research Grants
3	Manager, Research Partnerships	Commercial Research
4	Business Manager, Office of Commercial Services	Commercial Research
5	Director of the Office of Research Ethics and Integrity	Ethics & Governance
6	Research Governance and Compliance Coordinator	Ethics & Governance
7	Associate Director, Office of Commercial Services Project Officer, Office of Commercial Services	Commercial Research
8	Manager, Research Finance Unit	Research Finance
9	Research Quality Coordinator Research Information and Systems Support (3 members)	Reporting & Systems Support
10	Senior Research Fellow, School of Psychology	End User Academic

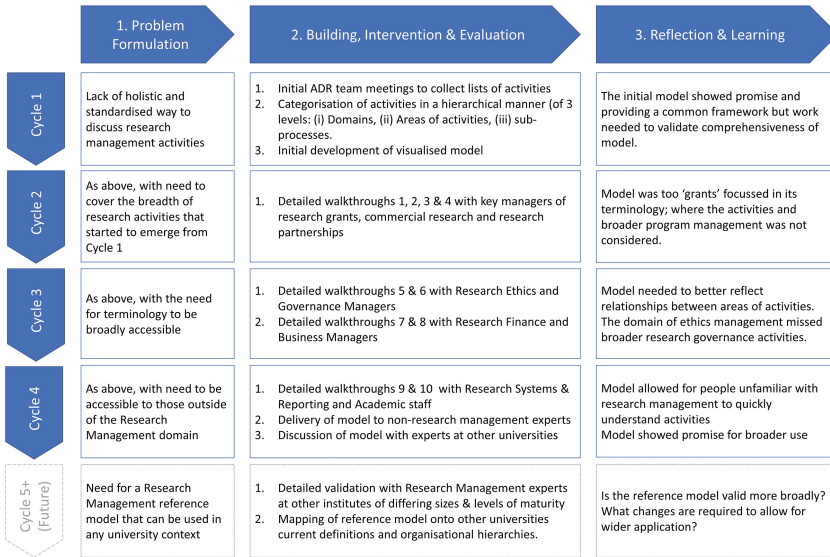


Fig. 1. Overview of ADR cycles

2.2.1 Cycle 1

The first cycle aimed to address the problem of the lack of a holistic framework to discuss research management activities. Initial discovery was synthesized from a variety of QUT corporate documentation, including service catalogues for the Division of Research and Commercialisation, organizational hierarchies as well as artefacts generated by RMSUP, such as business requirements, use cases and business process documentation. The RMSUP, through its market scan process, had been exposed to a wide variety of research management systems and was able to draw on the processes these systems supported and map this to the QUT context. Following this a series of brainstorming sessions were held with the RMSUP team to collate a list of activities that are part of research management. The RMSUP team consisted of 3 process-oriented business analysts, who had all been focused on the research management domain for the prior year and 3 subject matter experts who had worked in a series of roles across the gamut of research management for a decade each. The combination of knowledge and process-oriented thinking provided a solid basis for this initial work.

From this, a categorization of groups of activities and an overall flow (of research management activities) was recognized. This was split into a three-tier hierarchy of *domains, areas of activity and processes*³. Following this session, work was done to create a visual presentation of this hierarchy. This was conducted over the course of two weeks with the research team being embedded within the RMSUP team to allow for immediate feedback and iteration of the model. At the conclusion of this process ⁹⁴

³ Domains are high level groupings, ordered in a loose logical flow, each consisting of several areas of activity, with their own list of processes.

⁴ The final domains are presented in detail in Sect. 3.

key high-level domains were identified, each containing a collection of areas of activity and 153 different processes divided between those areas. At this stage, the model was deemed ready for initial engagement with members outside the research team.

2.2.2 Cycle 2

This engagement was focused on the area of managing research projects⁵, as this was the area at QUT with the greatest diversity of processes and terminology, and hence of key importance for validating the model.

The model showed promise in facilitating discussions around processes related to research project-management, but the language was considered by some to be too “research grants” focused. This prompted a rewrite of the activities to better address this. Additionally, the linear presentation of the model caused some people to assume it meant that all research projects must follow this path. To address this, the model was framed as not being prescriptive for every project, but more indicative of the norm.

One major discovery from these walkthroughs was the need for a 10th domain that was all-encapsulating, *program management*. This related to broader activities that the university would undertake to support research more generally. These didn’t fit well into the project-oriented process of the model, but instead helped to create a healthy ecosystem for new research projects to emerge.

2.2.3 Cycle 3

In the third cycle, the implementation of the model was expanded to new domains, looking at the areas of research ethics and finance, considering their significance to research management. At this point, the model had a domain entitled *Ethics Approval* but in discussions with the research ethics experts it became clear that there was far more involved than just ethics management. These more broadly went to activities of risk identification, mitigation and incident management. There was a common pattern of governance and compliance processes, of which ethics was just a subset. This domain underwent a name change, to *Governance & Compliance*, and a significant rewrite to accommodate these changes.

2.2.4 Cycle 4

By this cycle the model was becoming more robust and so engagement began to branch broader than active managers of research. Detailed walkthroughs were held with people working in the research system support and research reporting spaces as well as informal discussions with research managers from other institutions. Changes at this point were very minor, limited to process renaming and the addition of more connection points between domains. The model began to be used as a communication tool for people unfamiliar with research management and proved to be very effective in providing an initial grounding. This evidenced the completeness and relevance of the model within this case context, ending the cycles within the case. By the end, the total number of processes in the model had grown from 153 to 218.

⁵ Management of research projects covers activities to support funding application and the subsequent oversight of funding and project obligations. It forms part of most of the domains identified with the exception of research outputs, performance and HDR management.

3 The Research Management Reference Model

The resulting research management reference model was captured and presented across 4 levels of abstraction (following the example of other reference models, such as [4, 21]). It provides a model overview (see Fig. 2), which consists of 10 domains (see Sects. 3.1–3.10) which are ordered as a flow within the model, indicative of common practice, but not necessarily dictating a chronological flow. Each domain captures the core areas of activities that forms the domain, and there are relevant processes underlying each area⁶. While there already exists accepted language and notation to document process hierarchies (such as [8]), the end users of this model are administrators with little to no business process experience, hence a customized approach was needed to be easily comprehensible. For similar reasons the model uses icons and colors to avoid end-users disengaging with the model at first due to cognitive overload [26].

The model overview essentially presents a first high-level overview of the “life-cycle” of research management, reflecting the life of a research project, from initial conception through proposal development and approval, into conducting the project, including financial, ethical and contractual management, then into the outcomes of the project, in research outputs and the reporting around these. Note how the HDR (Higher Degree Research student) Management (domain 9, see Sect. 3.9 below) is kept separate from this, to reflect its individual processes, but a research student’s journey bares many similarities to a research project, so color-coding is used to indicate the points of overlap. Each of the domains are outlined briefly below, and the full reference model and descriptions of the areas of activity can be found at <https://doi.org/10.6084/m9.figshare.7819424>.



Fig. 2. Research management reference model overview (Color figure online)

⁶ For example, Domain (1) – ‘**Project Conception**’, consists of 6 areas (*Project Idea Initiation, Funding Sources Identification ... Review and Revision* - see Sect. 3.1) and each area has a list of clearly identified processes (see Fig. 3).



Fig. 3. Project conception

3.1 Project Conception

Project Conception refers to the initial stages of any research, where ideas are spawned, teams are formed, and proposals are developed. This work is often nebulous and difficult to define; hence it will often lack well defined processes or oversight. Within the research management field, this is often described as “pre-award”, followed by “post-award” management once the application is successful. Some universities have distinct teams to manage pre and post, with others having the same team follow the project throughout its lifecycle [29].

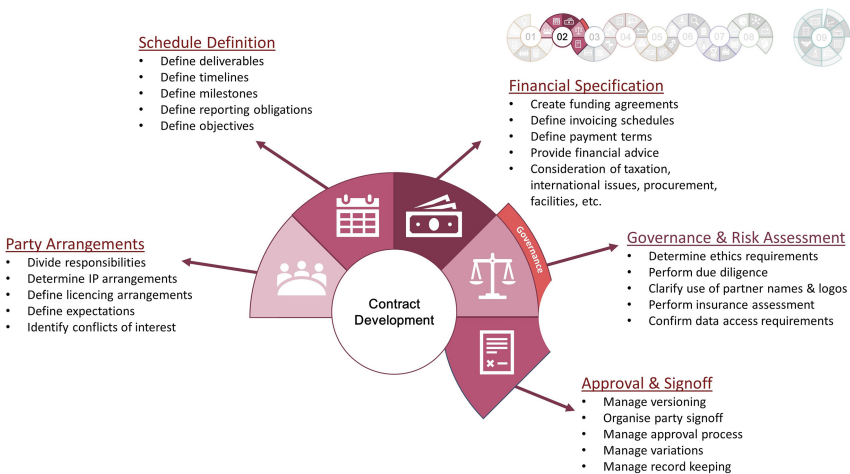


Fig. 4. Contract development

3.2 Contract Development

Contract Development (Fig. 4) covers the development of arrangements between parties and funding bodies. In some cases, such as research grants, this may occur after funding has been approved, while for commercial research the contract signing may signify the funding approval. Furthermore, there often will be multiple stages of contract development throughout the life of a project as variations are required. The domains, while placed in an order, should not be seen as stages to check-off and progress to the next, there will be overlap between them, and often multiple iterations.

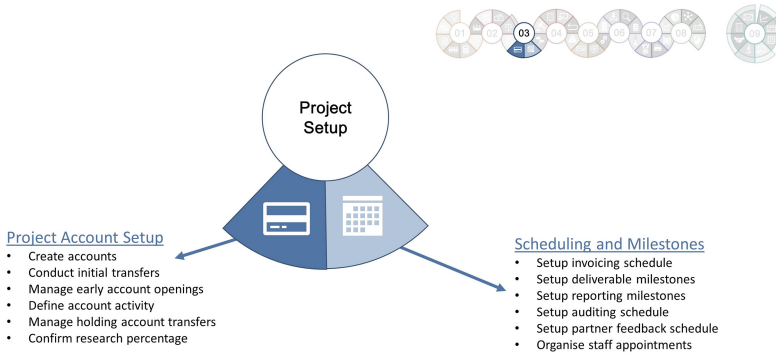


Fig. 5. Project setup

3.3 Project Setup

Project Setup (Fig. 5) refers to the administrative work that supports the project commencement. There are a number of financial administrative tasks as well as scheduling of milestones to ensure that obligations are met by all parties.

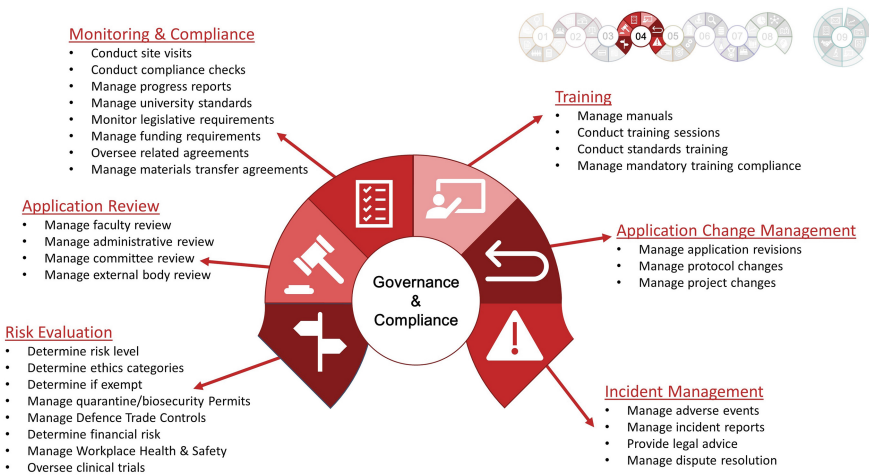


Fig. 6. Governance & compliance

3.4 Governance & Compliance

As mentioned previously, *Governance & Compliance* (Fig. 6) first began as looking at ethical management of the project but there are many other areas of project governance that follow a similar process of risk identification, mitigation strategy development, ongoing monitoring and incident reporting and management. Some of these may be managed by an ethics unit, others by different areas of the research institute and some may be managed by the researchers themselves. Due to the overlapping regulations between universities, governments and other organizations there can often be significant administrative overheads in this space, resulting in slow responsiveness [20].



Fig. 7. Project management

3.5 Project Management

Project Management (Fig. 7) outlines the administrative activities that occur during the life of the project. This domain will often cover the longest period, but the degree of administrative engagement may be minimal. However, some universities are starting to provide centralized resources to support the day-to-day management of projects [30]. This domain ensures compliance with the obligations and strategies identified in previously areas, as well as managing any changes to the project that may occur.

3.6 Research Outputs

Research Outputs (Fig. 8) covers the creation and dissemination of all outputs from the research activity. What was previously referred to as publications is increasingly being called research outputs in an effort to broaden the scope beyond standard publishing avenues. In this domain all of the outcomes of research, both tangible and intangible are considered as outputs. This includes traditional publications but also research data, commercial outputs (such as IP, patents or business relationship), media engagements

and real-world research impact. All of these items have value to a research institute and can feed into future research, but the standard reporting models of journal articles will fail to consider these.

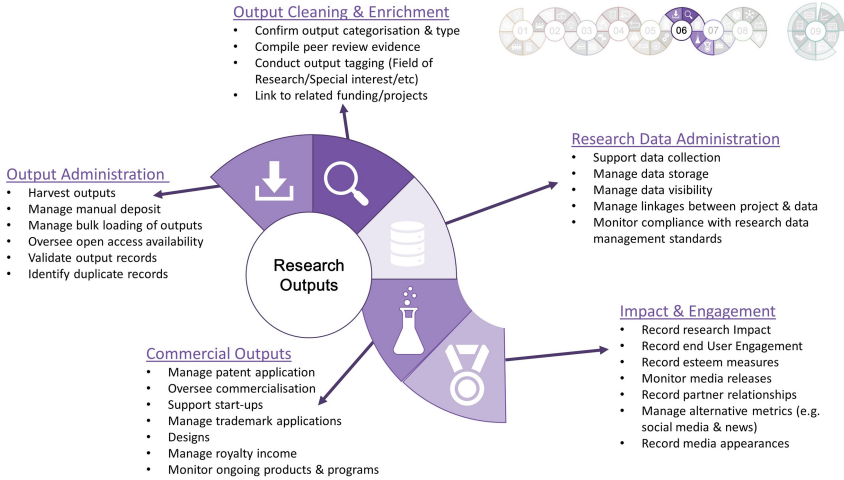


Fig. 8. Research outputs

3.7 Project Finalization

Project Finalization (Fig. 9) covers the stages of “clean-up” that need to occur once a project is completed. This can often be difficult as while certain projects may have a nominal date of completion, the various stages of final delivery and client acceptance, as well as final expenditure of funds, can continue long past this initial date as project timelines slip. Additionally, contracts may be extended or renewed, meaning that what was initially meant to be a 6-month project can continue for many years.

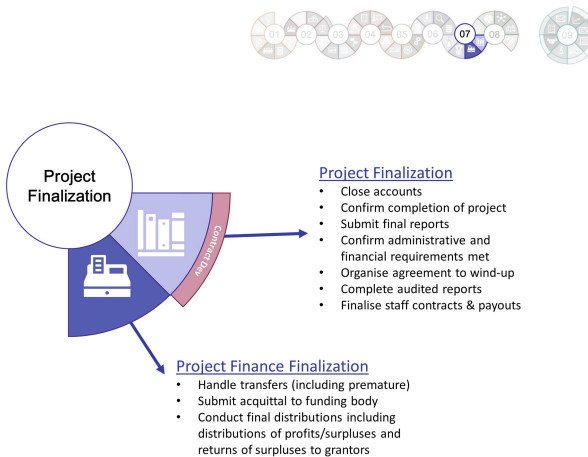


Fig. 9. Project finalization



Fig. 10. Research performance

3.8 Research Performance

Research Performance (Fig. 10) details the reporting and analysis applied to research activity. The fruits of the previous stages are collated and presented to give a clear picture of how a research institute is performing. To properly be able to demonstrate performance at this stage, data collection and aggregation needs to underpin all of the previous domains. Also benchmarks and KPI’s are used here to indicate expected levels of performance. External metrics, such as journal rankings or citation counts are used to give an indication of the quality of the research activity.

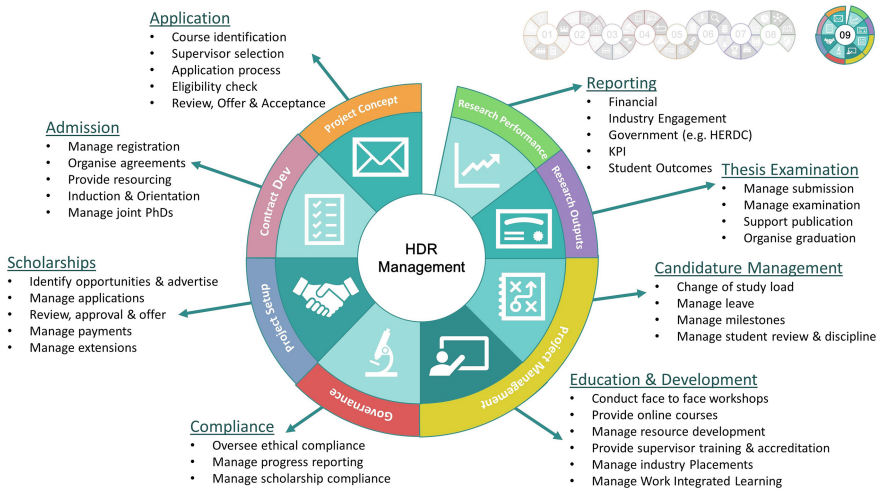


Fig. 11. HDR management

3.9 HDR Management

HDR Management (Fig. 11) refers to the management activities to support higher degree research students. A Higher Degree Research Student follows a similar path to a research project and so there is a large amount of overlap with existing areas already outlined. The color-coded bands surrounding the pie indicate these relationships, but the pie itself focuses on the activities specific to HDR management, which cover the full lifecycle of a student. It should be noted that while research management can be a consideration for undergraduate students, a study by [11] found that less than a quarter of universities have a specific strategy towards this, and of those more than 80% were handled separately from the Deputy Vice Chancellor of Research (or similar role).



Fig. 12. Program management (Color figure online)

3.10 Program Management

Finally, *Program Management* reflects the broader ecosystem that supports research activity at an institute. This is a higher level of proactive strategic management that looks beyond individual projects and support transformative research initiatives [9]. These programs support the creation of many new projects or students. As Fig. 12 shows, there will be similar analogues to many of the previous stages, with these programs going through the process of conception, contractual setup and governance. Similarly, they will create outputs, including those of the research projects they contain and will have reporting requirements to show the effectiveness of the program.

4 Discussion

Despite the growth of research within universities and the proliferation of support staff, the lack of a holistic framework that describes the activities of research management limits the capacity to provide efficient services, properly share learnings and consistently assess the effectiveness of research. In an endeavor to address this gap, this paper addressed the research question: *what are the processes fundamental for research*

management of a university? A preliminary reference model for research management was developed through an ADR effort consisting of 4 cycles. The first cycle aimed at identifying high level activities, the second cycle focused on capturing activities related to research project administration, the third cycle addressed the areas of governance, ethics and research finance with the final cycle confirming the reporting, system and end-user academic dependencies. This ADR effort served the goals of the research well by allowing for constant feedback loops with people well experienced in the research management space.

The findings illustrate the primary components of the reference model. The resulting model is a comprehensive and highly detailed model, which can be used as a point of guidance to identify the processes underlying research management at a higher education or research institute. It takes the form of normative guidelines to guide users to approach research management. It is a multi-disciplinary artefact bridging two communities of researchers, namely from the field of BPM and from the higher education sector. Diverse stakeholders (i.e. research managers, researchers and university leaders) can greatly benefit from the clarity and visibility of research management processes that the resulting reference model provides. It is a useful tool for research process management, investigations and communication [19, 32]. For example, on an anecdotal level, the project manager of RMSUP, who was involved in many of the end-user sessions, commented on the effectiveness of the model to elicit requirements from stakeholders who may not have been used to process-oriented thinking, as it provided a frame of reference for them to consider their daily activities. It also was used to introduce people unfamiliar with research management to the concept, in a simple and easily accessible way.

This research management reference model can be used by a research institute to not just assess what they are presently doing, but what sub-processes of the model they may not be currently addressing. Additionally, it can be used to assess what data is captured about different processes, to ensure that work is visible and can be reported upon. Similarly, it can be used to determine the systems used to support these processes and the adequacy of them. Finally, it can be used to assess service delivery, by examining researchers' experiences and satisfaction with each of the sub-processes. These are just examples of the potential applications of the reference model to a specific research institute context.

While this model has varied applications, we do acknowledge limitations. A reference model by its nature is designed in one specific context, but can be applied to other contexts [3], and as yet this model has not been validated beyond one university. Informal engagement has been conducted and shown promising signs of generalizability, but a rigorous evaluation is required to confirm that the reference model is an accurate representation of research management internationally. Additionally, engagement with research manager's specifically focusing on Higher Degree Research (HDR) was minimal in the current ADR cycles. Hence, more work is needed to confirm the completeness (with the appropriate inclusion of all processes like HDR) of the resulting reference model.

We suggest that the model be tested; re-specified as needed and validated with insights from other universities to consider both its generalizability and its completeness. Future studies should be extended to include other stakeholders (such as research

academics, funding bodies and HDR students) to enable the inclusion of diverse viewpoints. Mechanisms to adapt the model (for different institutional contexts and to address dynamic needs) and to allow the model to evolve with changing industry demands also needs to be investigated. As the model matures, work should be done to develop measures that can be used to assess a research institutes maturity in these different domains. Future research in this field can deliver real benefits for research institutes to provide effective services for research management.

References

1. APQC: APQC Process Classification Framework (PCF). APQC (2018)
2. Barber, K.D., et al.: Business-process modelling and simulation for manufacturing management. *Bus. Process Manag. J.* **9**(4), 527–542 (2003). <https://doi.org/10.1108/14637150310484544>
3. Becker, J., Delfmann, P., Knackstedt, R.: Adaptive reference modeling: integrating configurative and generic adaptation techniques for information models. In: Becker, J., Delfmann, P. (eds.) *Reference Modeling*, pp. 27–58. Physica-Verlag HD, Heidelberg (2007). https://doi.org/10.1007/978-3-7908-1966-3_2
4. Becker, J., Schutte, R.: A reference model for retail enterprise. In: *Reference Modeling for Business Systems Analysis*, pp. 182–205. IGI Global (1)AD. <https://doi.org/10.4018/978-1-59904-054-7.ch009>
5. Beerkens, M.: Facts and fads in academic research management: the effect of management practices on research productivity in Australia. *Res. Policy* **42**(9), 1679–1693 (2013). <https://doi.org/10.1016/j.respol.2013.07.014>
6. Bertolini, M., et al.: Business process re-engineering in healthcare management: a case study. *Bus. Process Manag. J.* **17**(1), 42–66 (2011). <https://doi.org/10.1108/14637151111105571>
7. Carpinetti, L.C.R., et al.: Quality management and improvement: a framework and a business-process reference model. *Bus. Process Manag. J.* **9**(4), 543–554 (2003). <https://doi.org/10.1108/14637150310484553>
8. Davis, R.: British telecom - six level process hierarchy. In: *Process Days Conference, Sydney, 22–24 August 2006* (2006)
9. Demes, K.W., et al.: Catalyzing clusters of research excellence: an institutional case study. *J. Res. Adm.* **L**(1), 108–122 (2019)
10. Department of Education and Training: *Research Income Data (2004–2017)*. <https://docs.education.gov.au/documents/herdc-research-income-time-series>
11. Droegeimer, K.K., et al.: The roles of chief research officers at American research universities: a current profile and challenges for the future. *J. Res. Adm.* **XLVIII**(1), 26–64 (2017)
12. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Process discovery. In: Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A. (eds.) *Fundamentals of Business Process Management*, pp. 155–184. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-33143-5_5
13. Fettke, P., Loos, P.: Perspectives on reference modeling. In: *Reference Modeling for Business Systems Analysis*, pp. 1–21. IGI Global (1)AD. <https://doi.org/10.4018/978-1-59904-054-7.ch001>
14. Greasley, A.: Using process mapping and business process simulation to support a process-based approach to change in a public sector organisation. *Technovation* **26**(1), 95–103 (2006). <https://doi.org/10.1016/J.TECHNOVATION.2004.07.008>

15. Hammer, M.: What is business process management? In: Brocke, J., Rosemann, M. (eds.) *Handbook on Business Process Management 1*, pp. 3–16. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-00416-2_1
16. Hollingsworth, D.: *The Workflow Reference Model. The Workflow Management Coalition Specification*, Hampshire, UK (1995)
17. IBM: Align IT with business goals using the IBM Process Reference Model for IT. IBM Corporation (2007)
18. Kirkland, J., et al.: *International research management: benchmarking programme*. Association of Commonwealth Universities (2006)
19. Lawrence, R.J., Després, C.: Futures of transdisciplinarity. *Futures* **36**(4), 397–405 (2004). <https://doi.org/10.1016/J.FUTURES.2003.10.005>
20. Liberale, A.P., Kovach, J.V.: Reducing the time for IRB reviews: a case study. *J. Res. Adm.* **XLVIII**(2), 37–51 (2018)
21. Mauer, A.: A reference model for savings bank. In: *Advances in Banking Technology and Management*, pp. 232–242. IGI Global (2008). <https://doi.org/10.4018/978-1-59904-675-4.ch014>
22. Petersson, A.M., Lundberg, J.: Applying action design research (ADR) to develop concept generation and selection methods. *Procedia CIRP* **50**, 222–227 (2016). <https://doi.org/10.1016/J.PROCIR.2016.05.024>
23. Du Preez, J.L., et al.: Developing a methodology for online service failure prevention: reporting on an action design research project-in-progress. In: *Australasian Conference on Information Systems (ACIS 2015)*, Adelaide (2015)
24. Rosemann, M.: Using reference models within the enterprise resource planning lifecycle. *Aust. Account. Rev.* **10**(22), 19–30 (2000). <https://doi.org/10.1111/j.1835-2561.2000.tb00067.x>
25. Rosemann, M., van der Aalst, W.M.P.: A configurable reference modelling language. *Inf. Syst.* **32**(1), 1–23 (2007). <https://doi.org/10.1016/J.IS.2005.05.003>
26. Saadé, R.G., Otrakji, C.A.: First impressions last a lifetime: effect of interface type on disorientation and cognitive load. *Comput. Hum. Behav.* **23**(1), 525–535 (2007). <https://doi.org/10.1016/j.chb.2004.10.035>
27. Schützenmeister, F.: *University Research Management: An Exploratory Literature Review*. Institute of European Studies, UC Berkeley (2010)
28. Sein, M.K., et al.: Action design research. *MIS Q.* **35**(1), 37–56 (2011). <https://doi.org/10.2307/23043488>
29. Squilla, B., et al.: Research shared services: a case study in implementation. *J. Res. Adm.* **XLVIII**(1), 86–99 (2017)
30. Wedekind, G.K., Philbin, S.P.: Research and grant management: the role of the project management office (PMO) in a European research consortium context. *J. Res. Adm.* **XLIX**(1), 43–62 (2018)
31. Wei Khong, K., Richardson, S.: Business process re-engineering in Malaysian banks and finance companies. *Manag. Serv. Qual. Int. J.* **13**(1), 54–71 (2003). <https://doi.org/10.1108/09604520310456717>
32. Wickson, F., et al.: Transdisciplinary research: characteristics, quandaries and quality. *Futures* **38**(9), 1046–1059 (2006). <https://doi.org/10.1016/J.FUTURES.2006.02.011>
33. Zimmermann, H.: OSI reference model-the ISO model of architecture for open systems interconnection. *IEEE Trans. Commun.* **28**(4), 425–432 (1980). <https://doi.org/10.1109/TCOM.1980.1094702>



What the Hack? – Towards a Taxonomy of Hackathons

Christoph Kollwitz^(✉)  and Barbara Dinter

Chemnitz University of Technology, 09126 Chemnitz, Germany
{christoph.kollwitz,
barbara.dinter}@wirtschaft.tu-chemnitz.de

Abstract. In order to master the digital transformation and to survive in global competition, companies face the challenge of improving transformation processes, such as innovation processes. However, the design of these processes poses a challenge, as the related knowledge is still largely in its infancy. A popular trend since the mid-2000s are collaborative development events, so-called hackathons, where people with different professional backgrounds work collaboratively on development projects for a defined period. While hackathons are a widespread phenomenon in practice and many field reports and individual observations exist, there is still a lack of holistic and structured representations of the new phenomenon in literature. The paper at hand aims to develop a taxonomy of hackathons in order to illustrate their nature and underlying characteristics. For this purpose, a systematic literature review is combined with existing taxonomies or taxonomy-like artifacts (e.g. morphological boxes, typologies) from similar research areas in an iterative taxonomy development process. The results contribute to an improved understanding of the phenomenon hackathon and allow the more effective use of hackathons as a new tool in organizational innovation processes. Furthermore, the taxonomy provides guidance on how to apply hackathons for organizational innovation processes.

Keywords: Hackathon · Taxonomy · Digital innovation · Open innovation · Innovation process

1 Introduction

One of the central tasks of business process management (BPM) is to deal with changing environmental conditions [1]. In recent years, such a transformation appears in trends like shorter product life cycles and increasingly heterogeneous customer requirements. In this context, business processes in the field of innovation management are opening up and changing rapidly, which is addressed by BPM, e.g. by new information sources. Companies invest significant sums in R&D to master the challenges of digital transformation and to survive in the global economy. Traditionally, they have innovated almost solely to prevent leaking knowledge, technologies and process know how to unauthorized third parties or competitors [2]. However, companies merely focusing on internal competencies and resources fall behind in a

hardening competition. Therefore, since the beginning of the 21st century a paradigm shift towards opening innovation changes the way how innovation processes are designed and how external knowledge contributes to the development of new products and services. As a bottom line of this so-called open innovation (OI) Chesbrough [3 p. XXIV] states that companies “can and should use external ideas as well as internal ideas, and internal and external paths to market” and over time various approaches for its operationalization have been developed. On the one hand, the advent of Web 2.0 technologies has enabled OI tools like online communities and product platforms for OI, while, on the other hand, the involvement of lead users and other stakeholders in (offline) innovation workshops was highlighted. All these approaches have in common that they understand innovation as the result of collaborative processes in (interdisciplinary) teams rather than as the work of individuals [4]. In addition to the paradigm of OI, digitalization has radically changed the nature of innovation. Digital innovation is in particular shaped by emerging (information) technology and the ubiquitous availability of (digital) data, enabling companies to provide “data-enriched offerings” to their customers [5].

Facing the trends towards openness and digitalization, companies need to find new ways to manage innovation processes in the digital age. More precisely, it poses a challenge for business process management to manage creative processes within and outside organizational boundaries [6], especially when organizations have little or no knowledge about their innovation partners [7]. In this paper, we will therefore investigate a phenomenon in which these two trends are manifested - so-called hackathons. Hackathons can be briefly described as events in which participants collaborate intensively on completing projects over a defined period of time [8]. Such projects focus on an IT-related topic, e.g. developing hardware and/or software, analyzing data sets or identifying IT-security issues. Hackathons can help companies, especially in the early stages of innovation, to generate new ideas, develop concepts or test solutions [9]. Although hackathons are becoming more and more popular in practice, related research is still in the fledgling stage. Thus, existing literature often consists of experience reports, white papers or reflections on specific application domains such as healthcare or smart cities. Companies need help for answering the question of how they can support the application of OI tools [10]. Therefore, they need a holistic view on phenomena like hackathons, which is currently not provided by the literature [11]. This leads us to our research question: How can the complex phenomenon of hackathons be systematically conceptualized in order to enable organizations to utilize them for their innovation processes? In order to answer the research question, we build a taxonomy of hackathons, applying the method for taxonomy development according to Nickerson et al. [12]. A taxonomy is a frequently hierarchical and evolutionary classification of empirical entities that can be used by researchers to organize research fields or entities [13, 14] and is frequently used in the Information Systems (IS) research domain.

The paper is organized as follows: In Sect. 2, we discuss hackathons as a manifestation of OI in more detail and clarify in particular which role they can play in OI processes. Afterwards, our research method is explained in Sect. 3, followed by the presentation of the resulting taxonomy in Sect. 4. The results are discussed in Sect. 5 before we draw a conclusion in Sect. 6.

2 A Process-Centric Perspective on Open Innovation and Hackathons

OI represents a challenge for the management of innovation processes, as it is linked to a shift from well-defined and structured processes to more interactive and agile processes [15]. Uncertainties regarding the results, process structures and required resources prevail within creativity-intensive process stages [16], which are particularly important at the beginning of the innovation process. Many different approaches to conceptualize innovation at the process level can be found in the literature (for an overview cf. [17]). In this paper, we follow Hansen and Birkinshaw [18], who make a simple distinction between: (1) idea generation, (2) idea conversion and (3) idea diffusion. Thereby, the innovation process is often described as a “funnel”, since at the early stages there are many opportunities for innovation, of which only a few are concretized and realized at the later stages [19]. Based on this basic process model, innovation processes can be opened for different purposes for external knowledge. Gassmann and Enkel [20] distinguish three types of OI: (1) outside-in OI describes the sourcing and acquisition of expertise and ideas for the innovation process, while (2) inside-out OI focuses on the exploitation of ideas and the results of innovation processes. If outside-in and inside-out processes are combined, it is referred to as (3) coupled OI. The different types are in turn associated with different tasks. The outside-in perspective of OI comprises the identification, procurement and integration of innovations as well as the interaction with external partners [21]. The inside-out process, which receives less attention compared to the first perspective in the literature, can in turn be subdivided into the search for technology users and the commercialization phase [10]. In order to operationalize these processes, many different means are discussed in research. Battistella et al. [22] identify a total of 23 practices which were used by companies to implement out-side (e.g. crowdsourcing), inside-out (e.g. out-licensing of intellectual property) or coupled OI (e.g. joint ventures). With a focus on Web 2.0 technologies, Möslein and Bansemir [23] distinguish between innovation contests, innovation markets, innovation communities, innovation toolkits and innovation technologies as OI tool categories. Additionally, various authors emphasize the involvement of stakeholders such as customers/users through innovation workshop [24].

In this context, hackathons are considered as an OI tool, which can hardly be classified into existing tool categories since it combines elements of different tool classes. The term hackathon is a portmanteau from “hack” and “marathon” and was first coined at an OpenBSD developer event in Calgary in 1999. There is a variety of synonyms or similar terms such as hack day or hackfest, however, hackathon is by far the most popular term. Furthermore, hackathons that focus on the collection, analysis and/or visualization of data are also referred to as “datathons”. Hackathons are rooted in the open source movement and have often been associated with civic engagement and open data [25]. Thus, there are initiatives from government agencies, which aim to increase the participation of citizens and to foster government transparency [26]. Other hosts of hackathons are organizations from the non-profit sector, such as educational institutions or research institutes or NGOs, which address social problems like environmental protection or poverty reduction [27]. However, since large digital players

such as Google and Facebook have regularly conducted hackathons in which also external developers have participated [28], they have become a more and more interesting topic for companies of all industries who aim to complement traditional organizational innovation processes. Although there is a high variance in the activities and routines that take place during hackathons, three phases can be roughly distinguished [11]. In the (1) pre-hackathon phase, the focus lies on planning and design tasks. Besides, team building and initial idea generation can also start in this phase. Subsequently, the (2) hackathon phase includes the execution of the event, i.e. the collaborative work of the participants. In the (3) post-hackathon phase, the decision must be made whether and how the results of the hackathon should be followed up (by the host organization or the participants) or should be dropped. Hackathons aim to harness external knowledge for organizations, which corresponds with the outside-in type of OI. The knowledge is integrated mainly in the phases of idea generation and idea conversion and thus serves the organization primarily for knowledge exploration. However, hackathons can also be used in the later innovation process and enhance the diffusion of innovations [9].

3 Research Approach

3.1 Taxonomy Development

As mentioned in the introduction, our research is motivated by the emerging phenomenon of hackathons - in particular, by the discrepancy between the large number of anecdotal observations and field reports on the one hand and the lack of a holistic and comprehensive view on hackathons on the other hand. We would like to emphasize that there are many (scientific) publications on hackathons, which are also shown by the results of our literature search. However, these describe single instances or potential applications in specific areas and do not provide a consolidated and comprehensive view of the phenomenon. Taxonomies are particularly suitable for structuring and classifying complex research topics and therefore play an important role in various areas of IS research [29]. Especially with regard to emerging digital technologies or the management of novel (open) processes, taxonomies can help to consolidate knowledge and make it usable for practitioners as well as for researchers [30, 31]. For the development of the taxonomy we apply the established method of Nickerson et al. [12], which is guided by best practices from other research disciplines as well as the principles of Design Science Research [e.g. 32].

The first step in taxonomy development is the definition of a (1) meta characteristic, which is intended to support researchers to identify meaningful categories and dimensions that relate to the purpose of the taxonomy. In our case, the taxonomy is supposed to support the integration of hackathons into organizational innovation processes. Therefore, we examine hackathons from an organizational perspective and focus on dimensions and characteristics that cover the broad spectrum of design decisions associated with the design and execution of hackathons. The second step of Nickerson's method involves the determination of objective and subjective (2) ending conditions. Objective ending conditions are achieved when all objects of a population

or a statistical sample have been analyzed and the result meets the requirements of a taxonomy (e.g. no redundancies/duplications, mutual exclusivity) [12]. Subjective ending conditions affect the researchers assessment of the resulting taxonomy and describe to what extent the taxonomy is considered to be concise, robust, comprehensive, extendible and explanatory [12]. In the third step, the taxonomy is created. Nickerson suggests an iterative process, which is performed until the previously defined ending conditions are met. In general, this process can be inductive (empirical-to-conceptual) or deductive (conceptual-to-empirical), whereby our approach focuses on the former one. Thus, our taxonomy is mainly based on a systematic literature review, which is described in more detail in Subsect. 3.2. The articles were manually screened and analyzed according to the meta characteristic using open, axial and selective coding [cf. 33]. Three iterations were carried out until the ending conditions were reached:

- Open coding: In the first iteration (empirical-to-conceptual), we examined the articles of the literature base according to the meta characteristic for statements on design, execution and objectives of hackathons and grouped by similar characteristics.
- Axial coding: In the second iteration (conceptual-to-empirical), existing taxonomies or taxonomy-like artifacts (e.g. morphological boxes, typologies) from similar research areas were included [e.g. 34–36] and compared with the attributes identified in the first iteration. In case of similarities, the dimensions and characteristics (partially modified) were included in the taxonomy.
- Selective coding: In the third iteration (empirical-to-conceptual), the characteristics identified in the literature, which could not be assigned to any dimension in the second iteration, were summarized in new dimensions and integrated into the taxonomy. Furthermore, the complete taxonomy was refined based on the objective and subjective ending conditions. In addition, the characteristics and their attributes were checked for correlations and dependencies (cf. Subsect. 4.1).

3.2 Literature Review

In recent years, a growing number of scientific publications in books, journals as well as conference proceedings reflect the increasing complexity of research. In this context, literature reviews can help to consolidate knowledge from different research areas and to gain insights into specific problem areas [37]. We decided to use a systematic literature review for the development of a hackathon taxonomy mainly for two reasons. First, such a meta-analysis allows us to access and investigate a large number of hackathon reports covering a broad spectrum of applications. Second, we can apply established methods for the literature review which facilitate the systematic development of our hackathon taxonomy [38, 39]. Vom Brocke et al. [38] propose a five phase model for literature reviews in IS.

The first step is to define the scope, which can be described using Cooper’s literature review taxonomy [37]. Table 1 shows the scope of our literature review. We focus on hackathons described in the literature, whereby analyzing their (A) application and design and only marginally considering the results, the methods and underlying theories presented in the papers. Our goal is to (B) integrate existing knowledge and make it usable, in other words, we aim at a (C) conceptualization of the hackathon phenomenon. Our perspective is (D) neutral to avoid distorting the results of the review. As already mentioned in the introduction, our target (E) audience consists of practitioners on the one hand and general researchers on the other. Our aim is to consider an exhaustive literature basis when developing the taxonomy, whereby we prove the individual dimensions and characteristics of the taxonomy based on selected articles. Therefore, we assign our review in category (F) coverage as exhaustive and selective.

Table 1. Taxonomy of literature reviews [37, 38]

Characteristics	Categories			
(A) Focus	research outcome	research method	theories	applications
(B) Goal	integration		criticism	central issues
(C) Organisation	historical	conceptual		methodological
(D) Perspective	neutral representation		espousal of position	
(E) Audience	specialized scholars	general scholars	practitioners / politicians	general public
(F) Coverage	exhaustive	exhaustive and selective	representative	Central / pivotal

The second step includes a broad conceptualization of the research subject, in our case hackathons, for which we would like to refer to Sect. 2. The third step consists of the literature search, which includes the selection of databases and keywords as well as the forward and backward search for literature [38]. Regarding the keywords, we searched for the terms, which are depicted in Table 2. We have deliberately excluded related terms such as “jam”, since they are often used in other contexts. Following the recommendations of vom Brocke et al. [38], we first searched the top journals of IS discipline (Senior Scholars’ Basket of Journals) for relevant publications. We did not find any relevant hits, which corresponds to our expectation that hackathons have not yet found their way into the most renowned journals. Then we expanded our search to the scientific databases AIS Electronic Library (AISeL), IEEE Xplore Digital Library (IEEE) as well as the citation database “Web of Science Core Selection” (WoS), where we searched in “titles”, “abstracts” and “keywords” for the mentioned terms.

Table 2. Database search results by keywords

		Keywords				
		hackathon / hack-a-ton	hack day	datathon	hackfest	codefest
Data- bases	AISeL	4	1	2	0	0
	IEEE	63	2	0	1	0
	WoS	147	6	6	1	1

We received 234 hits in total. After eliminating duplicates, removing irrelevant papers (by checking titles and abstracts) and a forward and backward search, we ended up with 189 publications, which we included in the literature analysis. These publications are mostly conferences proceedings or practice-oriented journal articles. Some of the papers describe hackathons that have taken place in the context of conferences, teaching in higher education or other events. The third step of a literature review according to vom Brocke et al. [38] contains the analysis and synthesis of literature. For this purpose, we followed the approach proposed by Nickerson et al. [12] as explained in Subsect. 3.1. The final step of the literature review framework is the development of a research agenda. Since our main interest is the taxonomy development for hackathons and we do not primarily aim to identify research gaps, we have decided not to derive a research agenda. However, in Sect. 6 we highlight potentials for further research.

4 A Taxonomy of Hackathons

4.1 Overview of the Taxonomy

In the course of the analysis, it became apparent that the dimensions could be assigned to two categories. Strategic design decisions (SDD) tend to be abstract in character and are derived from the overall goals and business model of organizations, while operational design decisions (ODD) mainly determine the workflow and processes that take place during a hackathon. With regard to the benefits of the taxonomy for organizations, these categories serve different purposes. The SDD support the organization in identifying useful application scenarios for hackathons. They outline the options in terms of which challenges could be addressed for which purposes. The ODD can in turn support organizations in designing specific settings that fit their organizational environment. For example, different dimensions can be adjusted according to the financial, human and spatial resources of an organization. Some SDD dimensions partly have an influence on ODD or determine them. Table 3 gives an overview of our taxonomy of hackathons, while we present the dimensions and characteristics in the Subsects. 4.2 and 4.3.

Table 3. Taxonomy of hackathons

Dimension		Characteristics			
SDD	OI integration	idea generation	idea conversion		idea diffusion
	Challenge design	technology-centric (API, software, hardware)	topic-centric (social issue, business problem)		data-centric (analysis, visualization, gathering)
	Solution space	open	semi-structured		structured
	Value proposition	focus on challenge output		focus on human interaction	
ODD	Duration	short (<24 h)	medium (>24h – 72h)		long (>72h)
	Degree of elaboration	ideas and broad concepts	conceptual solutions	functional solutions	finished products / services
	Venue	physical	virtual		combined
	Incentives	competition		collaboration	
	Target audience	domain experts	(semi-) professionals		general public
	Resources	provided	partially provided		not provided

4.2 Strategic Design Decisions

OI Integration. As already mentioned in Sect. 2, hackathons can be considered as OI tools that are typically applied to the outside-in process. In general, they can be applied in all phases of the innovation process [18], with different objectives being pursued. In the *idea generation* phase, hackathons aim to generate initial innovation impulses from the outside. With regard to the *idea conversion*, promising ideas are to be selected for further development in cooperation with external developers. The phase *idea diffusion* involves testing and presenting products and services that have already been available on the market [9]. For example, software can be provided in order to deduce room for improvement or possible applications from hackathon results. This dimension is of particular importance since it constitutes the interface between OI process management and hackathon design.

Challenge Design. A common characteristic of all investigated hackathons is that they are associated with the handling of a task or the solution of a problem. This dimension represents the focus of the hackathon’s task or challenge. The primal form of hackathons, originated in the open source movement, was strongly oriented towards specific *technology* issues, including software, hardware or APIs related tasks [11]. Hackathon

challenges can also pursue social or business *topics*, which does not mean that technology does not play a role, rather their purpose is focused on solving a problem by using technologies [27]. In addition to these two characteristics, which are similarly proposed by Briscoe and Mulligan [25], we add *data-centric* as a third characteristic. Thus, the general trend towards “big data” means that challenges aim at generating value from data without having a dedicated technology or a specific business case in mind [40]. Such tasks focus on the processing, analysis or visualization of data sets [41], and in some cases on the collection or generation of data [42].

Solution Space. This dimension refers to specifications made with regard to the execution of the hackathon. We distinguish between *open*, *semi-structured* and *structured* settings. Open settings are characterized by wide-ranging challenges that leave plenty room for interpretation and own ideas. Requirements and restrictions that could potentially limit creativity are reduced to a minimum. The SPIE Software Hack Day 2014 [43] offers a vivid example of such an open solution space. Participants were invited to “collaborate on innovative solutions to problems of their choice” [43]. The format took place without prior registration, a fixed schedule or formal presentations. Semi-structured settings on the one hand provide certain specifications that limit the solution space, but on the other hand leave room for individual approaches [44]. Either the procedure can be limited by the specification of e.g. technologies, data sets or methods that have to be used [45]; or the expected results are specified by technical and/or functional requirements [46]. Structured settings in turn place strict demands on the procedure and the results, which severely limits the solution space. For example, the JUCE Machine Learning Hackathon [47] was an event in which the technology to be used (a C++ framework focusing on audio applications) as well as the type of solution (application of machine learning) were specified.

Value Proposition. This dimension takes into account that hackathons are not autotelic, but are organized for specific purposes. In reality, there are overlaps, as organizations are likely to pursue different objectives simultaneously. This is contradictory to the principle that the characteristics of a taxonomy should be mutually exclusive [12]. However, we consider this dimension to be important, thus we assume that organizations, even if they pursue different goals, associate a primary value proposition with a hackathon. In our analysis, two primary value propositions with different focuses emerged. On the one hand, value propositions with a *focus on challenge output* aim to harness the results of the participative work in the hackathon. Usually this involves results developed in the hackathon such as ideas, models, prototypes or data visualizations as well as extension, improvement or evaluation of existing entities (e.g. extension of software functionalities or detection of security breaches) [25]. On the other hand, value propositions with a *focus on human interaction* aim to generate benefits with respect to the participants. This includes educational aspects as well as the recruitment of new employees [41]. Furthermore, hackathons can be utilized as a communication platform for stakeholders and marketing purposes [11]. In any case, it is important for organizations to be aware of their own expectations and to establish measures for making the results connectable in their own organization.

4.3 Operational Design Decisions

Duration. Hackathons are events that take place over a short period of time, whereby the concrete timing varies greatly in practice. On one side of the continuum, there are hackathons with *short* duration that last only a few hours as a one-day event, which is particularly likely when they are part of other events (e.g. scientific conferences). Lau and Lei [48], who describe a 30-min hackathon at the “International Microwave Symposium 2017”, provide a demonstrative but extreme example of this characteristic. The vast majority of the hackathons discussed in the literature lasted between 24 and 72 h, which we refer to as the *medium* period of time [9]. On the other side of the continuum, there exist *long* duration hackathons that can last from four days up to several weeks [49]. Hackathons with long duration are usually not continuous but consist of multiple events (e.g. kick-off and award ceremony) linked by an intervening development period.

Venue. Hackathons not only deal with technology-related topics. Information technology can also act as a medium for communication and cooperation during the events. While “classic” hackathons take place at *physical* locations [44], there are also formats that are completely organized *virtually* via online platforms or social networks (e.g. Kaggle) and therefore do not require physical presence [42]. Additionally there is the possibility to *combine* physical and virtual venues [49]. Concerning the physical venues, the analyzed articles describe frequently the importance of open and innovation-friendly spaces equipped with tools for collaboration and ideation (e.g. flipcharts or brown paper). Choosing a physical venue also means limiting capacity, while virtual venues allow a literally unlimited number of participants.

Degree of Elaboration. Hackathons are aimed at dealing with technology related issues, but differ greatly in terms of the intended results. We have decided to cluster the different characteristics according to the degree of elaboration, as there is an unlimited variety of resulting artifacts in practice. Artifacts with a relatively low degree of elaboration require only a basic understanding of technologies and operate at a high level of abstraction. The focus is on creativity and the development of *ideas and broad concepts* [50]. A higher degree of elaboration requires a further development of ideas and to *conceptual solutions*. These conceptual prototypes are usually demonstrative paper-based or computer-aided mock-ups that represent a concept resulting from a hackathons, but do not contain any functionalities [51]. The next higher degree of elaboration is obtained when functionalities of the solution are also implemented. Such *functional solutions* include core functions of an e.g. prototype and thus demonstrate the general feasibility of a concept (proof of concept) [8]. The highest degree of elaboration is reached when the hackathon results in *finished products/services*, which are at least mature enough to be launched (minimal viable product) [52]. As the degree of elaboration increases, the demands of the technical and professional skills of the participants usually also increase, while creativity and the ability of abstraction become less important.

Incentives. In general, hackathons are team events, whereby the type of team composition (e.g. before or during the event) and the team sizes can vary. Although

hackathons generally emphasize the value of cooperation, they can be designed as *competitions* in which participants compete among each other. Based on a jury decision, audience vote or self-assessment, the winners usually receive a prize which is intended to increase the extrinsic motivation of the participants [9, 53]. The alternative concept relies entirely on *collaboration* rather than any competition between participants [54].

Target Audience. Although hackathons are traditionally open events, there could be various restrictions concerning the participation. We identified various types of hard restrictions in the literature. For example, in-house hackathons which can only be visited by employees of an organization or hackathons that target socio-demographic characteristics of the participants, such as age, gender or profession [11]. Furthermore, tasks can be chosen in a way that only *domain experts* are able to participate, e.g. physicians [41] or architects [50]. In addition to these hard restrictions, there are soft criteria that are frequently based on the self-assessment of participants. Such constraints often aim to acquire participants with expertise in specific areas such as marketing, programming or data analysis [55]. We refer to this characteristic as *(semi-) professionals*. The last characteristic refers to hackathons, which have no restriction of participation apart from a basic interest in the topic [51], thus, they target the *general public*.

Resources. The last dimension differentiates whether resources are *provided* to the hackathon participants or *not*. Likewise, only some resources can be made available, which we characterize as *partially provided*. In our context, provided resources can be considered as an input, which is made available to the participants before or during the hackathon. The provided resources may be hardware, software or data sets as well as existing ideas, concepts or prototypes, which should be evolved [56]. Furthermore, human resources such as mentors or experts from practice can serve as an input for the participants [57]. Depending on the setting, the usage of resources can be voluntary or mandatory (cf. solution space). The question of whether or not resources will be provided may be related to single SDD dimensions. For example, existing ideas or concepts must be available as input for the participants if a hackathon aim to idea conversion or diffusion.

5 Discussion

OI has been around for several years now and scholars from different professions had already discussed many tools, especially in the context of web 2.0 and social software [26]. Hackathons combine elements of such OI tools with elements from the areas of open source and agile software development [49]. Thus, the dimensions and characteristics of our taxonomy of hackathons is not disconnected from other OI tools and practices, but features several similarities. Hackathons have event character and resemble innovations contests in the dimensions solution space, duration, degree of elaboration, venue and target audience [34]. Furthermore, the dimensions OI integration and solution space correspond to the typology of the customer co-creation by Piller et al. [36]. However, the taxonomy contains dimensions that cannot be found in other

classifications and dimensions that have completely different characteristics. For instance, challenge design describes the IT-related aspects of hackathons, while incentives show that hackathons can be both competitive and collaborative. Our taxonomy also shows that hackathons are very diverse in their practical manifestation, which contradicts several restrictive definition approaches from literature on hackathons. Hackathons are frequently characterized as competitive, short-term events in which software is developed [e.g. 9, 25]. Our analysis showed that hackathons could be considered as OI tools with a wide range of applications, rather than being limited to competitions, short periods or software development projects.

Since hackathons must be incorporated into BPM, our taxonomy highlights many dimensions, such as resources or process participants to be considered [1]. Our approach transfers the established method of taxonomy development to the immature field of hackathons and thus, contribute to the knowledge base by exaptation [58]. Since organizations need assistance in managing innovation processes in the age of digitalization [7], our taxonomy of hackathons is intended to enable organizations to utilize hackathons for successful innovating. In this context, where several uncertainties exist a central challenge poses the management of creativity-intense processes [6]. Our taxonomy can support organizations in planning creative processes in hackathons by constraining them, which in turn helps to manage uncertainties regarding (1) results, (2) processes and (3) resources [59]. Concerning the uncertainties of the (1) results, organizations can make detailed specifications regarding the solution space as well as the degree of elaboration in order to channel the creativity of the participants in a desired direction. In addition, uncertainties are generally more pronounced in the early stages of the innovation process than in the later ones, which is reflected by the dimension OI integration. The dimensions challenge design and the solution space have an influence on the (2) process uncertainties, which determine the form and substance of a hackathon. As mentioned above in Subsect. 4.1, all ODD dimensions also have a direct influence on the hackathon processes and their degree of uncertainty. For example, hackathons that have a short duration and take place at a physical location might be easier to predetermine beforehand than those that take place over long periods and include both physical and virtual forms of collaboration. In terms of (3) resource uncertainties, organizations can regulate e.g. the duration and, can provide resources for the participants. In addition, the availability of intangible resources such as expertise or skills can be influenced by the appropriate selection of a target audience.

Another aspect to be discussed is the categorization of SDD and ODD in our taxonomy of hackathons. We consider this a first step to gain a better understanding of how the strategic goals of organizations are linked to the design of hackathon processes. The taxonomy shows application scenarios (SDD) and operational design options (ODD), which can lead to a better strategic alignment, which is considered as a core element of BPM [60]. However, in this article our research approach focused on the design of the taxonomy rather than on the investigation of linkages between strategic and operational elements. In Sect. 6 we will discuss how we intend to achieve this in the future.

6 Conclusion

In this paper, we examine hackathons as a novel phenomenon at the crossroads of digital innovation and OI. We used the method of Nickerson et al. [12] combined with a systematic literature review to develop a taxonomy of hackathon. The result contributes to a better understanding of the opportunities and characteristics of hackathons and is therefore a first step towards a better integration of hackathons into organizational innovation processes. Our results not only give directions which kind of innovation challenges can be addressed, but also provide companies with initial recommendations on how to proceed when using this new resource in the BPM context. From a research perspective, the results contribute by expanding the knowledge base in the spectrum of OI tools and practices as well as in the field of collaborative work in the digital age.

The taxonomy can be considered as generally valid since it was derived from a comprehensive number of primary sources. However, our research is still in an early stage and some limitations exist. The taxonomy is currently based only on findings we have derived from a retrospective review of the literature. Thus, the significance of our results is limited due to the restrictions in the review strategy (restriction to certain databases and keywords). Although we have figured out which dimensions and objectives are discussed in the literature, we need further evidence to show that those aspects are actually relevant from a practical point of view. We aim to compensate for this shortcoming by conducting case studies and in-depth interviews examining the roles of the different actors in real-world hackathons in more detail. In this direction, our next step for further research is to study the relationships between individual dimensions or characteristics. In particular, we want to show the interplay of SDD and ODD dimensions in more detail and further develop the taxonomy into an ontology.

References

1. Smith, H., Fingar, P.: *Business Process Management: The Third Wave*. Meghan-Kiffer Press, Tampa (2003)
2. Chesbrough, H.W., Appleyard, M.M.: Open innovation and strategy. *Calif. Manag. Rev.* **50**, 57–76 (2007)
3. Chesbrough, H.W.: *Open Innovation: The New Imperative for Creating and Profiting from Technology* (2003)
4. Steen, M., Manschot, M., De Koning, N.: Benefits of co-design in service design projects. *Int. J. Des.* **5**, 53–60 (2011)
5. Davenport, T.H.: Analytics 3.0. *Harv. Bus. Rev.* **12**, 65–72 (2013)
6. Schmiedel, T., vom Brocke, J.: Business process management: potentials and challenges of driving innovation. In: vom Brocke, J., Schmiedel, T. (eds.) *BPM - Driving Innovation in a Digital World*. MP, pp. 3–15. Springer, Cham (2015)
7. Nambisan, S., Lyytinen, K., Majchrzak, A., Song, M.: Digital innovation management: reinventing innovation management research in a digital world. *MIS Q.* **41**, 223–238 (2017)
8. Pe-Than, E.P.P., Nolte, A., Filippova, A., Bird, C., Scallen, S., Herbsleb, J.D.: Designing corporate hackathons with a purpose: the future of software development. *IEEE Softw.* **36**, 15–22 (2019)

9. Rosell, B., Kumar, S., Shepherd, J.: Unleashing innovation through internal hackathons. In: *Digest of Technical Papers - InnoTek 2014: 2014 IEEE Innovations in Technology Conference* (2014)
10. Aloini, D., Lazzarotti, V., Manzini, R., Pellegrini, L.: Implementing open innovation: technological, organizational and managerial tools. *Bus. Process Manag. J.* **23**, 1086–1093 (2017)
11. Komssi, M., Pichlis, D., Raatikainen, M., Kindstrom, K., Jarvinen, J.: What are hackathons for? *IEEE Softw.* **32**, 60–67 (2015)
12. Nickerson, R.C., Varshney, U., Muntermann, J.: A method for taxonomy development and its application in information systems. *Eur. J. Inf. Syst.* **22**, 336–359 (2013)
13. Bailey, K.D.: *Typologies and Taxonomies an Introduction to Classification Techniques*. Sage Publications, Thousand Oaks (1994)
14. Glass, R.L., Vessey, I.: Contemporary application-domain taxonomies. *IEEE Softw.* **12**, 63–76 (1995)
15. Gassmann, O., Enkel, E., Chesbrough, H.: The future of open innovation. *R D Manag.* **40**, 213–221 (2010)
16. Seidel, S.: Toward a theory of managing creativity-intensive processes: a creative industries study. *Inf. Syst. E-bus. Manag.* **9**, 407–446 (2011)
17. Eveleens, C.: Innovation management; a literature review of innovation process models and their implications (2010). <http://www.academia.edu/download/33190893/Innovation-management-literature-review-.pdf>
18. Hansen, M.T., Birkinshaw, J.: The innovation value chain. *Harv. Bus. Rev.* **85** (2007)
19. Lazzarotti, V., Manzini, R.: Different modes of open innovation: a theoretical framework and an empirical study. *Int. J. Innov. Manag.* **13**, 615–636 (2009)
20. Gassmann, O., Enkel, E.: Towards a theory of open innovation: three core process archetypes. In: *Proceedings of the R&D Management Conference (RADMA)*, Lisbon, Portugal (2004)
21. West, J., Bogers, M.: Leveraging external sources of innovation: a review of research on open innovation. *J. Prod. Innov. Manag.* **31**, 814–831 (2014)
22. Battistella, C., De Toni, A.F., Pessot, E.: Practising open innovation: a framework of reference. *Bus. Process Manag. J.* **23**, 1311–1336 (2017)
23. Mölslein, K.M., Bansemir, B.: Strategic open innovation: basics, actors, tools and tensions. In: Michael, H., Pfeffermann, N. (eds.) *Strategies and Communications for Innovations: An Integrative Management View for Companies and Networks*, pp. 11–23. Springer, Berlin (2011)
24. Edvardsson, B., Kristensson, P., Magnusson, P., Sundström, E.: Customer integration within service development - a review of methods and an analysis of insitu and exsitu contributions. *Technovation.* **32**, 419–429 (2012)
25. Briscoe, G., Mulligan, C.: *Digital innovation: the hackathon phenomenon* (2014)
26. Robinson, P.J., Johnson, P.A.: Civic hackathons: new terrain for local government-citizen interaction? *Urban Plan.* **1**, 65–74 (2016)
27. Ciaghi, A., et al.: Hacking for Southern Africa: collaborative development of hyperlocal services for marginalised communities. In: *2016 IST-Africa Conference, IST-Africa 2016* (2016)
28. Choi, M.: Organizing open digital innovation: evidence from hackathons. In: *International Conference on Information Systems (ICIS)*, Dublin, Ireland, pp. 1–11 (2016)
29. Nickerson, R.C., Varshney, U., Muntermann, J., Isaac, H.: Taxonomy development in information systems: developing a taxonomy of mobile applications. In: *Proceedings of the European Conference on Information Systems (ECIS)*, Verona, Italy, p. 388 (2009)

30. Geiger, D., Seedorf, S., Schulze, T., Nickerson, R., Schader, M.: Managing the crowd: towards a taxonomy of crowdsourcing processes. In: *AMCIS 2011 Proceedings* (2011)
31. Berger, S., Denner, M.-S., Röglinger, M.: The nature of digital technologies – development of a multi-layer taxonomy. In: *26th European Conference on Information Systems (ECIS 2018)*, pp. 1–18 (2018)
32. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* **28**, 75–105 (2004)
33. Wolfswinkel, J.F., Furtmueller, E., Wilderom, C.P.M.: Using grounded theory as a method for rigorously reviewing literature. *Eur. J. Inf. Syst.* **22**, 45–55 (2013)
34. Bullinger, A.C., Möslein, K.: Innovation contests – where are we? In: *AMCIS 2010 Proceedings*, pp. 1–9 (2010)
35. Dinter, B., Kollwitz, C., Fritzsche, A.: Teaching data driven innovation – facing a challenge for higher education. In: *AMCIS 2017 - America's Conference on Information Systems: A Tradition of Innovation* (2017)
36. Piller, F.T., Ihl, C., Vossen, A.: A typology of customer co-creation in the innovation process. In: *New Forms of Collaborative Innovation and Production on the Internet: An Interdisciplinary Perspective*. Universitätsverlag Göttingen, Göttingen (2011)
37. Cooper, H.M.: Organizing knowledge syntheses: a taxonomy of literature reviews. *Knowl. Soc.* **1**, 104–126 (1988)
38. vom Brocke, J., Simons, A., Niehaves, B., Reimer, K., Plattfaut, R., Clevén, A.: Reconstructing the giant: on the importance of rigour in documenting the literature search process. In: *Proceedings of the European Conference on Information Systems (EICS 2009)*, Verona, p. 161 (2009)
39. Webster, J., Watson, R.T.: Analyzing the past to prepare for the future: writing a literature review. *MIS Q.* **26**, 13–23 (2002)
40. Vanauer, M., Bohle, C., Hellingrath, B.: Guiding the introduction of big data in organizations: a methodology with business- and data-driven ideation and enterprise architecture management-based implementation. In: *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS)*, Kauai, pp. 908–917 (2015)
41. Wyngaard, J., Lynch, H., Nabrzyski, J., Pope, A., Jha, S.: Hacking at the divide between polar science and HPC: using hackathons as training tools. In: *Proceedings - 2017 IEEE 31st International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2017*, pp. 352–359 (2017)
42. Alba, M., Avalos, M., Guzmán, C., Larios, V.M.: Synergy between smart cities' hackathons and living labs as a vehicle for accelerating tangible innovations on cities. In: *IEEE 2nd International Smart Cities Conference: Improving the Citizens Quality of Life, ISC2 2016 - Proceedings* (2016)
43. Kendrew, S., Deen, C., Radziwill, N., Crawford, S., Gilbert, J.: The first SPIE software hack day. In: *Proceedings of SPIE, Montreal* (2014)
44. Saravi, S., et al.: A systems engineering hackathon - a methodology involving multiple stakeholders to progress conceptual design of a complex engineered product. *IEEE Access* **6**, 38399–38410 (2018)
45. Charvat, K., Bye, B.L., Mildorf, T., Berre, A.-J., Jedlicka, K.: Open data, VGI and citizen observatories INSPIRE hackathon. *Int. J. Spat. Data Infrastruct. Res.* **13**, 109–130 (2018)
46. Pathanasethpong, A., et al.: Tackling regional public health issues using mobile health technology: event report of an mHealth hackathon in Thailand. *JMIR mHealth uHealth* **5**, e155 (2017)
47. Bernardo, F., Grierson, M., Fiebrink, R.: User-centred design actions for lightweight evaluation of an interactive machine learning toolkit. *J. Sci. Technol. Arts.* **10**, 25–38 (2018)

48. Lau, K., Lei, B.J.: IMS2017 hackathon: 30-minute circuits. *IEEE Microw. Mag.* **18**, 84 (2017)
49. Ahalt, S., et al.: Water science software institute: agile and open source scientific software development. *Comput. Sci. Eng.* **16**, 18–26 (2014)
50. Thomas, R.S.: Using design slam to foster lifelong learning solutions. *Comput. (Long Beach Calif.)* **50**, 32–33 (2017)
51. Concilio, G., Molinari, F., Morelli, N.: Empowering citizens with open data by urban hackathons. In: *Proceedings of the 7th International Conference for e-Democracy and Open Government, CeDEM 2017*, pp. 125–134 (2017)
52. Avalos, M., Larios, V.M., Salazar, P., Maciel, R.: Hackathons, semesterathons, and summerathons as vehicles to develop smart city local talent that via their innovations promote synergy between industry, academia, government and citizens. In: *2017 International Smart Cities Conference ISC2 2017* (2017)
53. Brenner, W., et al.: User, use & utility research: the digital user as new design perspective in business and information systems engineering. *Bus. Inf. Syst. Eng.* **6**, 55–61 (2014)
54. Decker, A., Eiselt, K., Voll, K.: Understanding and improving the culture of hackathons: think global hack local. In: *Frontiers in Education Conference* (2015)
55. Alekseenko, A., Sanchez-medina, J.: Lane departure prediction with naturalistic driving data (2017)
56. Tsukada, M., et al.: Software defined media: virtualization of audio-visual services. In: *IEEE International Conference on Communications*, pp. 1–7. IEEE (2017)
57. Madelska, S.: News from the field: coders and the creative unite to design and build apps for surface haptics. *IEEE Trans. Haptics* **8**, 128–129 (2015)
58. Gregor, S., Hevner, A.R.: Positioning and presenting design science research for maximum impact. *MIS Q.* **37**, 337–355 (2013)
59. Seidel, S., Mueller-Wienbergen, F., Rosemann, M.: Pockets of creativity in business processes. *Commun. Assoc. Inf. Syst.* **27**, 415–436 (2010)
60. Rosemann, M., vom Brocke, J.: The six core elements of business process management. In: vom Brocke, J., Rosemann, M. (eds.) *Handbook on Business Process Management 1*. IHIS, pp. 105–122. Springer, Heidelberg (2015)



Design Patterns for Business Process Individualization

Bastian Wurm¹(✉), Kanika Goel², Wasana Bandara², and Michael Rosemann²

¹ Vienna University of Economics and Business, Vienna, Austria
bastian.wurm@wu.ac.at

² Queensland University of Technology, Brisbane, Australia
{k.goel,w.bandara,m.rosemann}@qut.edu.au

Abstract. Competition is forcing organizations to constantly innovate and identify ways to deliver high quality services and products. The Business Process Management (BPM) discipline has contributed by providing a rich set of analysis and re-design techniques. However, BPM methods and guidelines are often driven by process standardization and economies of scale, while emerging digital technologies (e.g. advanced manufacturing, sophisticated data analytics) increasingly facilitate process individualization. In this paper we contribute to an extended BPM body of knowledge by presenting design patterns for process individualization. We argue that (1) technological developments have made scalable process variant management viable and that (2) these technologies enable new forms of process individualization altogether. In our research, we identified and analyzed design patterns that make use of rapid digitalization to obtain individualized products and services. A conceptual model supported by literature and case examples is presented. This model forms theory on design and action of business process individualization in the digital age. Companies can deploy the design patterns developed in this paper as guidelines in their quest for process individualization.

Keywords: Process individualization · Design patterns · Process design · Digital technologies

1 Introduction

An increasingly competitive environment and differentiated customer demands have shifted the focus from mass production to individualization of products and services over the last decades [57]. However, individualization entails increased complexity costs, as it requires different business process variants to be designed, implemented, managed and maintained. The more diverse the customer needs a company wants to address, the more process variants are required [51], leading to higher costs-to-serve. This creates a dilemma between revenue-sensitive process individualization and cost-sensitive process standardization, which can be observed in a number of industries (e.g. see detailed elaboration for the car manufacturing industry in [25]).

Established Business Process Management (BPM) methods and process-oriented improvement programs such as Six Sigma, Total Quality Management, Lean Management, and popular process management life-cycle models (e.g. [16]), tend to focus on process standardization over individualization. These methods are driven by measures such as processing time, cost-per-outcome unit or minimal variation (Six Sigma), but rarely give guidance in terms of how to individualize processes leading to higher variety of services and products [17,41]. Ignoring individualization will no longer be viable for organizations [57], creating an increased demand for insights on how to approach it.

The rapid developments in the digital age, especially in the form of advanced manufacturing, robotic workflow management and data analytics brings tremendous potential to make process individualization cost-effective. These new affordances materialize in a higher variety of process outcomes and a reduced time to individualize. Thus, we argue that digital technologies have broadened the design space of business processes, providing the means for firms to individualize their products and services. However, the existing BPM method set does not provide sufficient support for capitalizing on this emerging affordance. We contribute towards addressing this gap by developing a defined set of design patterns for process individualization. In this way, we address the following research question: *What are design patterns for process individualization?*

This paper presents a conceptual model that explains how business processes can be individualized in the digital age. In particular, we derive four different design patterns for business process individualization. These patterns cover the essential constructs making up a business process. Contemporary case examples are provided in support of each design pattern as a means of illustrative evidence. The resulting framework contributes towards theory building by presenting a forming typology of process individualization with actionable design options. Thereby, we contribute towards forming theory of analyzing (Type I) and theory of design and action (Type V), as described in [21].

This paper is structured as follows. First, we outline the research background. Second, we introduce the design method that explains how we derived our theoretical model of process individualization patterns. Third, we present the conceptual model and explain the different design patterns for process individualization. We conclude the paper by summarizing its key contribution, and pointing to limitations and future work.

2 Research Background

Creating diverse process variants in an efficient way is a major barrier to the individualization of products and services. The developments in the digital age, especially robotic workflow management, sophisticated data analytics and advanced manufacturing (e.g. 3D printing) provide an entire new level of cost effective capabilities and make previously impossible forms of scalable process individualization accessible. In this section we summarize the contextual background to the key concepts underlying this study.

2.1 Stages of Individualization

Individualization refers to the degree to which products, services, and processes are configured to meet explicit as well as latent customer needs [57]. We distinguish between three stages of individualization (see Fig. 1).

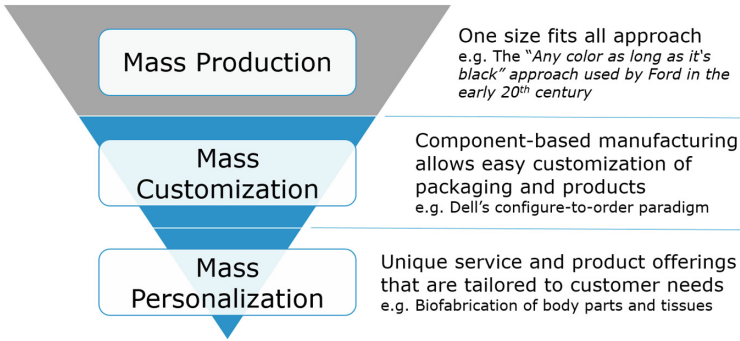


Fig. 1. Stages of individualization

The first stage consists of only standard products with no or very little individualization. Products and services that fall into this category are off-the-shelf products that follow a “one size fits all” approach. These products and services are uniform, as variation is either technically or economically not feasible, or there is no market for these offerings. On this stage of product and service individualization the customer does not play a role other than deciding and buying one of the products available in the market [31]. An example is Ford’s “any color as long as it’s black” approach in the early 20th century. By eliminating variation in the product, Ford was able to scale production in a way that was unprecedented. Nowadays, this level of individualization is predominant for daily products, where the costs of individualization outweigh the potential mark-up in price.

The second stage encompasses mass customized products or services that exhibit a certain level of individualization by offering different variants of the same product or service. As outlined by [19], there are different strategies that can be employed. Mass customization heavily relies on component-based manufacturing that enables companies to offer limited variation while still realizing returns of scale. In mass customization, the role of the customer is to choose from a variety of different modules that are produced and resembled by the manufacturer on large scale [31]. For example, Dell and many other computer manufacturers allow customers to configure their device based on a pre-selection of components. In this stage, individualization is characterized by the plethora of options that are provided to the customer and the sophistication of product configuration and pricing engines that guide the user through the process.

The third and last stage reflects the highest level of individualization. Here, products and services are bespoke to an individual customer or purpose. Mass personalization combines the following four key properties as according to [57]. (1) As these products and services are tailored to specific customer needs, they are one of a kind. This is also referred to as the market-of-one. (2) Mass personalization needs to be paired with mass efficiency in order to be economically viable. (3) Companies need to employ customer co-creation to integrate the customer in all phases of the product life-cycle. (4) With an increasing level of product and service individualization, it becomes more important to detect and understand explicit as well as latent customer needs [34]. Modern technology enables this level of individualization. For example, hospitals can produce body parts by use of additive manufacturing (colloquially know as 3D printing), considering a patient's unique physical characteristics [33].

2.2 Business Process Variant Management

Process variants are created to configure processes to diverse contexts due to varying environmental and market conditions [10]. To respond to customer needs in different markets, products and services are adapted. In turn, underlying processes often need to be altered in order to reflect these changes [51].

To allow for the generation and management of different process variants, research has investigated configurable process models [20], software product lines [39], and assembly system design [26], to name only but a few. These approaches have in common that they exclusively focus on the sequence of activities and their causal relationship to create variation, but do not consider other components of business processes. Thus, the more customized products and services a firm wants to offer, the more diverse its business processes need to be leading to inefficiency and increased complexity in the management of the business process portfolio [51]. This is, why many companies refrain from competing via business process individualization, but aim at standardizing their business processes instead [53].

Furthermore, process variants operate on the level of mass customization, i.e. they result in a set of product options the customer can choose from. However, process variants are limited in that they cannot (and do not aim to) provide a unique customer experience and tailoring of the process.

2.3 Business Process Improvement and Redesign Patterns

Prior research provides a rich set of process improvement and process re-design methods and patterns. These patterns “target the resolution or mitigation of problems” [18, p. 8] to increase efficiency and improve other metrics of the devil's quadrangle [18] by addressing the “mechanics of the process” [42, p. 283]. Yet, how to provide companies with a set of design guidelines to differentiate their processes and customer touch points from competitors has so far been neglected.

We address these limitations by developing design patterns for process individualization. The patterns contribute to the body of knowledge in BPM by explaining and demonstrating how different components of business processes

can be manipulated to individualize business processes and broaden their design space.

3 Design Method

In this section we explain how we derived design patterns for process individualization. We first introduce the key elements forming a business process, as all of these can potentially be manipulated to increase process variation. Second, we introduce the notion of design patterns [4] and describe how we utilized this concept for our research.

3.1 Conceptual Framework

Business processes are a sequence of activities leading to an output that generates value to an internal or external customer of an organization [16] by transforming inputs to outputs [32]. Figure 2 visualizes the different components of business processes and their interplay. A business process is composed of (1) process activities and buffers together with their respective sequence. Activities and steps of the process are carried out by (2) resources that are either capital assets or labor. Further, (3) the information and data associated with the process help to make process decisions and trigger process activities or sub-processes [32]. (4) The flow unit is a transient entity “that proceeds through various activities and finally exits the process as finished output” [32, p. 19].

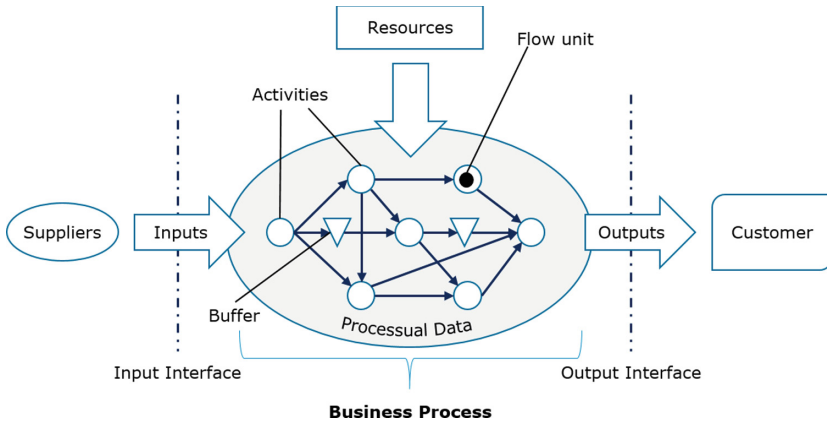


Fig. 2. Business process meta-model (adapted from [32])

Traditionally, companies focused on creating changes to the output of the process by using different inputs or adapting process activities and the sequence

of the process. However, this increases complexity and costs. How business processes can be individualized by changes to other components such as resources, data, and the flow unit of the business process has received limited attention.

In the remainder of this paper, we address business process individualization and how digital technologies can contribute to resolving the dilemma of increased complexity costs versus satisfying external demands for individualized services. First, variation to a process cannot only be achieved by creating diverse process sequences and activities, but also by manipulation of resources, data, and the flow unit of the business process. Second, technological developments enable lower costs of providing process variation. Especially, ready access to and sophisticated analyses of vast amounts of data can increase the cost effective feasibility of process variations and change the organizing logic [54] of process design.

3.2 Synthesis of Design Patterns

The approach we employ in this study is based on the notion of design patterns introduced by Alexander [3]. With the term pattern, Alexander describes the description of an artefact endowed with a guideline that can be used for creating the artefact [4]. Further, “the term pattern appeals to the replicated similarity in a design, and in particular to similarity that makes room for variability and customization in each of the elements” [15, p. 1]. Thus, design patterns serve as general solutions to reoccurring problems, while leaving room for creative freedom [3]. In BPM, patterns have been discussed, amongst others, in the context of process models [49], control flow [2], and data flow [45].

To derive the design patterns we employed *heuristic theorizing* as outlined by Gregory and Muntermann [22]. This approach is suitable, since design patterns are a form of heuristics that help to reduce the search for a satisfactory solution [22, p. 642]. First, we *entered the heuristic search process and defined the problem at hand*, i.e. the generation of design patterns for business process individualization. Since we soon realized that process individualization can be approached from different angles, we *decomposed the problem* into simpler problems that could be approached individually. That is, we reformulated the problem to derive patterns for individualization for each of a business process’ components, as defined above [32]. Next, we reviewed literature from real-life and published sources pertaining to case examples and the theoretical analysis of individualization. This extraction of prescriptive design knowledge from existing artifacts is also referred to as *design archaeology* [12]. Based on this information, we started *generating the design patterns*, followed by multiple rounds of *heuristic synthesis* and what Sein and associates call ‘*reflection and learning*’ [48]. When our patterns became stable, i.e. new cases did not change the derived design components, we *finalized the design patterns* and *exited the heuristic search process*. This *abductive* line of reasoning is common in design synthesis [30] and similar to the line of argument by Reijers et al. [42], who derive redesign patterns based on principles observed in practice and described in literature.

4 Design Patterns for Process Individualization

Based on the cases and literature analyzed, we derived four design patterns: (1) process sequence and activity individualization, (2) flow unit individualization, (3), resource individualization, and (4) data individualization. These patterns are distinct as each of them addresses a different facet of a business process. Thus, each design pattern can be used individually as well as in combination with one or more of the other design patterns. We capture this distinctive, yet interacting behavior of the patterns in Fig. 3, which depicts the resulting conceptual model.

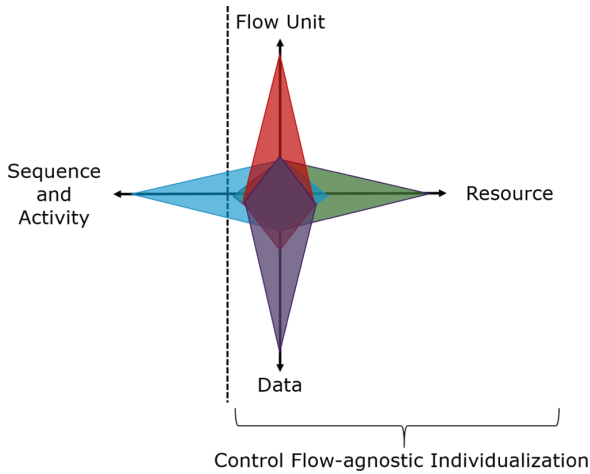


Fig. 3. Design patterns for business process individualization

Process sequence and activity individualization captures how technological developments can be used to more efficiently and effectively adjust the orchestration of tasks/steps and their sequence to provide an individualized outcome. The other three design patterns (flow unit, resource and data individualization) are *control flow-agnostic*, i.e. they do not change the control flow. We define control flow-agnostic individualization as: *Individualization that is characterized by the manipulation of data, resources, and/or the flow unit of a business process in order to tailor process outcomes to a specific customer or purpose, while at the same time, sequence and activities of the respective business process remain unaltered.*

Each design pattern is clustered in opportunity-driven and demand-driven strategies of business process individualization. Demand-driven individualization addresses explicit as well as implicit customer needs. This allows organizations to cope with increasingly differentiated customer demands [57]. In contrast, opportunity-driven individualization covers new ways that companies can use to individualize their processes by capitalizing on the rich pool of their internal resources. By aligning the business process with its context [10], effectiveness and efficiency can be further increased. In the following sections, we explain

each design pattern in detail and provide evidence from practice how they can be utilized.

4.1 Sequence and Activity Individualization

Description. Sequence and activity individualization refers to the adaptation of process sequence and/or activities without changing the actual product or service. Accordingly, process sequence and execution of activities are bespoke to a particular customer, situation or condition.

Demand-Driven Sequence and Activity Individualization. First, process sequence and activity individualization can serve as a means to respond to customer needs. For example, consider the hotel brand Ritz-Carlton. Even though every guest receives the same service, i.e. an overnight stay at a hotel, the way different activities are executed is very much tailored to each individual customer [24]. Wilder, Collier and Barnes [52] discuss in detail how frontline employees can be supported in providing an adaptive service experience to customers. Beyond the question of how activities are carried out, Artificial Intelligence can provide support in choosing the next most suitable activity, i.e. answering the question what activity should be carried out. For example, IBM Watson for oncology can recommend the best next treatment option for cancer patients based on patient information and historical treatment data [27].

Opportunity-Driven Sequence and Activity Individualization. Second, companies can use process-related characteristics and information, such as process logs, to increase early individualization of their business processes. Based on historic and run-time information about the business process, more (precise) business rules and conditions can be formulated to trigger alternative process sequences. Pattern recognition can be used to derive detailed business rules given prior process instances and help configuring more individualized, robotic workflows that would not have been possible with human resources. Using current information including information about the flow unit (e.g. customer history), run-time adjustments to the process can be made. For example, banks use a combination of various attributes to calculate the loan default risk employing neural networks. These attributes can be re-incorporated into the process. Depending on the characteristics of a loan applicant, the extent of the process varies. Thus, applicants with high default risk run through a comprehensive screening, while for wealthy individuals with negligible default risk this process is shortened.

Third, environmental events or conditions can affect the execution of business processes. For example, Rosemann et al. [44] describe the claims handling process at an Australian insurance provider, which is highly context-sensitive. As some of Australia's regions are prone to storms in summer, there are considerable more claims requests in summer compared to winter. Due to the large variance in numbers of cases (more than double), the insurance provider operates two season-dependent variants of the claims handling process. While in winter there is a full registration of the claim, in summer (when there is storm season) the claims handling process is shortened to include a rapid lodgment of the claim.

Additionally, staff from other departments is re-deployed and casual staff are hired. This way, the insurance provider can offer a similar processing time, irrespective of the season. For a more detailed description of this case see [44]. The availability of micro-data has made such process individualization even more specific. For example, context-specific data about the situation of a customer (e.g. location of a car) can be used to trigger specific processes (e.g. automated remote speed-control).

Technical Realization. This type of individualization makes use of available environmental, processual, and customer related information. As processes become increasingly branched and individualized early, Artificial Intelligence can help route the respective flow units through the process. AI-algorithms can determine patterns in the data and create new process variants as the process is being executed. New variants can be compared to the existing process ‘in the shadow’ without impacting customers or process workers [46]. This allows not only for the formulation of more decision points, but also more complex business logic. For example, multiple attributes can be combined in order to make a decision. We can imagine that this eventually causes a shift from ex-ante process design to ongoing and even run-time and predictive process design. At the same time, declarative process modelling [38] can serve as an important tool to specify constraints in which the process has to be executed.

4.2 Flow Unit Individualization

Description. Flow unit individualization refers to the manipulation of the flow unit of a business process to deliver a unique outcome. This is the first of the three control flow-agnostic forms of individualization we present here. The sequence and activities of the process remains the same, but the flow unit is manipulated and exits the process as final individualized output.

Demand-Driven Flow Unit Individualization. Unique demands of customers can be met using flow unit individualization. For example, in the medical industry, biofabrication enables to print cell fibers, bones and even organs [33]. In case of an emergency, doctors and paramedics might soon be able to collect the details related to the injury of the patient and print the required body part. Therefore, a fiber, bone, or any other body part can be produced based on patient (customer) needs. The asset sharing platform launched by Healx is another example of demand driven flow unit individualization. It uses a machine learning algorithm based on patient’s biological information to match drugs to disease symptoms and also reveal the level of effectiveness for that particular patient [29].

Koren et al. [31] discuss how the development of so-called open products allows for the re-configuration of products even after production. This product class is characterized by its customizability that allows adding physical components once the product has been purchased by the customer. Further, where the physical and the digital meet, products become re-programmable [55]. This allows producers as well as third party providers to constantly add new features

to an existing physical device. Any smart device that allows for the installation of apps, such as smartphones, smartwatches, smartspeakers, etc., falls under this category. This is referred to as ‘late individualization’ as it only occurs when the customer is using the product over time, e.g. a standard smart phone at the point of purchase becomes a highly individualized product due to customers’ downloads and settings.

Opportunity-Driven Flow Unit Individualization. Opportunity-driven individualization is initiated by internal needs of the organization. Internal needs cover various factors such as past knowledge, dearth or scads of resources or immediate need of a resource. The aerospace industry is utilizing the potential of 3D printing, as it provides unparalleled freedom in component design and fabrication [28]. Rolls-Royce is leveraging the potential of 3D printers to design and manufacture power systems for aircraft. The technology provides various advantages including design flexibility, quick iteration, and part consolidation [8]. 3D printing can also be used to produce spare parts and to repair machinery. This is of particular advantage, when idle time of capital assets result in large costs, as it is the case for trains [23], aircraft [5], different types of water-craft [40] as well as machines that are critical for production.

Flow unit individualization provides various advantages to consumers as well as businesses. Being able to produce a product based on unique consumer needs, increases customer satisfaction and trust [37]. Whereas, using available technology and resources to satisfy the internal needs of the organizations, results in reduced cost and time for businesses.

Technical Realization. Using technologies such as additive manufacturing and making use of open products and re-programmability, flow unit individualization can assist in producing a variety of products and services catered to individual consumer needs in an affordable manner. Flow unit individualization makes use of sophisticated technology to produce products based on internal needs, or cater for specific customer needs.

4.3 Resource Individualization

Description. Resource individualization determines the most appropriate resource for the activities a flow unit runs through. Because activities and process sequence remain unaltered, this type of individualization is control flow-agnostic. While resource specialization has undoubtedly led to a high level of efficiency in regards to activity execution, we argue that resources can also be used as a means to broaden the design space of business processes. Whether it is selecting an Uber driver with unique language skills for your ride to the airport, finding the right nanny for your child, or getting advice from someone with the same medical issue as yourself on patientslikeme.com, this type of individualization matches customer specific characteristics and requirements with available resources, while leaving sequence and activities of the process unchanged.

Demand-Driven Resource Individualization. Demand-driven resource individualization allows companies to respond to characteristics and likings of

the customer by selecting a resource that matches these specifications. Applied correctly, resource individualization helps to increase customer satisfaction by tailoring services to customer needs. For this type of individualization, resources' personal traits and characteristics that match with customers' desires are of particular relevance. While complete business models of many start-ups and apps rely on matching resources, there are only few companies that use this form of resource individualization in their regular end-to-end business processes.

Platforms like Patientslikeme [36], Tinder [50], and CareGuide [11] allow customers to find patients with similar medical symptoms, a potential partner of their liking, or a perfect nanny to take care of their child. These companies provide a platform to choose resources as per one's own requirements, making the customers feel privileged and taken care of. By providing a platform, these applications essentially connect people with a predefined set of characteristics.

Opportunity-Driven Resource Individualization. Opportunity-driven resource individualization makes use of the diversity that the company internal resource-pool provides. Companies can incorporate screening processes into their regular business processes to check for and optimize current resources' utilization. When new process instances arrive, they can be routed to the most appropriate resource based on a set of predefined constraints. For example, an airline may route customer inquiries depending on the language proficiency of the customer calling. I.e. a Mandarin speaking customer will be matched with an agent that is familiar with Mandarin.

Resource individualization can also be used to ease critical process steps. This is the case when particular process activities determine the process outcome to a large extent. Often these parts of the process are non-routine and knowledge-intensive. By using specialized resources, companies are able to increase customer satisfaction, increase profitability, and guarantee that safety critical, legal or health related process aspects are executed correctly. In comparison to customer-oriented resource individualization, resources employed in process critical steps need to be highly specialized and have expert knowledge.

Insurance companies, for example, use resource individualization as part of a two-step procedure for their claims handling process. Customers calling to report on a claim, first report on some general information to an artificial call agent. The artificial call agent screens the information for inconsistencies and conspicuous patterns. If the algorithm detects any anomalies, the customer will be routed to an experienced, human call center agent. In a second step, the customer will be asked to provide more detailed information on the claimed case. A very experienced call agent can work on hard cases where fraud seems likely, while a new employee can work on easy cases that demonstrated low fraud potential in the first step of the process. As this case shows, resource individualization is most powerful, if combined with the analysis of customer information collected prior to or during process execution.

Technical Realization. From a technological perspective, resource individualization is enabled by data analytics and the information availability on customer characteristics and requirements as well as resource specific characteristics and

features. With this information, companies can use classification algorithms to detect similarities and matches between customers and resources.

For demand-driven resource individualization, employing or buying all this different types of resources is not an economically viable strategy as this will cause high expenditures and yield low resource utilization. For this reason, companies can make use of crowdsourcing [9] and source tasks to individuals. For example, companies can use 99 Designs to commission different types of designs [1]. In many cases the outsourcing of tasks and sub-processes will require companies to build a community that they can default and delegate to.

For process critical steps, resource individualization helps to match already available resources with tasks that require their level of expertise. As these tasks often mark critical points of the process, require a high level of skills, experience, and trust, they cannot simply be sourced out. Also, there are only small incentives for highly experienced knowledge workers to participate in crowdsourcing.

4.4 Data Individualization

Description. With big data at the forefront, data serves as a valuable resource to make decisions [43]. Firms have access to vast amounts of customer, processual, and environmental data that provides innovation opportunities [35]. Amongst others, data can be used in a more integrated manner, to enable more agile, more accurate decision-making, and modify digital products.

Demand-Driven Data Individualization. Data can be individualized to meet the unique demands of the customer. Products and services that do not have a physical representation (anymore) such as music and movie streaming, but also insurances can make use of data individualization. For example, YouI car insurance adapts insurance premia of policy takers depending on how they use their car [56].

With the use of smart contracts on blockchain [14] or any other decentralized ledger, contracts can be automatically enforced. Insurance provider AXA offers a flight insurance against delayed flights [7] based on customer related data saved on an Ethereum blockchain. As the underlying smart contract is connected to a global air traffic database, policy holders are automatically reimbursed, once their flight is delayed. This principle can be used to adjust insurance payments to biological information, as available from fitbits or any smartwatch. This enables the tailoring of health insurances to individual characteristics and behavior of the customer. From an economic perspective, this tailoring can contribute to resolving information asymmetry, adverse selection, and moral hazard leading to more fair insurance premia.

Opportunity-Driven Data Individualization. Data individualization can also be used to enhance the decision-making process, contributing to the effectiveness of process outcomes. E.g. the Australian Taxation Office uses information from a range of social media sources to ensure relevant information is provided. Third-party sources include government bodies, employers, online selling platforms, stock exchanges, amongst others [6]. Social media and other

personal data may be reviewed further, to make the right decision, when reviewing tax applications. Therefore, individual data can be used to make enhanced decisions, enabling organizations to achieve their goals in an effective manner.

Technical Realization. Data individualization is enabled by capturing, storing, and analyzing large volumes of data. Data analytics in form of Machine Learning and Artificial Intelligence provide the means to do so. To reap the benefits associated with this pattern, a strong foundation in data management is obligatory. For instance, to transform to a data driven insurer, AXA required cooperation of the data analytics, data management office and data engineering units [47]. The 3Vs, respectively 4Vs feature of big data [13] summarize the key challenges. First, companies need to be able to integrate and process data of different types and from various sources. Second, the volume of data is constantly increasing. Third, new data is continuously created (e.g. by sensors and equipment) and streamed to assigned data bases.

5 Summary and Conclusion

This paper presented a conceptual framework of four distinct design patterns for process individualization, namely: (1) sequence and activity individualization, (2) flow unit individualization, (3) resource individualization, and (4) data individualization. Drawing upon design patterns theory [3,4], the framework is built to extrapolate how different components of a process can be manipulated to derive individualization options that broaden the design space of business processes. The suggested design patterns also guide organizations on how to best apply digitalization to efficiently obtain individualized products and services at lower costs.

The resulting framework is the first to theorize about process individualization. First, by presenting a classification of process individualization options, we contribute towards theory of analyzing (Type 1), as explained by Gregor [21]. We identify and describe what different process individualization options are. We provide various examples that demonstrate how the patterns (can) materialize. Secondly, the actionable design options provide guidance on how to usefully deploy the design patterns. Presenting each design pattern in the form of sub-patterns, technical realizations and illustrative examples contributes towards theory of design and action (Type 5) [21] on process individualization.

The framework acts as a useful reference and guide for different stakeholders. For example, process owners/process change champions can use the framework to identify new opportunities to enhance their business processes with individualized products and services. Process architects can use the framework when (re-)designing a single process or a portfolio of processes; it can be applied to individualize existing processes or to design new individualized processes. It is also applicable for those engaged in product and service innovations, where a single or few of these patterns may be considered to create innovative customer experiences.

While there are many benefits, this work is still in its genesis and calls for further research. The empirical support for the design patterns were based on literature and (limited) case examples from practice. Detailed empirical validation, i.e. through in-depth case studies of existing individualization practices or

Action Design research where these patterns are newly executed, is warranted to further validate the patterns. The design patterns also need further specification in order to enhance their utility; in terms of its contextual applicability (i.e. how the patterns may be differently applicable based on process, organization and external-environmental contexts), and having detailed and validated procedure guidelines on how to implement the design patterns. Also, given the reliance of data across all patterns, data management considerations need to be carefully thought through and managed in order to maintain customer trust.

Acknowledgements. The work of Bastian Wurm has received funding from the EU H2020 program under the MSCA-RISE agreement 645751 (RISE_BPM).

References

1. 99 Designs: Design makes anything possible. <https://99designs.com.au/>
2. van der Aalst, W.M.P., ter Hofstede, A., Kiepuszewski, B., Barros, A.P.: Workflow patterns. *Distrib. Parallel Databases* **14**(1), 5–51 (2003)
3. Alexander, C.: *Notes on the Synthesis of Form*. Harvard University Press, Cambridge (1964)
4. Alexander, C.: *The Timeless Way of Building*. Oxford University Press, New York (1979)
5. Alhart, T.: Brothers in arms: these robots put a new twist on 3D printing (2017). <https://www.ge.com/reports/brothers-arms-robots-put-new-twist-3d-printing/>
6. Australian Taxation Office: ATO Privacy Policy (2018). <https://www.ato.gov.au/About-ATO/Commitments-and-reporting/In-detail/Privacy-and-information-gathering/Privacy-policy/>
7. AXA: AXA goes blockchain with fizzy (2017). <https://www.axa.com/en/newsroom/news/axa-goes-blockchain-with-fizzy>
8. Boissonneault, T.: Rolls-Royce moves ahead with 3D printed Advance3 aircraft demonstrator engine (2018). <https://www.3dprintingmedia.network/rolls-royce-advance3-engine/>
9. Brabham, D.C.: Crowdsourcing as a model for problem solving. *Int. J. Res. New Media Technol.* **14**(1), 75–90 (2008)
10. vom Brocke, J., Zelt, S., Schmiedel, T.: On the role of context in business process management. *Int. J. Inf. Manag.* **36**(3), 486–495 (2016)
11. Careguide: Hire a Nanny or Find a Nanny Share. <https://careguide.com/>
12. Chandra Kruse, L., Seidel, S., vom Brocke, J.: Design archaeology: generating design knowledge from real-world artifact design. In: Tulu, B., Djamasbi, S., Leroy, G. (eds.) *DESRIST 2019*. LNCS, vol. 11491, pp. 32–45. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19504-5_3
13. Chen, M., Mao, S., Liu, Y.: Big data: a survey. *Mob. Netw. Appl.* **19**(2), 171–209 (2014)
14. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. *IEEE Access* **4**, 2292–2303 (2016)
15. Coplien, J.O.: Software design patterns: common questions and answers. In: *The Patterns Handbook: Techniques, Strategies, and Applications*, pp. 311–320 (1998)
16. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*, 2nd edn. Springer, Heidelberg (2018). <https://doi.org/10.1007/978-3-662-56509-4>

17. Estrada-Torres, B., del-Río-Ortega, A., Resinas, M., Ruiz-Cortés, A.: Identifying variability in process performance indicators. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNBIP, vol. 260, pp. 91–107. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45468-9_6
18. Falk, T., Griesberger, P., Johannsen, F., Leist, S.: Patterns for business process improvement - a first approach. In: ECIS 2013 (2013)
19. Gilmore, J.H., Pine, B.J.: The four faces of mass customization. *Harv. Bus. Rev.* **75**(1), 91–101 (1997)
20. Gottschalk, F., Wagemakers, T.A., Jansen-Vullers, M.H., van der Aalst, W.M., La Rosa, M.: Configurable process models: a municipality case study. In: CAiSE 2009 (2009)
21. Gregor, S.: The nature of theory in information systems. *MIS Q.* **30**(3), 611–642 (2006)
22. Gregory, R.W., Muntermann, J.: Heuristic theorizing: proactively generating design theories. *Inf. Syst. Res.* **25**(3), 639–653 (2014)
23. Hafner, B.: Ersatzteile selber machen: 3D-Drucker machen's möglich (2018). <https://blog.railcargo.com/ersatzteile-selber-machen-3d-drucker-machens-moeglich/>
24. Hall, J.M., Johnson, E.M.: When should a process be art, not science? *Harv. Bus. Rev.* **87**(3), 58–65 (2009)
25. Heese, H.S., Swaminathan, J.M.: Product line design with component commonality and cost-reduction effort. *Manuf. Serv. Oper. Manag.* **8**(2), 206–219 (2006)
26. Hu, S.J., et al.: Assembly system design and operations for product variety. *CIRP Ann. - Manuf. Technol.* **60**(2), 715–733 (2011)
27. IBM: IBM Watson for Oncology. <https://www.ibm.com/us-en/marketplace/ibm-watson-for-oncology>
28. Joshi, S.C., Sheikh, A.A.: 3D printing in aerospace and its long-term sustainability. *Virtual Phys. Prototyp.* **10**(4), 175–185 (2015)
29. Kavadias, S., Ladas, K., Loch, C.: The transformative business model. *Harv. Bus. Rev.* **94**, 91–98 (2016)
30. Kolko, J.: Abductive thinking and sensemaking: the drivers of design synthesis. *MIT's Des. Issues* **26**(1), 15–28 (2010)
31. Koren, Y., Shpitalni, M., Gu, P., Hu, S.J.: Product design for mass-individualization. *Procedia CIRP* **36**, 64–71 (2015)
32. Laguna, M., Marklund, J.: *Business Process Modeling, Simulation and Design*. CRC Press, Boca Raton (2013)
33. Mironov, V., Kasyanov, V., Markwald, R.R.: Organ printing: from bioprinter to organ biofabrication line. *Curr. Opin. Biotechnol.* **22**(5), 667–673 (2011)
34. Montgomery, A.L., Smith, M.D.: Prospects for personalization on the internet. *J. Interact. Market.* **23**(2), 130–137 (2009)
35. Parmar, R., Mackenzie, I., Cohn, D., Gann, D.: The new patterns of innovation: how to use data to drive growth. *Harv. Bus. Rev.* **92**(Jan–Feb), 86–95 (2014)
36. Patientslikeme: Living better starts here. <https://www.patientslikeme.com/>
37. Piccoli, G., Lui, T.W., Grün, B.: The impact of IT-enabled customer service systems on service personalization, customer service perceptions, and hotel performance. *Tour. Manag.* **59**, 349–362 (2017)
38. 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. LNBIP, vol. 99, pp. 383–394. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_37

39. Pohl, K., Böckle, G., van Der Linden, F.J.: *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer, Heidelberg (2005). <https://doi.org/10.1007/3-540-28901-1>
40. Pomerantz, D.: Ship shapes: new 3D printing research aims to rejuvenate navy gear (2018). <https://www.ge.com/reports/ship-shapes-new-3d-printing-research-aims-rejuvenate-navy-gear/>
41. Reichert, M., Hallerbach, A., Bauer, T.: Lifecycle management of business process variants. In: vom Brocke, J., Rosemann, M. (eds.) *Handbook on Business Process Management 1*. IHIS, pp. 251–278. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-642-45100-3_11
42. Reijers, H.A., Liman Mansar, S.: Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. *Omega* **33**(4), 283–306 (2005)
43. Repenning, N.P., Kieffer, D., Repenning, J.: A new approach to designing work. *Harv. Bus. Rev.* **59**(2), 29–38 (2018)
44. Rosemann, M., Recker, J., Flender, C.: Contextualization of business processes. *Int. J. Bus. Process Integr. Manag.* **3**(1), 47–60 (2008)
45. Russell, N., ter Hofstede, A.H., Edmond, D., van der Aalst, W.M.P.: *Workflow data patterns*. Technical report FIT-TR-2004-01, Queensland University of Technology (2004)
46. Satyal, S., Weber, I., Paik, H., Di Ciccio, C., Mendling, J.: Shadow testing for business process improvement. In: Panetto, H., Debruyne, C., Proper, H., Ardagna, C., Roman, D., Meersman, R. (eds.) *OTM 2018*. LNCS, vol. 11229, pp. 153–171. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_9
47. Scheffler, A., Wirths, C.P.: Data innovation @ AXA Germany: journey towards a data-driven insurer. In: Urbach, N., Roeglinger, M. (eds.) *Digitalization Cases: How Organizations Rethink their Business for the Digital Age*, pp. 363–378 (2019)
48. Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R.: Action design research. *MIS Q.* **35**(1), 37–57 (2011). <https://doi.org/10.2307/23043488>
49. Smirnov, S., Weidlich, M., Mendling, J., Weske, M.: Action patterns in business process models. *Comput. Ind.* **63**(2), 98–111 (2012)
50. Tinder: Match. Date. Chat. <https://tinder.com/?lang=en>
51. Tregear, R.: Business process standardization. In: vom Brocke, J., Rosemann, M. (eds.) *Handbook on Business Process Management 2*. INFOSYS, pp. 307–327. Springer, Berlin Heidelberg (2010). https://doi.org/10.1007/978-3-642-01982-1_15
52. Wilder, K.M., Collier, J.E., Barnes, D.C.: Tailoring to customers’ needs: understanding how to promote an adaptive service experience with frontline employees. *J. Serv. Res.* **17**(4), 446–459 (2014)
53. Wurm, B., Schmiedel, T., Mendling, J., Fleig, C.: Development of a measurement scale for business process standardization. In: *ECIS 2018* (2018)
54. Yoo, Y., Boland, R.J., Lyytinen, K., Majchrzak, A.: Organizing for innovation in the digitized world. *Organ. Sci.* **23**(5), 1398–1408 (2012)
55. Yoo, Y., Lyytinen, K., Boland, R., Berente, N., Gaskin, J., Schutz, D.: The next wave of digital innovation: opportunities and challenges. Research workshop: “digital challenges in innovation research”, pp. 1–37 (2010)
56. You! Car Insurance: A different kind of car insurance. <https://www.youi.com.au/car-insurance>
57. Zhou, F., Ji, Y., Jiao, R.J.: Affective and cognitive design for mass personalization: status and prospect. *J. Intell. Manuf.* **24**(5), 1047–1069 (2013)



Understanding the Alignment of Employee Appraisals and Rewards with Business Processes

Aygun Shafagatova^(✉)  and Amy Van Looy 

Faculty of Economics and Business Administration,
Department of Business Informatics and Operations Management,
Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium
{Aygun.Shafagatova, Amy.VanLooy}@UGent.be

Abstract. A successful implementation and adoption of Business Process Management (BPM) requires an alignment with other management areas in an organization, and specifically with practices related to Human Resource Management (HRM). While the BPM literature highlights the importance of aligning HRM appraisals and rewards to business processes, little is known about how organizations actually align these two areas in practice, and how they face the challenges of this alignment. For this purpose, we conducted ten explorative case studies to acquire empirical evidence and gain better insights on this issue. We uncovered four patterns of BPM-HRM alignment and determined their important components. This work discusses the critical factors that are important for a successful BPM-HRM alignment and provides recommendations for this alignment by differentiating between lower and higher levels of BPM maturity.

Keywords: Business Process Management · Process orientation · Alignment · Process-oriented appraisals · Process-oriented rewards · Case study

1 Introduction

In response to changing competitive environmental pressures, organizations are undergoing fundamental transformations in their structures and management systems [1]. The move to a horizontal process orientation is essential for organizations that are interested in breaking down barriers within the vertical structures, improving end-to-end communication for problem solving and increasing customer value [2]. In this regard, Business Process Management (BPM) has been emerged as a response to these calls to help organizations become more process-oriented.

While the technological aspects of BPM have drawn much attention, relatively limited work has been delivered regarding the people factor of BPM [3]. However, change in technology, processes, and structures is unlikely to yield long-term benefits without altering human knowledge, skills and behaviors [4]. In this respect, a better alignment between BPM and HRM practices is crucial if organizations intend to reap the full benefits of a process orientation in the long run. More specifically, employee appraisals and reward practices play an important strategic role to potentially influence

employee behavior and performance. Therefore, BPM should be integrated into those HRM practices since linking appraisals and rewards to business processes will create a better line of sight for employees to focus on behavior and performance that ultimately serve to process success. Otherwise, employees will not be interested in process goals if they are merely evaluated and rewarded for functional/individual goals, and process success will be sub-optimized in this way.

However, today’s holistic understanding of BPM as a management approach [3] doesn’t go further than merely recognizing the importance of aligning people and HRM aspects with business processes, without a deeper examination. In particular, the strategic issue of aligning employee appraisals and rewards is still an under-researched area without profound empirical evidence from practice. To fill this gap, our research question sounds:

RQ. How do organizations align employee appraisals and rewards with their business processes?

This research contributes to a holistic approach to BPM by gaining a better understanding and deeper insight into how organizations actually align their appraisals and rewards to process needs. To this end, we report on multiple explorative case studies to offer first-hand empirical evidence and provide details about our pattern-matching exercise that identified the most important components related to the alignment of appraisals and rewards with a process orientation.

The remainder is structured as follows. Section 2 discusses relevant concepts and related works from both the BPM and HRM literature. Section 3 specifies our research method. Next, the findings are presented (Sect. 4) and discussed (Sect. 5). Section 6 ends up with concluding thoughts.

2 Theoretical Background

2.1 HRM

Employee performance appraisal is “a variety of activities through which organizations seek to assess employees and develop their competence, enhance performance and distribute rewards” [5] [p. 473]. Performance evaluation is based on what people achieve and how they achieve it. Consequently, the HRM literature identifies the following dimensions of employee evaluation: (1) goals and objectives, and (2) competencies [6–8] (Fig. 1). Aligning employees with the organization’s strategic goals and values has become increasingly important as organizations struggle to gain or sustain a competitive advantage [9].

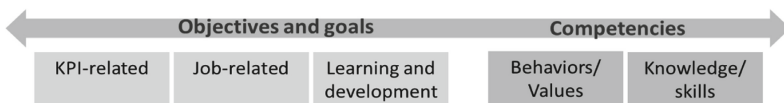


Fig. 1. Employee performance appraisal dimensions [23–25].

A reward system can be described as any conscious intervention within an organization aimed at encouraging or reinforcing required behaviors, or which compensates people for taking particular actions [4]. Typically, the reward strategies of organizations consist of two reward types: (1) financial (transactional) rewards and (2) non-financial (relational) rewards. [8, 10] (Fig. 2).

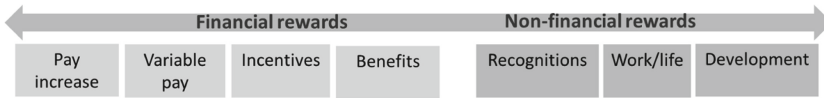


Fig. 2. Reward types [25, 27].

Employee appraisals and reward practices are grounded upon fundamental theories that shed light on divergent aspects such as motivation, goal-setting and self-determination [11–13]. Appraising and rewarding certain behaviors (and not others) has clear implications for performance. Thus, decisions about what is to be rewarded need to be made carefully and with attention to the overall business strategy [14].

The HRM literature recognizes the challenge that the existing traditional appraisals and reward systems do not fit within the context and needs of an organization that is downsized, delayed, dynamic, and diverse [1]. As a result, many organizations are rethinking their appraisals and reward strategies to better align them with the new business realities. Various studies have highlighted “reward and recognition” systems as one of HRM’s major critical success elements [4].

In this regard, the recent focus on strategic HRM (SHRM) emphasizes its important role in defining and achieving the strategic goals of an organization [15]. According to the contingency debate within SHRM, there should be a fit between HRM practices and an organization’s context and strategy in order to be successful and to contribute to organizational performance [16]. In this regard, however, no study was found that specifically covers this HRM fit with a business process context. Therefore, research on generic HRM appraisals, performance management and rewards (i.e. with its limited ‘one-size-fits-all’ approach) provides only a starting point to truly understand process-oriented appraisals and rewards.

2.2 BPM/BPO

BPM refers to a holistic management approach which focuses on the capabilities required to optimize process management practices within the organization [17]. Additionally, a Business Process Orientation (BPO) means focusing on business processes ranging from customer to customer, instead of placing emphasis on functional and hierarchical structures [18]. BPM and a process orientation have an effect on process speed improvements, increase in customer satisfaction, improvement of quality, reduction of costs, and improvement of financial performance [19].

However, only focusing on process design and process improvement is insufficient to ensure those benefits. If a redesigned process is embedded back into an existing functional organizational structure without other changes (e.g. no change in rewards

and evaluations), the prevailing managerial emphasis on functional concerns will continue, and process performance will eventually suffer [20]. The horizontal processes will pull employees in one direction, while the traditional vertical (appraisals and rewards) management systems will pull them in another, resulting in confusion, conflict of interests, and eventually undermining performance [21]. Therefore, an alignment with other management systems, and especially with HRM appraisals and rewards, is essential to eliminate confusion and conflicts, and to reinforce employees to work towards process success.

While the recent focus of BPM and BPO as a holistic management discipline encourages research in a broader manner, aligning HRM appraisals and rewards to business processes remains an under-researched area. The literature agrees on the fact that if organizations want to successfully embrace BPM and BPO, their way of appraising and rewarding employees should become more process-oriented [18, 22, 23]. [24] identified HRM appraisals and rewards as one of the sub-capabilities of a process-oriented culture. [25] studied the value dimension of culture by presenting CERT (customer focus, excellence, responsibility and teamwork) as main values supporting BPM, without covering other HRM aspects. To our knowledge, no academic research exists that investigates how these aspects are aligned. Previous BPO research highlighted the people aspects of a process orientation in broad terms, such as linking employee management to a process orientation [26], exploring competencies of BPM professionals [27], roles of chief process officers [28] and process owners [29], or individual process orientation aspects [30], most of which assume an alignment between employee goals and process goals. A small number of academic publications has addressed appraisals and rewards in the context of total quality management (TQM) [31], which is less specific for BPM.

Likewise, few authors from industry have discussed the importance of making changes in performance evaluations and rewards to avoid conflicts between business processes and functions [32, 33]. While [33] implicitly covered feedback (i.e. appraisals) and consequences of performance (i.e. rewards), it does not encompass all components and facets of how process-oriented appraisals and rewards should be. Several BPM Maturity models [34–38] also mention appraisals and rewards, albeit with high-level statements. In sum, given the increasing attention for BPM/BPO and for further contextualizing employee performance, an opportunity exists to establish research that combines the HRM and BPM disciplines with regard to appraisals and rewards.

3 Methodology

Given the study's explorative character, a qualitative research approach was employed. A detailed understanding is needed since little knowledge exists [39]. Moreover, the case study research method is particularly useful in information systems (IS) research when interest has shifted to organizational rather than technical issues [40]. [41] defines a case study as "*an empirical inquiry that investigates a contemporary phenomenon (the case) in depth and within its real world context*" [p. 5]. According to [42], case

study research allows to answer ‘how’ and ‘why’ questions by learning the state-of-the-art in practice to generate theory, and thus allowing insights into an emerging topic.

We conducted multiple case studies with different organizations in order to gain empirical evidence of how appraisals and rewards are actually being aligned with BPM/BPO. We employed ‘maximal purposeful sampling’ [40, 42] to select the case organizations in order to explore best practices. To this end, we contacted 27 potential candidate organizations via LinkedIn (of which ten positively responded) by searching for process-related management profiles (i.e. a Chief Process Officer, BPM manager or process owner), assuming that those organizations would have a higher process focus.

Table 1. Profile of the case organization and its representatives.

Coding companies	BPO maturity score	Sector	Size	Coding participants	Participant’s experience in company	Participants position level	Participants expertise
Company A	3	Production-pharma	>10000	repA1	3–5 years	Low Level Management	BPM
				repA2	<1 year	Mid Level Management	HRM
Company B	2,6	Retailwholesale	>10000	repB1	5–10 years	Mid Level Management	BPM
				repB2	15–20 years	Mid Level Management	HRM
Company C	4,5	ICT Services	5001–10000	repC1	10–15 years	Top Level Management	BPM
				repC2	10–15 years	Top Level Management	HRM
Company D	2,9	Banking	>10000	repD1	1–3 years	Mid Level Management	BPM
				repD2	1–3 years	Mid Level Management	HRM
Company E	3.9	Production-pharma	>10000	repE	3–5 years	Mid Level Management	BPM
Company F	3.4	ICT Services	>10000	repF	1–3 years	Mid Level Management	IT/BPM
Company G	2.9	Banking	>10000	repG	5–10 years	Top Level Management	BPM
Company H	2.5	Human health	501–1000	repH	1–3 years	Mid Level Management	BPM
Company I	3.6	Banking	>10000	repI	>20 years	Top Level Management	IT/HRM
Company J	3.4	Production-beverages	>10000	repJ	5–10 years	Mid Level Management	IT/BPM

Data collection was conducted in two iteration rounds, resulting in 14 interviews across ten organizations in total. The first round covered four organizations with both the BPM and HRM managers as interview respondents, while the second round included six organizations with only BPM-related managers as respondents. After the first round, we

felt that data from the HRM side was saturated, since the generic HRM practices that were used were all similar and confirming the HRM literature on practices, and we were specifically interested in more nuances about the process orientation side. More details about the case organizations are given in Table 1. The organizations and representatives are anonymized and coded with letters, while the rest of the data is real.

Most organizations are large in size and representatives are mostly from the top-level or mid-level management. The cases represent different sectors, such as banking (3), pharma (2), retail (1), ICT production and service (2), non-for-profit (1) and beverage production (1). Their BPM maturity varies between 2.4 and 4.5, on a 5-point Likert scale (i.e. in line with McCormack's [43] maturity assessment). This instrument has been repeatedly used in different studies and is a simple and effective way of measuring BPM maturity. The least mature case belongs to a non-for-profit healthcare organization, while the most mature case is an ICT production and service company.

3.1 Data Collection and Analysis

We used multiple sources of information [41]: semi-structured interviews, internal documents and online resources. Based on the BPM/HRM literature study, we developed semi-structured interview questions. An interview protocol was designed to guide the interview. The combination of both dimensions enabled us to depict the whole situation. For the BPM representatives, the interview questions focused more on process-oriented appraisals and rewards, while the HRM representatives were asked about generic performance management and reward practices. Nonetheless, both interviews were similarly structured: (1) general questions about the case organization and (2) the respondent, (3) BPO maturity assessment questions, followed by (4) semi-structured questions on appraisals and rewards (e.g. "Does a performance appraisal include process-related dimensions?", "Is an employee evaluated for the end-to-end Process KPIs that he/she executes? If yes, how? If no, Why not?") and (5) for BPM managers also open questions on how they align appraisals and rewards with business processes. Our explorative character enabled us to regularly refine and adjust the questions with newly acquired concepts. The interview duration was 45–60 min. Each interview was conducted face-to-face or via Skype and each respondent was interviewed once, with additional inquiries via email correspondence when needed. The respondents were also asked to provide relevant internal documents. In total, 129 pages of interview transcripts and 306 pages of internal and online documents were analyzed for the study. Thus, data triangulation was employed to gain a more nuanced understanding and to enhance reliability and validity [42]. The collected internal documents were related to HRM policies, the company strategy, together with online content. All data were analyzed for further examination.

Coding, pattern-matching and content analysis techniques [40] were used to analyze the data and to gain meaningful information [39, 42]. Coding was employed to categorize data (i.e. around concepts, key ideas or themes) by assigning labels (e.g. 'process owner appraisal', 'process improvement related') as units of meaning to data pieces. Since a qualitative data analysis is highly analytical and interpretive [42], we applied the NVivo coding tool. More specifically, we employed 'relational content analysis' [42, 44] since we were interested in the presence of certain concepts and in

examining how those (pre-defined or emergent) concepts are related to each other within the data [39], as well as presence of certain patterns across the cases. In total, 63 nodes were made to code the data in NVivo based on 14 interviews and 28 internal and online source documents.

3.2 Research Rigor

To ensure the rigor of our qualitative study, five strategies [39] were employed: (1) triangulation of data sources and of analysis methods to increase construct validity, (2) rich analytical descriptions through summary figures to ensure transferability and validity, (3) protocols (i.e. a case study protocol and interview protocol) to enhance reliability, (4) member checking with participants to validate the accuracy and credibility of the results, and (5) interview tapes and transcripts for further reliability. Thus, we followed a rigorous approach for data collection, data analysis and reporting to ensure accuracy [39]. Yin's [41] principles of data collection were also taken into account: (1) using multiple sources of evidence (triangulation), (2) creating a case study database, (3) and maintaining a chain of evidence (for reliability). The presence of a co-author also ensured accuracy via cross-analysis and reviewing the findings [40]. Finally, an analytical generalization through theory and analysis was done in the form of best practices, namely for understanding the case's complexity [39].

4 Results

We observed a subtle gradation in how far the case organizations have aligned their appraisals and rewards to their business process needs and BPM context, indicated in Fig. 3 by means of four patterns across three gradations.

The first gradation refers to an "implicit alignment", which means embedding process-related goals and objectives in the appraisals indirectly, and only for certain departmental or functional processes. This is especially done for joint process owner roles (i.e. a functional manager who also takes a process ownership role) and for employees who execute functional processes.

The second gradation category is "limited explicit alignment", which means explicitly aligning appraisals and rewards for a limited number of end-to-end processes. In this category, two roles are remarkable: a separate (or distinct) process owner role and the process improvement teams (i.e. teams that are responsible to improve processes).

The third category is "explicit alignment", meaning explicitly applying process-oriented appraisals and rewards across the company for process owners, process improvement teams and process executing employees (i.e. employees who perform a process).

Based on pattern-matching, we differentiated four patterns or groups among the case organizations (i.e. symbolized with geometrical figures in Fig. 3). The first group has only an implicit alignment for process owners or managers, while the second group has both an implicit alignment (i.e. on the process executing employee level) and a limited explicit alignment (i.e. on the process owner and process improvement team level). The third group of cases showed a limited explicit alignment on the level of

Organizations	Comp H	Comp A	Comp B	Comp D	Comp J	Comp I	Comp E	Comp G	Comp F	Comp C
BPO Maturity score	2.4	3	2,6	2,9	3.3	3.5	3.8	2.9	3.5	4,5
Implicitly aligned (embedded and within certain department)	PO	PO	PO	PE	PE	PE	PE			
Limited explicitly aligned (certain e2e processes or units only)			PO; PIT	PO; PIT	PO; PIT	PO; PIT	PO; PIT	PO; PIT; PE; PET	PE; PIT; PET; PO	
Explicitly aligned across company										PO; PE; PIT

PO - process owner; PE - process executor; PIT - process improvement team; PET - process executing team

Fig. 3. Identified patterns among case organizations with regard to the appraisal and reward alignment.

process owners, process executing employees and process executing teams (i.e. teams that perform the process). Finally, the fourth group covers a single case as being a best-in-class example which has explicit alignment across the entire company for process executing employees, process owners and process improvement teams. Figure 3 also shows that BPM maturity plays a role, ranging from lower (group 1) over lower-medium (group 2, group 3) to higher BPM maturity (group 4).

4.1 Process-Oriented Appraisals and Performance Dimensions

According to the cases, crucial components for obtaining process-oriented appraisals and rewards are: (1) level and role (i.e. for whom the appraisals and evaluations is done), (2) performance dimensions (i.e. what kind of process-related performance and behavior components are included in their evaluation), and (3) reward types (i.e. which rewards they get for the process results and their contribution). As shown in Fig. 4, we identified two main components and five sub-components that can be aligned with business processes. The “objectives and goals” component has three process-related sub-components, related to: (1) process KPIs, (2) process improvements, and (3) learning and development. The “competence and behavior” component has two sub-components: (1) process-supportive behaviors and values, and (2) process-related knowledge and skills. Figure 4 summarizes the process-related evaluation dimensions per case group.

The first group of cases includes some high-level process KPIs in appraisals, but mainly for a joint process owner role. Appraisals are only within the functional, departmental boundaries. In some situations, organizations belonging to this group can have process-related objectives, or some process-supportive behavioral competencies (e.g. continuous improvement initiatives) can also be an evaluation dimension.

The second group of cases displays more BPM-HRM alignment. It concerns an explicit alignment for process owners and process improvement teams. Most of the organizations in this group already have a separate process owner position (i.e. someone who is explicitly appraised for end-to-end process performance and improvement efforts, as defined in a job description). Cross-functional process improvement teams

(e.g. agile, lean, six sigma, supply chain teams) are usually responsible for process improvement projects, and are evaluated for reaching their improvement targets. Team-based appraisals in a process improvement context mostly happen informally as a project-based reflection that includes an evaluation of process KPIs or targets (e.g. in the six sigma control phase, or agile tribe and squad meetings (Case ID: i, j, d), in a post-implementation survey or in quarterly reviews of team KPIs (Case ID: e)).

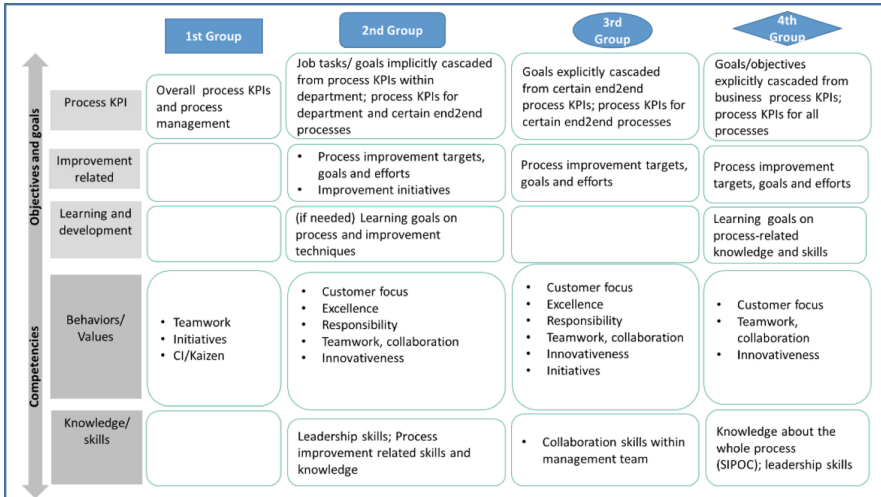


Fig. 4. Mapping of performance dimensions related to the process context of the case groups.

The appraisals of process executing employees are generally implicitly aligned, and largely depend on the department (i.e. managers decide whether to include process KPIs in evaluations or not). If included, process KPIs are mostly bounded to a department and thus internally embedded. All roles can have learning objectives regarding processes and process improvement skills. They can also be evaluated for process-supportive behaviors (e.g. customer focus, teamwork, excellence). For (Case ID: d, b), management decides whether or not to include process-related knowledge.

The third group of cases has a more explicit alignment of performance dimensions. Appraisals of employees executing certain end-to-end processes tend to include cascaded process goals. Process owners have more empowerment and are also strongly encouraged to collaborate with other process managers. Employee evaluations also contain process-supportive behavioral competencies. These organizations also have appraisals for certain end-to-end process executing teams (e.g. in monitoring and performance meetings). Process improvement teams are similar to the second group.

Finally, the fourth group has the highest BPM maturity, which also explains the highest BPM-HRM alignment across the company. Process executing employees have cascaded process KPIs, and are evaluated for SIPOC knowledge which is part of core competencies for each employee. SIPOC (i.e. supplier, input, process, output and customer) is aimed to rise process-awareness among employees. The observed issues

and problems in process performance are translated into opportunities (e.g. improvement opportunities in process performance, and development opportunities for employees). Informal feedback meetings are held regularly to discuss and evaluate team performance. BPM-supportive CERT values [25] are present in appraisals, such as accountability, continuous improvement/excellence, team work, and a customer focus.

Thus, throughout the different groups, the process-oriented components eventually become part of the overall employee appraisals and evaluations which can result in rewards like yearly pay increase or bonuses (Sect. 4.2).

4.2 Process-Oriented Rewards

Similar to the performance appraisals of Sect. 4.1, rewards can also be given through different reward and recognition programs. Figure 5 represents the observed process-oriented rewards. Respectively, reward practices in organizations can be aligned on two dimensions: financial (i.e. pay increase, bonuses and incentives linked to process performance and improvement) and non-financial rewards (i.e. recognitions, celebrations, praise, positive feedback for process success). We bundled the observed reward possibilities along with the identified case groups.

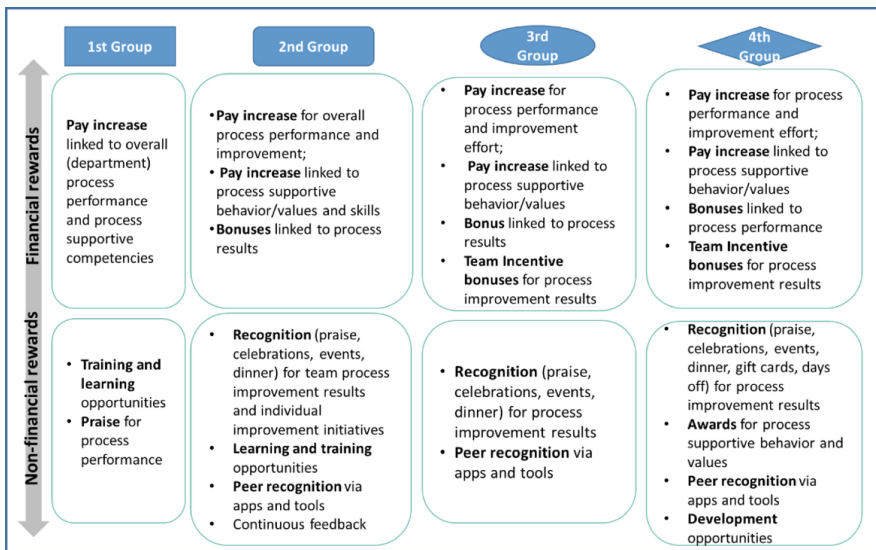


Fig. 5. Mapping of reward types related to the process context of the case groups.

The first group of organizations has the least aligned rewards. Joint process owners can get an annual pay increase as a result of yearly appraisals, which can also include process-related goals and competencies. Furthermore, process owners can get development opportunities related to process skills and praise for a good process operation.

Since most process owners are a separate role in the second group, they get explicitly rewarded with annual pay increase or bonuses, as well as recognitions for

process performance and improvement efforts. Process improvement teams in this group rather get non-financial recognitions in the form of celebrations, praise and events, but mainly without financial rewards. They also get opportunities to develop their process-related skills. Process executing employees are indirectly rewarded with pay increase as a result of their overall assessment, namely if their appraisals include process-related goal achievement, process-supportive competence or a behavior evaluation (i.e. rather embedded within departments). Peer recognition via apps and tools are also used to acknowledge a contribution to common goals. Some organizations (Case ID: b, d) have initiative programs that reinforce bottom-up improvement ideas, and recognize them once they are implemented.

The third group of cases displays a more explicit reward alignment. All above-mentioned financial and non-financial rewards can be given to process owners, process improvement teams and process executing employees, with a more direct link to processes. Employees have clear process-related goals and objectives, but limited to certain end-to-end processes and certain units. They can get a pay increase and bonuses, as well as non-financial recognitions for achieving their goals. Process improvement teams usually get short-term incentive bonuses and recognitions if their improvement targets are met.

The fourth group of cases applies all possible rewards across the organization for almost all business processes. Besides the above-mentioned rewards, they also organize corporate-wide award and recognition programs to reinforce company values, including process-supportive behaviors (e.g. teamwork, innovation, excellence). Process owners can get more frequent bonuses for process results and improvement efforts.

In general, mostly middle management takes decisions on what type of rewards can be given to employees and teams. Case organizations (Case ID: d, c, b, i) recognize the fact that if feedback and informal evaluations are becoming the new normal, then rewarding should also become more frequently and aligned with continuous evaluations. In this regard, non-financial and team-based rewarding will get more important because they are flexible, foster cooperation, and have longer-term effects.

Based on our observations, Fig. 6 summarizes the evolution of the process-related employee roles depending on the scope of BPM-HRM alignment. This evolution is also related to the BPM maturity levels.

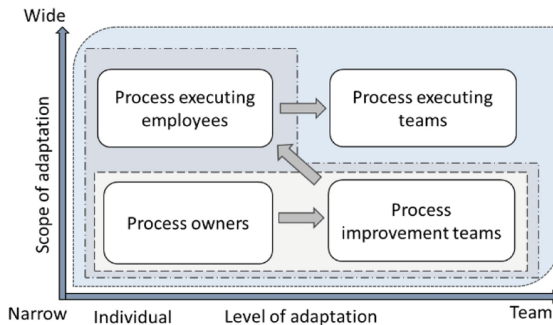


Fig. 6. Alignment scope linked to process-related roles.

Figure 6 shows that, when the scope of adaptation is narrow, only the process owners' appraisals and rewards are aligned with processes. It can grow to include process improvement teams to move from an individual level to a team level. When the scope of adaptation becomes wider, organizations also start aligning their process executing employees' appraisals and rewards to business processes (i.e. together with the two previously mentioned roles and levels). Ultimately, the scope of adaptation is at its widest point when appraisals and rewards exist on the level of process executing teams (i.e. together with the three previously mentioned roles of process owners, process improvement teams, and process executing teams).

5 Discussion

We summarized our findings on how the case organizations are aligning their appraisals and rewards with business processes in Figs. 4, 5 and 6. These figures are in line with generic HRM frameworks, while also elaborating on different process contexts. The literature of BPM maturity already positioned appraisals and rewards as a sub-capability of a process-oriented culture, which is now further enriched by the refinements of this study on process-oriented appraisals and rewards.

5.1 Critical Success Factors

The case interviews pointed us towards challenges and factors important for the success of aligning process-oriented appraisals and rewards. The respondents specifically mentioned the following five success factors.

- (1) The level of BPM maturity in an organization matters a lot. The basics of BPM should be in place before organizations might consider process-oriented appraisals and rewards. Examples of preconditions are that business processes should be defined, full-time process owners should be appointed, and a process performance measurement system should be in place (Case ID: a, h). According to (Case ID: b, a, d), having a joint process owner position is not effective: while it costs less for the organization, those managers do not fully gain the required process thinking nor fully engage with the process due to a lack of time. Mostly, those process owners also lack empowerment to influence other people.
- (2) An organization's culture can make or break the intended alignment. For instance, an individual mindset makes it more difficult to have team-based appraisals and rewards (Case ID: e), whereas a value-based mindset conflicts with evaluating process KPIs because behavior and values become more important in this case (Case ID: b). Resistance to change is another challenge, for which change management should be in place to facilitate a successful BPM-HRM alignment. Silo-thinking (Case ID: h, e, f) conveys an "it's not my problem" mentality in every department, which endangers cross-departmental collaboration and cooperation for process success.
- (3) Top management commitment and managerial support are crucial (Case ID: d, e, b, j, c). Most respondents agree on the fact that changing behavior is easier when

it is driven top-down. For instance, in (Case ID: c), top managers were a key driver to progress that far. Leaders as a team should drive collaboration across their areas of responsibility, leading to change in the employee’s attitudes and behaviors and in the way people work (Case ID: g). Furthermore, the existence of a Chief Process officer (CPO) role is equally important to drive changes (Case ID: g, c). In most cases, also the role of middle management was decisive for defining appraisals and reward procedures. Therefore, training middle managers on process thinking is crucial.

- (4) All process participants should be engaged and involved in process design, process improvement and KPI definitions. Process goals/targets should be communicated to all employees involved, and the KPIs should be cascaded to lower levels to create a process awareness and increase commitment from employees (Case ID: b, d). It is also important to achieve a mind shift to give/get horizontal feedback on process executions (Case ID: b).
- (5) For large and complex organizations (e.g. active across different countries and with complex legal systems), it can be hard to achieve standardization companywide and in all countries (Case ID: d, g, j, i, e). What in one country works well, does not necessarily work out in another country. Managers should acknowledge that the required BPM-HRM alignment can be achieved for certain units and employees, but not necessarily for everybody. Cascading process KPIs to the executing employee level in an effective way, seems very challenging for many cases (Case ID: d, b, e, f). A single employee does not have control over the entire business process, which can be demotivating if it is incorrectly imposed.

5.2 Recommendations Across BPM Maturity Levels

We now present recommendations for process-oriented employee appraisals and across lower and higher levels of BPM maturity (Fig. 7).

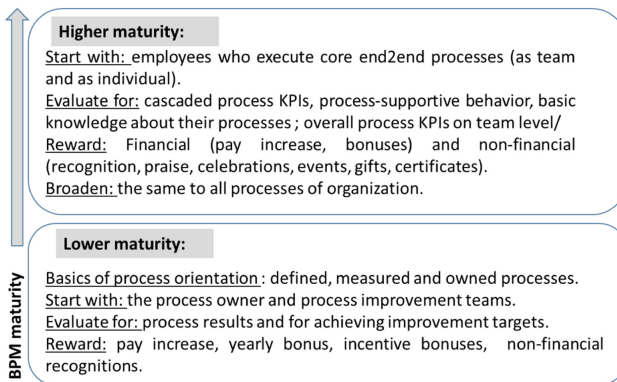


Fig. 7. Recommendations for aligning appraisals and rewards to lower and higher levels of BPM maturity.

If BPM maturity is low to medium, an organization should first make sure that the basics of BPM are in place, namely that its business processes are defined, measured and owned. In this case, the process owners and process improvement teams can be a starting point. These roles can be evaluated for process results and for achieving improvement targets. The evaluation of managers who are involved in cross-functional processes can include process-related goals and objectives, as well as behavioral competencies that support a process orientation (e.g. collaboration, change management). If an overall appraisal is tied to rewards, managers and process owners can get a pay increase or yearly bonus, as well as non-financial recognitions. Improvement teams can be rewarded with incentive bonuses (if possible) or with non-financial recognition (e.g. praise, celebration, dinner or gift cards).

In case of higher BPM maturity, organizations can go further in their alignment efforts by including process-related goals and objectives (i.e. mostly cascaded from end-to-end processes) within the evaluations of all employees who execute the core end-to-end processes. Furthermore, employees can be evaluated and rewarded for their process-supportive behavior, such as teamwork, customer focus, and continuous improvement initiatives. Process KPIs can also be defined on team level for the process executing teams, and they can be rewarded for achieving a better process performance. Continuous feedback and recognition on process performance can be given to those process teams. Afterwards, all business processes can be included in this BPM-HRM alignment. Basic knowledge about an organization's processes can be included in every employee's evaluation as being a core competence. Financial rewards (e.g. pay increase, bonuses) and non-financial rewards (e.g. days off, praise, celebrations, events, gifts, certificates) can be given to process executing employees and teams.

6 Conclusion

This exploratory case study approach has offered empirical evidence of how organizations align their employee appraisals and rewards to their particular business process context. We observed four patterns that organizations can use to ensure the required BPM-HRM alignment. We also presented an overview of the observed dimensions for process-oriented appraisals and rewards. Furthermore, we made suggestions for lower and higher levels of BPM maturity to improve the success of future alignment efforts.

Since this study intends to provide initial empirical insights, we acknowledge some limitations. The findings are based on a limited though reasonable number of cases and thus represent the experience of those organizations, albeit across different varieties of BPM maturity. Our study elaborates on the human aspects associated to the BPM discipline, and calls for a more extensive investigation with possibly a larger case sample. In future research, we will broaden our approach to end up with a comprehensive overview of possible types for process-oriented appraisals and rewards in the form of a managerial decision tool. Future research avenues can include linking and analyzing different HRM theories to an organization's BPM maturity level or to BPM practices specifically, as well as theorizing the findings with more quantitative methods and by also considering the individual employee perspective. Despite these limitations, the current work is relevant for practitioners who face challenges of misfit and

conflicting messages between their organization's process orientation and their traditional way of appraising and rewarding.

References

1. Agarwal, N.C.: Reward systems: emerging trends and issues. *Can. Psychol.* **39**, 60–70 (1995)
2. Nadarajah, D., Syed, A., Kadir, S.L.: Measuring business process management using business process orientation and process improvement initiatives. *Bus. Process Manag. J.* **22**, 1069–1078 (2016)
3. vom Brocke, J., Rosemann, M.: *Handbook on Business Process Management 1*. Springer, Heidelberg (2015). <https://doi.org/10.1007/978-3-642-45100-3>
4. Zairi, M., Jarrar, Y.F., Aspinwall, E., Practices, B.: A reward, recognition, and appraisal system for future competitiveness: a UK survey of best practices, pp. 1–19 (2010)
5. Fletcher, C.: Performance appraisal and management: the developing research agenda. *J. Occup. Organ. Psychol.* **74**, 473–487 (2001)
6. Armstrong, M.: *A Handbook of Human Resource Management Practice*. Kogan Page, London (2014)
7. Aguinis, H.: *Performance Management*. Pearson Education Limited, Harlow (2019)
8. Noe, R., Hollenbeck, J., Gerhart, B., Wright, P.: *Human Resource Management: Gaining a Competitive Advantage*. McGraw-Hill, New York (2014)
9. Boswell, W.R., Boudreau, J.W.: Employee line of sight to the organization's strategic objectives – what it is, how it can be enhanced, and what it makes happen. In: CAHRS Working Paper Series, p. 69 (2001)
10. Armstrong, M.: *Reward Management Practice: Improving Performance Through Reward*. Kogan Page, London (2010)
11. Herzberg, F.: Pinpointing what ails the organization. *J. Appl. Psychol.* **55**, 73–789 (1974)
12. Ryan, R., Deci, E.: Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *Am. Psychol.* **55**, 68–78 (2000)
13. Latham, G., Locke, E.: Goal-setting: a motivational technique that works. *Organ. Dyn.* **8**, 68–80 (1979)
14. Lawler, E.E.: Creating a new employment deal: total rewards and the new workforce. *Organ. Dyn.* **40**, 302–309 (2011)
15. Delery, J.E., Doty, D.: Modes of theorizing in strategic human resource management: test of universalistic, contingency and configurational performance predictions. *Acad. Manag. J.* **39**, 802–835 (1996)
16. Wood, S.: Human resource management and performance. *Int. J. Manag. Rev.* **1**, 367–413 (1999)
17. De Bruin, T., Rosemann, M.: Towards a business process management maturity model. In: Bartmann, D., Rajola, F., Kallinikos, J., Avison, D., Winter, R., Ein-Dor, P., et al. (eds.) *ECIS 2005 Proceedings of the Thirteenth European Conference on Information Systems*, Germany, 26–28 May 2005 (2005)
18. Willaert, P., Van den Bergh, J., Willems, J., Deschoolmeester, D.: The process-oriented organization: a holistic view. *Data Knowl. Eng.* **64**, 1–2 (2009)
19. Kohlbacher, M.: The effects of process orientation: a literature review. *Bus. Process Manag. J.* **16**, 135–152 (2010)
20. vom Brocke, J., Rosemann, M.: *Handbook on Business Process Management 2*. Springer, Heidelberg (2015). <https://doi.org/10.1007/978-3-642-45103-4>

21. Hammer, M.: The process audit. *Harv. Bus Rev.* **85**(4), 111–123 (2007)
22. Kohlbacher, M., Gruenwald, S.: Process orientation: conceptualization and measurement. *Bus. Process Manag. J.* **17**, 267–283 (2011)
23. Hammer, M., Stanton, S.: How process enterprises really work. *Harv. Bus Rev.* **77**(6), 108–118 (1999)
24. Van Looy, A., De Backer, M., Poels, G.: A conceptual framework and classification of capability areas for business process maturity. *Enterp. Inf. Syst.* **8**, 188–224 (2014)
25. Schmiedel, T., vom Brocke, J., Recker, J.: Which cultural values matter to business process management? *Bus. Process Manag. J.* **19**, 292–317 (2013)
26. Babic-Hodovic, V., Arslangic-Kalajdzic, M.: The influence of quality practices on BH companies' business performance. *J. Manag. Cases Spec. Issue* **14**(1), 305–307 (2015)
27. Müller, O., Schmiedel, T., Gorbacheva, E., vom Brocke, J.: Towards a typology of business process management professionals: identifying patterns of competences through latent semantic analysis. *Enterp. Inf. Syst.* **10**, 50–80 (2016)
28. Kratzer, S., Lohmann, P., Roeglinger, M., Rupprecht, L., zur Muehlen, M.: The role of the chief process officer in organizations. *Bus. Process Manag. J.* **25**, 688–706 (2018)
29. Danilova, K.B.: Process owners in business process management: a systematic literature review. *Bus. Process Manag. J.* (2018)
30. Leyer, M., Wollersheim, J.: How to learn process-oriented thinking: an experimental investigation of the effectiveness of different learning modes. *Schmalenbach Bus. Rev.* **65**, 454–474 (2013)
31. Waldman, D.A.: Designing performance management systems for total quality implementation. *J. Organ. Change Manag.* **7**, 31–44 (1994)
32. Harmon, P.: *Business Process Change*. Morgan Kaufmann Publishers, Burlington (2007)
33. Rummler, G., Brache, A.: *Improving Performance: How to Manage the White Space on the Organization Chart*. Jossey-Bass, San Francisco (2013)
34. OMG: *Business Process Maturity Model (BPMM)* (2008)
35. SEI: *CMMI for Services, Version 1.3* (2010)
36. Rohloff, M.: An approach to assess the implementation of business process management in enterprises
37. Fisher, D.M.: The business process maturity model a practical approach for identifying opportunities for optimization, pp. 1–7 (2004)
38. Harmon, P.: *Evaluating an organization's business process maturity executive* (2004)
39. Creswell, J.W.: *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*. Sage, Thousand Oaks (2007)
40. Myers, M., Avison, D.: An introduction to qualitative research in information systems. *Qual. Res. Inf. Syst.* **326**, 3–13 (2002)
41. Yin, R.K.: *Case Study Research: Design and Methods*. Sage Publishing, Los Angeles (2013)
42. Recker, J.: *Scientific research in information systems: a beginner's guide* (2013)
43. McCormack, K.P.: Business process orientation: do you have it? *Qual. Prog.* **34**, 51–58 (2001)
44. Krippendorff, K.: *Content Analysis: An Introduction to Its Methodology*. Sage Publications, Thousand Oaks (2004)



Business Process Improvement Activities: Differences in Organizational Size, Culture, and Resources

Iris Beerepoot^{1,2(✉)}, Inge van de Weerd², and Hajo A. Reijers^{2,3}

¹ ICTZ B.V., Hoorn, The Netherlands

² Utrecht University, Utrecht, The Netherlands

{i.m.beerepoot, i.vande.weerd, h.a.reijers}@uu.nl

³ Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract. Although there are many business process improvement (BPI) methods, organizations are struggling to apply them effectively. We answer to the call to focus more on the organizational context in BPI projects. We use workarounds – deviations from the prescribed way of using an information system – as a specific angle to approach BPI. In five healthcare organizations of different contextual types, we study workarounds and make recommendations for process improvements. Based on this explorative multiple-case study, we propose a set of contextual activities for each stage of a BPI project. Thereby, we shed light on the differences in tackling process improvements in organizations that differ in size, culture, and the availability of resources for BPI projects. We evaluate the completeness and expected adoption of the proposed contextual BPI activities by organizing two focus groups and conducting a survey.

Keywords: Business process improvement · Context-awareness · Workarounds

1 Introduction

Business Process Improvement (BPI) is on the agenda of many organizations since it has the potential to improve performance, including stakeholder satisfaction and process cost and time [1, 2]. Many methods for process improvement exist, albeit under different titles: process reengineering, improvement, and process innovation [3]. Despite the availability of many methods, actually improving a business process is not an easy endeavor. A problem that may be at the heart of this is that many BPI projects follow a “cookbook approach” that does not adapt to organizational context [4]. Vom Brocke et al. [4] join Benner and Tuschman [5] in claiming that the lack of *context-awareness* is the reason that many of such projects fail. A study by Denner et al. [6] shows that only one in three Business Process Management (BPM) methods takes organizational dimensions into account, which underlines this viewpoint. A number of methods do take account of size and cultural differences – specifically, whether or not the organizations are supportive of BPM – but this is yet a limited view on the range of

contextual factors that may be relevant. Additionally, none of the methods provide guidelines for both ends of the spectrum within these factors: e.g. for small start-ups and large multinationals [6].

We attempt to answer the call of multiple scholars [3, 7, 8] for more focus on context-awareness in BPM research and methods. We do so by focusing specifically on how to adapt BPI methods to the organizational context of the projects in which they are applied. Through our own work on the development and application of a specific BPI method, centered around “workarounds”, we had the opportunity to carry out five improvement projects. These projects have all taken place within the same domain, i.e. healthcare, which ensured that we could apply our improvement method in a very similar way across the cases. By identifying and addressing workarounds, we also gained an in-depth understanding of the processes in question and closely engaged with various stakeholders. At the same time, the organizational contexts of these projects differed to such an extent that we could study and identify relevant contextual factors. On the basis of the experiences we collected in these projects, we provide an answer to the following question: *depending on an organization’s context, which activities are essential in process improvement projects?* We identified the organizational contexts that are worthwhile to distinguish from each other and derived a set of essential improvement activities for each of these contexts. Throughout the paper, we will refer to these as *contextual BPI activities*.

The contribution of this work lies in our proposal of a list of contextual activities for each stage of an improvement project. These insights can help both researchers and practitioners to fine-tune their BPI method of choice. This may be beneficial to improve the success rate of the projects in which such a BPI method is applied. To ensure that our insights can indeed be transferred to and made specific for a wide range of BPI methods, we adopted the Stage-Activity framework by Kettinger et al. [9], which was recently extended by Gross et al. [3]. The framework identifies broadly recognizable stages in a BPI projects, as well as the typical activities that are carried out in these.

The structure of this paper is as follows. The next section contains an overview of the relevant literature. In Sect. 3, we describe our study’s research methods. We present our proposed contextual BPI activities in Sect. 4. In the evaluation section, Sect. 5, we reflect on the completeness of our proposal and investigate its expected adoption in practice. We end our paper with a discussion of the related work on contextual factors and improvement activities in the context of our study and present ideas for future work.

2 Related Research

2.1 Context-Aware Business Process Management

Schilit and Theimer first coined the idea of context-awareness in relation to computing [10], to describe software that adapts to the location in which it is used, as well as to the objects nearby. The concept was later adopted in the BPM area and used in the sense of modeling context-aware processes [e.g. 7, 11] and context-aware process mining [12]. Vom Brocke et al. [4] designated context-awareness as the first of ten principles of good BPM. They argued that awareness of contextual factors plays a major role in the success

of a BPI project and should be taken into account in relation to BPM methods. We attempt to answer the call of multiple authors [3, 7, 8] for more focus on context-awareness in BPM research and methods. The organizational factors from the framework by Vom Brocke et al. [8] and the activity framework by Gross et al. [3] form the basis for our proposal. From the extant literature, Vom Brocke et al. derive a set of contextual factors relevant for BPM. They distinguish four dimensions: goals, processes, organizations, and environments. As we are especially interested in the differences in types of organizations, we focus on the organization dimension. The organization dimension includes the factors scope, industry, size, culture, and resources.

Gross et al. [3] built on the Stage-Activity framework by Kettinger et al. [9]. Kettinger et al. distinguished six stages in Business Process Reengineering projects: (1) envision, (2) initiate, (3) diagnose, (4) redesign, (5) reconstruct, and (6) evaluate. They proposed a set of activities to be executed during each stage. Gross et al. [3] extended this framework with several more contemporary activities. In this study, we highlight from Gross et al.'s BPI activities the essential ones for each stage, depending on the contextual factors of an organization.

2.2 Workarounds as a Source for Business Process Improvement

In BPM literature, workarounds are often discussed in the context of users of process modeling languages, such as BPMN, inventing alternative ways of modeling processes [13–16]. Studies in other research domains discuss workarounds enacted by end users of ISs in general, or specific types of ISs such as Health Information Systems (HISs). They are often described as a form of appropriation [20] and a response to blockages [16], rigid constraints [17], or a misalignment between design and practice [18]. Fortunately, there is a positive side to workarounds. By acknowledging them, instead of ignoring them, organizations can perform corrective actions and improve their work systems [17, 18]. In earlier work, we developed the *Workaround Snapshot Approach* for identifying, analyzing and addressing workarounds in organizations, in order to achieve work system improvement [19]. We use this approach as a context for studying the role of organizational dimensions in improvement projects and to derive a set of contextual activities.

3 Methods

In this study, we explore how process improvement is to be tackled within different organizational contexts. We followed an explorative multiple-case study approach to identify contextual factors that influence the choice of activities in process improvement projects. The multiple-case study approach enabled us to investigate a contemporary phenomenon in its real-world context [20]. Furthermore, it allowed us to recognize general patterns in different settings [21] and to increase the external validity of our insights [20]. We assessed the completeness of these contextual factors and activities by engaging with two focus groups. Finally, we carried out a questionnaire to evaluate the adoption of the contextual activities in future process improvement projects.

3.1 Case Selection

We investigated five different organizations. Because the goal of our study is to replicate findings across cases [20], we chose our cases from one sector: healthcare. Focusing on organizations in one sector made it easier to compare the cases, as several variables (industry, scope) remained constant. In the healthcare sector, optimal process support is particularly important since care processes transcend departments [22] and are less predictable than industrial processes [23]. Table 1 presents an overview of the five case organizations we studied. All organizations use the same HIS, which is used for managing information related to patient records, patient logistics, and other administrative data. Although all organizations are from the same sector, they have several distinctive characteristics in terms of organization type, size, culture, etc.

Table 1. Overview of case organizations and their characteristics.

Case	Type	Department	Size	Culture	Resources
A	General hospital	Orthopedics and surgery	Medium	Flat	Average
B	District hospital	Urology and cardiology	Large	Hierarchical	Many
C	District hospital	Urology and pulmonary	Large	Hierarchical	Many
D	Specialized center	Rehabilitation	Small	Flat	Few
E	Specialized center	Rehabilitation	Small	Flat	Few

3.2 Data Collection

Data collection was performed by the first author of this paper and took place between April 2017 and August 2018. As presented in Table 2, data were collected via observations of caregivers, unstructured interviews with the observed caregivers, and semi-structured interviews with team leads, IT managers, and HIS experts. By using these multiple sources of data we enhanced the reliability of our analysis [21].

Table 2. Overview of data collection techniques and informants.

Type	Amount	Informants	Collection
Observations and unstructured interviews	16 (106 h)	Caregivers: physicians, nurses, office secretaries, clinical secretary, physician assistant, team lead, therapists	Field notes
Semi-structured interviews	22 (24 h)	Team leads, information architect, HIS experts, IT managers and coordinators, care administration employee	Recorded and transcribed

3.3 Data Analysis

We analyzed our data in several iterations. First, we conducted a *within-case analysis* of each of our case organizations. We reduced and made sense of the collected data by structuring our interview transcripts and field notes in 51 workaround snapshots. These

snapshots capture a description of the workaround, the roles involved, a process model, an illustration of the impact on the existing process, the motivation of the user to enact the workaround, and an advice on how to use the snapshot as a basis for BPI in the organization [19]. This advice was based on the interviews with caregivers and HIS experts.

In our *across-case analysis*, we compared our workaround snapshots with the activity framework of Gross et al. [3]. Furthermore, we analyzed for each case organization the corresponding contextual factors from the framework by Vom Brocke et al. [8]. As the scope and industry of our cases were all equal – intra-organizational and healthcare sector – we focused on the differing contextual factors in size, culture, and resources. We collected information about those three contextual factors (presented in Table 1) from the caregivers and experts. Finally, for each type of context, we prioritized the most important activity for change. Figure 1 illustrates the methodological framework of our case study by showing how our within-case analysis and across-case analysis are connected. The result of our case analysis was a matrix containing activities for BPI linked to contextual factors.

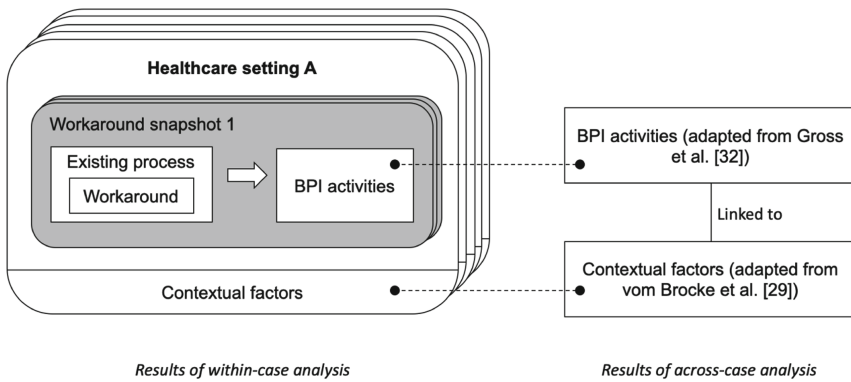


Fig. 1. Methodological framework of the multiple-case study.

3.4 Evaluation

We evaluated the case study results through two focus groups and a questionnaire. The participants in these evaluations were all employees of the company that implemented the HISs in the five case organizations. In addition to their current role as HIS consultant or developer, most of the participants also had an extensive background in the healthcare industry (e.g. as nurse or IT manager in a hospital). Table 3 provides an overview of the participants involved in the evaluation.

The goal of the *focus groups* was to test the *completeness* of the found BPI activities and their linked contextual factors. We organized two focus groups of six and two participants; HIS experts with extensive experience in healthcare organizations. We presented our BPI activities and contextual factors to the participants and asked

them to evaluate these. We encouraged them to propose different contextual factors or activities if they did not fully agree.

The goal of the *questionnaire* was to discover whether possible end users intended to follow our proposed prioritization of BPI activities in their role as process change agents in new encounters. We developed a questionnaire based on Moody's method evaluation model [24] to evaluate our proposed set of activities on ease of use, usefulness, and intent to use. The questionnaire was administered to three HIS consultants. We first explained our proposal of factors and activities in detail and then let them answer the questions.

Table 3. Evaluation participants (BC, FC, TC = Business, Functional, Technical Consultant).

Evaluation part	Occupation	Previous occupation(s)	Years in healthcare
Focus group 1	Manager Business Improvement	IC nurse & head of IT (in hospital)	33
Focus group 1	Senior BC	IT developer (in hospital)	15
Focus group 1	Business Developer	Account manager	9
Focus group 1	Team Lead TCs	Senior TC	7
Focus group 1	Account Manager	N.a.	4
Focus group 1	CISO & Service Delivery Manager	N.a.	4
Focus group 2	Team Lead FCs	Senior FC	12
Focus group 2	Product Owner & Senior FC	N.a.	3
Questionnaire	Team Lead BCs and Senior BC	Nurse & Senior IT Advisor (in hospital)	29
Questionnaire	Senior BC	Nurse & Manager IT (in hospital)	30
Questionnaire	Junior BC	N.a.	1

4 Contextual BPI Activities

Based on our interviews and observations with HIS users and experts in five organizations that differ in context, we derived a set of contextual BPI activities and discuss these in detail in the following sections.

4.1 Envision

What is evident from our observations and interviews is that the identification of workarounds and the development of snapshots needs to be preceded by a set of preparation activities. For all types of organizations, it is essential to identify the process stakeholders and boundaries. What distinguishes the different contexts is the culture factor, specifically in terms of hierarchy. For hierarchical settings, it appeared to

be especially important to establish management commitment and adhere to their vision. In case C, not all managers were sufficiently included in the pre-stages of the project. As a result, the improvement project was discontinued. On the other hand, in non-hierarchical settings such as case A, D and E, it appeared much more important to evaluate the existing culture when starting a BPI project. In such organizations, although change procedures are often undocumented, there are implicit procedures in place. When these procedures are not adhered to, resistance from staff can be expected. Table 4 presents the contextual BPI activities for the first stage.

Table 4. Contextual BPI activities in the envision stage.

Dominant factor	Values	Activities
Culture	Hierarchical	Establish and adhere to management commitment and vision
	Flat	Evaluate existing culture and implicit procedures
	All	Identify process stakeholders and identify process boundaries

4.2 Initiate

The findings presented in the envision stage highlight the importance of gaining commitment from staff – either through establishing commitment from management or through adhering to implicit procedures. In the initiation stage, gaining commitment from all those involved only becomes more important. If the staff is not committed, the diagnosis stage will be unsuccessful. Different types of organizations can be distinguished in this stage by their size. In large organizations we experienced the importance of defining ownership: during our research in case B, we discovered that another group within the organization felt they were assigned the task of improving the process in question. This could have been prevented by establishing ownership in the initiation stage of the improvement project. In smaller organizations it is less likely that two groups are working on the same task without them knowing about each other. In such organizations, it has turned out to be much more important to inform stakeholders of the initiation of the improvement project, giving them a chance to express their interest in the project and their willingness to contribute. Furthermore, in these organizations it is much more manageable to include the larger part of the stakeholders involved than in larger organizations. Table 5 shows the contextual activities related to this stage.

Table 5. Contextual BPI activities in the initiate stage.

Dominant factor	Values	Activities
Size	Large	Define ownership
	Small	Inform stakeholders
	All	Gain staff commitment

4.3 Diagnose

In the diagnosis stage, we again use size to distinguish different contexts, as presented in Table 6. Larger organizations allow for comparison of processes and workarounds over different departments within the same organization. A team lead described a specific workaround used in the urology department in case C and recalled the use of a similar workaround in the cardiology department. Such settings allow for benchmarking comparable processes in different departments of the same organization. In smaller organizations such as medical rehabilitation centers, there are seldom similar processes to compare with. What is more common in such organizations is to organize discussion meetings with similar organizations that encounter the same obstacles. A solution found by one can sometimes be directly implemented by another. For example, medical rehabilitation center D used an open source tool developed for autistic children to create daily schedules for rehabilitants with neurological damage. In medical rehabilitation center E, they used Microsoft Word to make such schedules. One of the recommended actions captured in the snapshot was for organization E to use the same tool as it was much more efficient. What appears to be important for all types of organizations is to obtain quantitative data on processes using techniques such as process mining. Diagnosis is currently most often done qualitatively, using a small sample. Analysis of a larger data set would allow for a more complete diagnosis of inefficient processes.

Table 6. Contextual BPI activities in the diagnose stage.

Dominant factor	Values	Activities
Size	Large	Benchmark process from within company
	Small	Benchmark process from competitors
	All	Obtain quantitative process data, e.g. via process mining

4.4 Redesign

During the redesign stage, we found that it is important for all organizations to estimate the required resources and organizational change needed. Only when this is done, it can be decided whether to move forward with the redesign. Not making a thorough estimation of the required resources and organizational change can endanger the continuity of the improvement process and can result in the loss of staff commitment. High-resource organizations making a significant investment in process improvement will also need to develop an elaborate improvement plan on top of this estimation in order to make the most of their investment. Organizations with a smaller budget will benefit from utilizing their stakeholders' knowledge of the process in coming up with improvement ideas in order to save resources. Moreover, having the stakeholders contribute improvement ideas often raises their engagement with the improvement project. Table 7 shows the contextual activities related to this stage.

Table 7. Contextual BPI activities in the redesign stage.

Dominant factor	Values	Activities
Resources	Many	Develop detailed improvement plan
	Few	Collect improvement ideas from stakeholders
	All	Estimate required resources and organizational change needed

4.5 Reconstruct

In Table 8, we present the contextual activities related to the reconstruct stage. We noticed in our case organizations the many consequences process changes can have on other processes. In smaller organizations, these consequences can be easily overseen. However, in larger organizations, the potential impact of changes on other processes need to be analyzed in order to prevent harmful consequences. We also experienced a certain ‘change fatigue’ in these larger organizations. Participants were frequently confronted with new change programs, receiving many communications on what was happening and what they needed to change in their work practices. In smaller organizations, stakeholders constantly reminded the interviewer that they wanted to be involved in any process changes. We therefore recommend smaller organizations to emphasize the communication of any information related to the improvement project, whereas we recommend larger organizations to hold back on heavy communication. For both types we see the importance of integrating process changes into existing processes. If not, keeping up with process changes will become unmanageable for process stakeholders.

Table 8. Contextual BPI activities in the reconstruct stage.

Dominant factor	Values	Activities
Size	Large	Analyze potential impact for other processes
	Small	Communicate process changes
	All	Integrate process

4.6 Evaluate

Building on the previous stage, we again make the distinction between different sizes, as described in Table 9. As larger organizations often have other improvement programs running, we suggest they should look for opportunities to link individual process improvement activities to existing programs. Doing so will hopefully decrease the change fatigue that participants are experiencing in these organizations. As mentioned in the previous stage, we found that participants in smaller organizations would like to be more involved and would like to hear about any outcomes of process changes. We therefore recommend smaller organizations to emphasize the communication of these outcomes to stakeholders. The importance of monitoring the changing environment and processes applies for all types of organizations. Processes and workarounds are always in flux and need to be monitored over time.

Table 9. Contextual BPI activities in the evaluate stage.

Dominant factor	Values	Activities
Size	Large	Link to continuous improvement programs
	Small	Report key process change outcomes
	All	Monitor environment for future needs to change

5 Evaluation

To evaluate the completeness of the contextual activities and the expected adoption of our proposal in practice, we organized two focus groups and distributed a questionnaire among potential end users of the method.

5.1 Completeness of the Contextual Factors

During one of the focus groups, an interesting discussion on the organizational factors of healthcare organizations arose. One of the critical notes was that, in the future, the amount of beds in hospitals would not be a valuable indicator of size, since healthcare is moving more and more towards home care. Looking at revenue and number of employees would give a more realistic view of the size of these organizations.

Another proposal made in the focus group was to add the contextual factor of maturity. Some organizations are more mature than others, for example by having procedures in place to address problems and knowledge present to bring HIS projects to successful completion. It was mentioned that in mature organizations, it would be possible to focus more on quality and patient satisfaction. In contrast, immature organizations need to focus on solving problems and getting their processes in order in the first place. However, participants in the focus groups acknowledged that it would be difficult to categorize organizations into a scale of maturity and the organizations themselves might be inclined to make misjudgments as to how they fare on the ladder. Moreover, many examples were given of small organizations that are in some aspects very mature and big organizations being surprisingly immature on some levels. This shows that it would be difficult to define simplified profiles, such as big, mature organizations and small, immature organizations. Doing so, we would exclude many organizations. The other three factors – size, culture and resources – are often inter-dependent. Most big organizations are hierarchical and have more resources than the more flat and smaller organizations, with some exceptions. In Table 10 we summarize the focus group's evaluation of the proposed context factors.

Table 10. Summary of the evaluation of contextual factors.

Opposed	Confirmed	Proposed additions
Operationalization of size: number of beds	Culture: flat or hierarchical	Operationalization of size: revenue and number of employees
	Resources: many or few	Maturity: mature or immature

5.2 Completeness of the Contextual Activities

Focusing on the activities of the method, some possibly missing ones were noted. First, the importance of the activities ‘realize need for change’ in the envision stage and ‘outline key measurement variables’ in the improvement stage were stressed by the focus group participants. This would apply for all types of organizations. The key measurement variables would then need to be evaluated in the evaluate stage. Such an activity is not included in the list of Gross et al. [3], although the activity ‘evaluate process performance’ comes close. Another activity considered important for all types of organizations in the evaluate stage is also not in the list of activities, namely ‘solicit feedback’. This activity is listed in the improvement stage but is considered even more important in the evaluate stage according to the participants.

The participants also mentioned that – apart from the distinction in *which* activities to perform depending on context, which they mostly agreed on – a distinction can be made in *how* to perform certain activities. For example, when performing the activity ‘analyze existing process’ during the diagnosis stage, the way the data is collected differs depending on the type of organization. In a small medical rehabilitation center with only two secretaries at the front desk, the means of data collection and communication of process changes would differ considerably from a big hospital with sixty to seventy secretaries at multiple front desks.

What was evident both from our experience in looking at workarounds in the five cases and from the participants’ experience in other healthcare organizations, many users of HISs experience a significant level of change fatigue. Especially caregivers in bigger organizations have participated in several reorganizations and process improvement programs. It is therefore important to prioritize process changes; to not only gain their commitment but also to retain their commitment, by soliciting feedback when necessary and by feeding back the results they helped achieve. In Table 11 we summarize the focus group’s evaluation of the contextual BPI activities.

Table 11. Summary of the evaluation of activities.

Opposed	Confirmed	Proposed additions
None	All	Realize need for change (stage: envision)
		Outline key measurement variables (stage: redesign)
		Evaluate process performance (stage: evaluate)
		Solicit feedback (stage: evaluate)

5.3 Expected Adoption of Our Proposal in Practice

The questionnaire on the ease of use, usefulness and intent to use of our proposed set of contextual BPI activities was completed by two senior business consultants (one of whom was also the team lead of the business consultancy team) and one junior business consultant. We scored the answers from 1 to 5 (e.g. for statement #1: strongly disagree = 1 and strongly agree = 5). Note that the scores on the negatively worded statements #4, #7, and #9 need to be inversed for a correct interpretation.

Our proposal is considered easy to understand and use (average of statements #1 through #4 = 3.75), although for those spending little time in the concerning organizations it may be difficult to apply in practice. Moreover, it is considered useful (statements #5 through #8 = 3.75 on average) but does not necessarily make it easier to perform BPI projects than other methods. The intention to use the ideas we proposed is high (statements #7 and #8 = 4.0 on average). The full results are depicted in Table 12.

Table 12. Results from the questionnaire on ease of use, usefulness and intent to use.

#	Statement	Strongly disagree (1)	Dis-agree (2)	Neutral (3)	Agree (4)	Strongly agree (5)
1	In general, the method seems to be well-applicable	0	0	0	3	0
2	It seems easy to learn the method	0	0	0	2	1
3	I find the stages and activities of the method clear and easy to understand	0	0	0	3	0
4	I am not confident I can apply the method in practice	0	1	1	0	1
5	I believe that this method can improve the work practices of HIS users	0	0	1	1	1
6	This method makes it easier for me to tackle improvement projects in healthcare organizations	0	0	2	1	0
7	I find other improvement methods more useful than this method	0	1	2	0	0
8	In general, I find this method useful	0	0	0	2	1
9	I would definitely not use this method to improve the use of HISs in healthcare organizations	1	2	0	0	0
10	I intend to use this method in future projects	0	0	1	2	0

6 Discussion

In this study, we argued that the essential BPI activities differ for organizations of varying size, culture, and resources. For each stage in a BPI project, we pointed out the dominant factor to distinguish organizations and suggested the corresponding contextual activities. In the following, we discuss related work on contextual factors and BPI activities and their relation to our study.

In four of the six stages of BPI projects, we found size to be the dominant factor in determining BPI activities for an organization. The importance of organizational size has been noted in several other studies. For example, a large firm size appeared to be the largest contributor to Total Quality Management success after industry type [25]. Similarly, Shah and Ward [26] studied the role of organizational context in lean manufacturing and concluded that plant size was the largest influencer in the likelihood of implementing lean practices. In IT innovation studies, organizational size also has been considered an important predictor of IT innovation adoption [27]. Our study complements these findings by suggesting that size is also an important factor in another way, namely in distinguishing which activities should be carried out during BPI projects.

The second contextual factor we studied was culture. As Schmiedel et al. [28] state: “bluntly put, BPM initiatives often fail for cultural reasons”. Culture has been argued to be an important factor in BPM. BPM is often more successful when cultural values are high [29]. Moreover, the success of BPM methodologies depends on the culture of an organization. Thiemich and Puhlmann [30] for example, argued that an organization open for change benefits from the use of agile methodologies, while a continuity-valuing organization might benefit more from using traditional methods [8]. The difference in suitable management styles in organizations varying in culture has also been noted by Donaldson [31]. The latter also mentioned that size and culture are linked in this respect. Bureaucracy and hierarchy are often more suitable in bigger organizations than in smaller ones. Our results confirm these insights: we found the hierarchical culture of an organization an important factor in determining the pivotal activities in BPI projects.

The third contextual factor that we looked into was resources. This factor has received less attention in BPM studies than size and culture, but our study suggests that it is nonetheless an important aspect to consider in BPI projects. In the context of open process innovation, Niehaves [32] studied the role of personnel resource scarcity. He found that BPM outcomes are affected by personnel scarcity as it decreases customer involvement. Several authors have mentioned the importance of stakeholder involvement for improving processes, also in the context of workarounds. Wheeler et al. [33], for instance, state: “in the case of workarounds, organizations could capitalize on the mindfulness of employees by encouraging employees to share their workarounds in order to improve task design”. It is believed that insights from users can guide system design [34, 35] and decrease resistance towards the system [36, 37]. In other words, even though previous studies have touched on this topic, our study puts the resources factor firmly on the map as an important contextual factor.

In the evaluation of our proposal, another contextual factor was raised: maturity. In the BPM literature, several studies have distinguished the difference between mature and immature organizations. For example, Reijers et al. [38] argued that “BPM projects are performed in a more systematic manner in larger and more mature organizations”. Similarly, according to Burlton [39], “the more mature the organization is with regard to BPM, the more sophisticated their process governance framework and their commitment to it”. Ravesteyn and Jansen [40] went a step further and proposed a situational BPMS implementation method that uses an organization’s maturity level to configure the activities that should be executed. In our study’s evaluation it was mentioned that immature organizations need focus on improving existing processes – called *exploitation* [41] – while more mature organizations can move beyond their existing processes and focus on *exploration*. However, we recognize that most current organizations focus on exploitation and are not yet ready to move towards exploration [3]. Additionally, we found that it was difficult to assess the healthcare organizations of our study as either mature or immature. BPM maturity models such as the one by Rosemann [42] might be of help to operationalize the contextual factor maturity.

Until now, we discussed the different contextual factors separately. However, the factors size, culture, and resources are tightly linked. Most larger organizations have a hierarchical structure and more resources than the smaller and flatter organizations, with some exceptions. This finding of interdependency of contextual factors supports statements by several others [8, 26, 43, 44].

Our study does have limitations. The data collection was performed by one researcher only. However, we did collect data in multiple ways and have performed different methods of evaluation (including a quantitative survey) in order to make sure subjective views did not cloud the findings too much. Moreover, we proposed contextual BPI activities based on an intensive case study of five organizations, all of which in the healthcare sector, which provided a meaningful set for comparison. The small number of cases and the sole industry makes generalization difficult. Therefore, we extensively evaluated the proposed activities, leading to a number of clues for where our proposal might fall short in generalizability. Future studies may reveal whether our proposal would be applicable in other industries.

7 Conclusion and Future Work

In this study, we attempted to identify which activities are essential in improvement projects depending on organizational size, culture, and resources. We used a multiple-case study approach to discover how improvement is to be tackled in organizations of different contexts. We focused specifically on organizations in the healthcare sector, although findings may be generalizable to other sectors as well. We proposed a set of contextual activities for each stage in process improvement projects and evaluated our proposal on multiple levels. The evaluation revealed several points of departure for further refining our proposal. (1) In addition to size, culture, and resources, the maturity of an organization may be an important factor in tackling improvement projects. (2) The contextual factors size and maturity need to be further operationalized. For example, in the future, distinguishing healthcare organizations using number of beds

will become irrelevant, as most of the care will be brought to the home. (3) In addition to defining the essential activities for each organizational context, we might also make a distinction in the way in which an activity is performed. (4) The stakeholders in improvement projects may experience a high level of change fatigue, which will need to be taken into account when tackling improvement projects in organizations.

In general, our proposal for the identification of contextual factors is considered relatively useful and easy to understand, although it may not be easy to apply for all. The intention to use the ideas is high among the three participants we involved in the questionnaire. Future work may look into the role of an organization's maturity in identifying contextual improvement activities. It may also focus on evaluating our proposal for sectors other than healthcare.

References

1. Altinkemer, K., Ozcelik, Y., Ozdemir, Z.D.: Productivity and performance effects of business process reengineering: a firm-level analysis. *J. Manag. Inf. Syst.* **27**, 129–162 (2011)
2. Vanwersch, R.J.B., et al.: A critical evaluation and framework of business process improvement methods. *Bus. Inf. Syst. Eng.* **58**, 43–53 (2016)
3. Gross, S., Malinova, M., Mendling, J.: Navigating through the maze of business process change methods. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019)
4. Vom Brocke, J., Schmiedel, T., Recker, J., Trkman, P., Mertens, W., Viaene, S.: Ten principles of good business process management. *Bus. Process Manag. J.* **20**, 530–548 (2014)
5. Benner, M.J., Tushman, M.L.: Exploitation, exploration, and process management: the productivity dilemma revisited. *Acad. Manag. Rev.* **28**, 238–256 (2003)
6. Denner, M.-S., Röglinger, M., Schmiedel, T., Stelzl, K., Wehking, C.: How context-aware are extant BPM methods? - development of an assessment scheme. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *BPM 2018*. LNCS, vol. 11080, pp. 480–495. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_28
7. Rosemann, M., Recker, J.C., Flender, C.: Contextualisation of business processes. *Int. J. Bus. Process Integr. Manag.* **3**, 47–60 (2008)
8. vom Brocke, J., Zelt, S., Schmiedel, T.: On the role of context in business process management. *Int. J. Inf. Manag.* **36**, 486–495 (2016)
9. Kettinger, W.J., Teng, J.T.C., Guha, S.: Business process change: a study of methodologies, techniques, and tools. *MIS Q.* **21**, 55–80 (1997)
10. Schilit, B.N., Theimer, M.M.: Disseminating active map information to mobile hosts. *IEEE Netw.* **8**, 22–32 (1994)
11. Ploesser, K., Recker, J.C.: Context-aware methods for process modeling. In: *Business Process Modeling: Software Engineering, Analysis and Applications*, pp. 117–134. Nova Publishers (2011)
12. Günther, C.W., Rinderle-Ma, S., Reichert, M., Van Der Aalst, W.M.P., Recker, J.: Using process mining to learn from process changes in evolutionary systems. *Int. J. Bus. Process Integr. Manag. Spec. Issue Bus. Process Flex.* **3**, 61–78 (2008)
13. Recker, J.: Opportunities and constraints: the current struggle with BPMN. *Bus. Process Manag. J.* **16**, 181–201 (2010)

14. Hahn, C., Recker, J., Mendling, J.: An exploratory study of IT-enabled collaborative process modeling. In: zur Muehlen, M., Su, J. (eds.) BPM 2010. LNBIP, vol. 66, pp. 61–72. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20511-8_6
15. Puhlmann, F., Weske, M.: Investigations on soundness regarding lazy activities. In: Dustdar, S., Fiadeiro, J.L., Sheth, Amit P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 145–160. Springer, Heidelberg (2006). https://doi.org/10.1007/11841760_11
16. Weber, I., Haller, J., Mulle, J.A.: Automated derivation of executable business processes from choreographies in virtual organisations. *Int. J. Bus. Process Integr. Manag.* **3**, 85 (2008)
17. Lalley, C., Malloch, K.: Workarounds: the hidden pathway to excellence. *Nurse Lead.* **8**, 29–32 (2010)
18. Safadi, H., Faraj, S.: The role of workarounds during an opensource electronic medical record system implementation. In: ICIS 2010 Proceedings, St. Louis (2010)
19. Beerepoot, I., Van De Weerd, I.: Prevent, redesign, adopt or ignore: improving healthcare using knowledge of workarounds. In: ECIS 2018 (2018)
20. Yin, R.K.: *Case Study Research and Applications: Design and Methods*. Sage Publications, Thousand Oaks (2017)
21. Eisenhardt, K.M.: Building theories from case study research. *Acad. Manag. Rev.* **14**, 532–550 (1989)
22. Lenz, R., Reichert, M.: IT support for healthcare processes. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 354–363. Springer, Heidelberg (2005). https://doi.org/10.1007/11538394_24
23. Reijers, H.A., Russell, N., van der Geer, S., Krekels, G.A.M.: Workflow for healthcare: a methodology for realizing flexible medical treatment processes. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 593–604. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_57
24. Moody, D.L.: The method evaluation model: a theoretical model for validating information systems design methods. In: ECIS 2003 Proceedings, p. 79 (2003)
25. Jayaram, J., Ahire, S.L., Dreyfus, P.: Contingency relationships of firm size, TQM duration, unionization, and industry context on TQM implementation: a focus on total effects. *J. Oper. Manag.* **28**, 345–356 (2010)
26. Shah, R., Ward, P.T.: Lean manufacturing: context, practice bundles, and performance. *J. Oper. Manag.* **21**, 129–149 (2003)
27. Lee, G., Xia, W.: Organizational size and IT innovation adoption: a meta-analysis. *Inf. Manag.* **43**, 975–985 (2006)
28. Schmiedel, T., vom Brocke, J., Recker, J.: Culture in business process management: how cultural values determine BPM success. In: vom Brocke, J., Rosemann, M. (eds.) *Handbook on Business Process Management 2*. IHIS, pp. 649–663. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-642-45103-4_27
29. Schmiedel, T., vom Brocke, J., Recker, J.: Which cultural values matter to business process management? Results from a global Delphi study. *Bus. Process Manag. J.* **19**, 292–317 (2013)
30. Thiemich, C., Puhlmann, F.: An agile BPM project methodology. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 291–306. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_25
31. Donaldson, L.: *The Contingency Theory of Organizations*. Sage, Thousand Oaks (2001)
32. Niehaves, B.: Open process innovation: the impact of personnel resource scarcity on the involvement of customers and consultants in public sector BPM. *Bus. Process Manag. J.* **16**, 377–393 (2010)

33. Wheeler, A.R., Halbesleben, J.R.B., Harris, K.J.: How job-level HRM effectiveness influences employee intent to turnover and workarounds in hospitals. *J. Bus. Res.* **65**, 547–554 (2012)
34. Blandford, A., Furniss, D., Vincent, C.: Patient safety and interactive medical devices: realigning work as imagined and work as done. *Clin. Risk* **20**, 107–110 (2014)
35. Park, S.Y., Chen, Y.: Adaptation as design: learning from an EMR deployment study. In: *Proceedings of the Conference on Human Factors in Computing Systems*, Austin, TX, pp. 2097–2106 (2012)
36. Barrett, A.K., Stephens, K.K.: Making electronic health records (EHRs) work: informal talk and workarounds in healthcare organizations. *Health Commun.* **32**, 1004–1013 (2017)
37. Malaurent, J., Avison, D.: From an apparent failure to a success story: ERP in China - post implementation. *Int. J. Inf. Manag.* **35**, 643–646 (2015)
38. Reijers, H.A., van Wijk, S., Mutschler, B., Leurs, M.: BPM in practice: who is doing what? In: Hull, R., Mendling, J., Tai, S. (eds.) *BPM 2010. LNCS*, vol. 6336, pp. 45–60. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15618-2_6
39. Burlton, R.: BPM critical success factors lessons learned from successful BPM organizations. *Bus. Rules J.* **12**, 1–6 (2011)
40. Ravesteyn, P., Jansen, S.: A situational implementation method for business process management systems. In: *AMCIS 2009 Proceedings*, p. 632 (2009)
41. Rosemann, M.: Proposals for future BPM research directions. In: Ouyang, C., Jung, J.-Y. (eds.) *AP-BPM 2014. LNBIP*, vol. 181, pp. 1–15. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08222-6_1
42. Rosemann, M., de Bruin, T.: Towards a business process management maturity model. In: *ECIS 2005 Proceedings*, p. 37 (2005)
43. Malhotra, R., Temponi, C.: Critical decisions for ERP integration: small business issues. *Int. J. Inf. Manag.* **30**, 28–37 (2010)
44. Germain, R., Spears, N.: Quality management and its relationship with organizational context and design. *Int. J. Qual. Reliab. Manag.* **16**, 371–392 (1999)



Regulatory Instability, Business Process Management Technology, and BPM Skill Configurations

Patrick Lohmann^(✉) and Michael zur Muehlen^(✉)

Stevens Institute of Technology, Castle Point on Hudson, Hoboken 07030, USA
{patrick.lohmann,michael.zurmuehlen}@stevens.edu

Abstract. This paper investigates how firms configure their business process management efforts in different industries. We generate a business process management (BPM) skills taxonomy through the computational linguistic analysis of job ads from [Monster.com](https://www.monster.com). We apply the taxonomy to [LinkedIn.com](https://www.linkedin.com) resumes of professionals employed at retailer Walmart, pharmaceutical company Pfizer, and investment bank Goldman Sachs. We find that Walmart and Pfizer distribute change- and operations-related BPM skills among the same roles whereas Goldman Sachs distributes both kinds of skills among more separate roles. This separation reflects a trilateral configuration where line managers and analysts focus on operational BPM tasks related to running processes while change-related tasks are covered by project managers. At Walmart and Pfizer the tasks of the BPM project manager are shared among managers and analysts, reflecting a bilateral configuration. Comparing each firm's regulatory environments and BPM technology capabilities, we conjecture that the organizational configuration pattern is influenced by a firm's ability to reliably automate business processes, since this affects how much attention line managers and analysts have to spend on monitoring processes and on reconciling issues and exceptions. This attention could otherwise be spent on regulatory-imposed process change efforts. This configural logic suggests a reconfiguration of BPM professionals towards a bilateral configuration when an organization transforms its business with digital technology, because the focus of such efforts includes process and decision automation.

Keywords: BPM skills · BPM taxonomy · BPM professionals · BPM function

1 Introduction

Regulatory interventions are a frequent source of organizational change in industries such as Pharmaceuticals or Financial Services, whereas other industries such as Retail or Transportation are subject to more measured changes of the regulatory regime. By regulation we mean authoritative operating rules accompanied by some formal governance mechanisms that promote rule compliance and sanction non-compliance and misconduct [2]. Governments typically impose regulation on organizations to increase market stability and transparency by permitting, directing, constraining their operations. When regulations change, organizations need to adapt their policies and

procedures, and hence they require business process management (BPM) professionals and a BPM function that can absorb the regulatory instability of their regulatory regime, akin to Ashby's [1] law of requisite variety. Which actions organizations take to match the complexity of their BPM function to that of their regulatory regime is thus an important question for regulators and those affected by regulation alike.

A regulatory regime is unstable when it comprises multiple supervisory authorities that collectively impose rules that are complex and frequently changing. A stable regulatory regime can be characterized by a limited number of authorities and a continuance of rules over time. The instability is different from a regulatory shock, i.e., sudden and extensive changes in the operating constraints [12]. *Regulatory shocks* are typically less predictable and more harmful to the business practices of affected organizations, either because they occur unanticipated (e.g., the imposition of tariffs in a trade conflict) or organizations delay the implementation of substantial rule changes (e.g., new privacy laws). In contrast, *regulatory instability* is a second-order measure of the delta in month-over-month, year-over-year changes in the rule book imposed by the regulatory regime that directs and constrains how business processes can be performed, making it a more predictable variable to which organizations can adapt and attend to.

Individuals are cognitively bounded with regard to how much information they can process at a time, and hence organizations need to configure their professionals to distribute their attention to salient issues and their solutions [22]. When organizations face a regulatory shock such as the introduction of the U.S. Sarbanes-Oxley Act, they often increase and/or complement their business and technology staff through new hires and audit and consulting engagements to ensure regulatory adaptation, and invest in regulatory technology to automate compliance monitoring and control [14, 16]. Unlike a shock, however, an unstable regulatory regime should require continual attention, and it stands to question whether affected organizations simply require more professionals than those operating under a stable regime or whether these organizations make more permanent changes to the configuration of their BPM function to reflect the different tasks performed by different professionals on a daily basis.

The skills of the professionals involved in BPM and their configurations can serve as a suitable proxy for the BPM function overall because organizations frequently practice BPM as a method to analyze, design, implement, monitor, and control their operations, treating processes as the socio-technical change objects [4]. When organizations practice BPM, they assume that processes have an ostensive and a performative aspect [9]. The ostensive aspect describes the process as a model and executable script that specifies the logical workflow design and implementation. The performative aspect relates to the actions performed by human and algorithmic system participants. The skills of BPM professionals constrain the change-related (ostensive) and operations-related (performative) BPM tasks they could possibly perform [21], and their collective configuration provides an upper bound of the ordinary and dynamic capabilities of what the BPM function can achieve in the context of an organization [18, 24].

In this paper, we ask *how the BPM function can be affected by regulatory instability, how such demands translate into different configurations of BPM professionals, and whether BPM technology can mediate between regulatory instability and these configurations*. By *BPM technology* we mean algorithms and computer systems that

automate operational tasks and processes, substituting human performance without causing exceptions that require human attention [26]. First, we empirically generate a skills taxonomy from job ads. Second, we apply the taxonomy to public resumes of professionals employed at three organizations from differently regulated industries. Specifically, we look at the BPM skills configurations at retailer Walmart, pharmaceutical company Pfizer, and investment bank Goldman Sachs, exposing the skill configurations of their BPM functions. We interpret the configurations considering the relative differences in the stability of each company's regulatory regime and BPM technology capability.

We find that while the fundamental skills within the BPM functions of Walmart, Pfizer, and Goldman Sachs are similar, their configurations differ. Change- and operations-related BPM skills are distributed among line managers and analysts at unstable-regulated Pfizer. The configuration mirrors that of stable-regulated Walmart, reflecting a bilateral BPM configuration where both roles possess the necessary skills to collectively operate and adapt processes. The skills to operate and adapt processes are more distributed at Goldman Sachs which relies on project managers to complement its line managers and analysts in process change efforts. Goldman Sachs allows its line managers and analysts to attend more to operating business processes, reflecting a trilateral BPM configuration. Comparing the BPM technologies of Pfizer and Goldman Sachs, we conjecture that the BPM configuration also depends on the capacity to automate processes affected by unstable regulation. A sophisticated BPM technology capability can allow line managers and analysts to shift their attention to change-related BPM tasks rather than compliance and escalation-related operational tasks, rendering the project manager redundant when performative process exceptions are caught and resolved by BPM technology as specified in ostensive models and scripts.

2 Regulatory Instability and BPM Professionals

Regulation affects an organization's processes and systems. Organizations that produce similar products or services are historically assigned to the same industry [8], and hence are regulated by the same regulatory regime. Although the designs of their processes often differ, regulation manifests as controls that must be implemented within existing processes (e.g., margin requirements of an investment bank's client borrowing liquidity to trade securities), or that formulate new processes (e.g., which party reports a trade), or that must not be performed (e.g., the trading desk must not engage in proprietary trading). Not all regulation permits and limits the product and service portfolio of organizations, but it may affect internal operating policies and procedures. The stability of the regulatory regime therefore not only varies between organizations from different industries but also between processes and units within the same organization.

As for the United States, the amount of regulation published over time through the Office of the Federal Register can serve as a suitable proxy of an industry's regulatory stability.¹ The number of regulators and rules which organizations must monitor,

¹ A list of publications released can be provided by the authors on request.

comply with, and adapt to depend on the size of their product and service portfolio and on their operational footprint because regions and countries can have their own authorities and standards. As for the European Union, it is safe to say that the stability of regulatory regimes is comparable to their U.S. peers. In relative terms, organizations in Consumer Goods and Financial Services are regulated by a more unstable regime, while organization in Retail experience a more stable regime.

Like other individuals, BPM professionals have a finite absorptive capacity to attend to all information. According to the attention-based view, organizations need to specify roles assumed by individuals and their reporting relationships to distribute time and effort to issues that require attention [22]. By *attention* we mean the noticing, enticing, and focusing of time and effort on issues (i.e., problems, threats, and opportunities) and their solutions facing the organization [22]. An unstable regulatory regime is a significant concern for affected organizations, and their challenge is to configure skills into roles so that everyone can effectively perform a finite set of tasks, attending to the subset of information relevant to their tasks. Collectively, the configuration should maintain the capacity to monitor and ensure day-to-day operational compliance and at the same time ensure a timely response to rule volatility imposed by the unstable regime.

The skills required of individuals to perform BPM tasks relevant to their role are commonly outlined in job descriptions [20]. To maintain a BPM capability, organizations need to articulate the skills they deem desirable using some formal or technical language and assign them to roles. BPM roles require operational knowledge and thus sit at middle and lower managerial ranks. We therefore expect organizations independent of their industry to also publicly hire to fill vacancies.

3 Method

We select a method to computationally explore and extract skills from job descriptions that does not rely on a pre-specified classification schema. The categories of such a taxonomy (in our case skills) are unknown because we cannot specify ex-ante which skills organizations seek in the context of their BPM practice and what words they use to articulate their needs. We acknowledge that O*Net provides a generic taxonomy for skills across different professions, but we found that this taxonomy would not allow us to discriminate between managerial and technical professional roles that are characteristic of BPM [17].

3.1 Datasets

Job Ads. We obtained job descriptions from job ads because they describe the perceived deficiencies of organizations in their staffing to maintain specific capabilities [23]. We downloaded the complete set of job ads that contained *business process*, *BPM*, *process improvement*, *process innovation*, or *process change* from [Monster.com](#), resulting in 45,484 job ads published in the U.S. and U.K. between January 2015 and 2016. We removed 20,121 ads related to non-permanent and heavy-industry production

occupations, leaving 25,363 job ads. We focused on both countries because they share a similar regulatory regime and language. We randomly sampled 1,000 job ads from Retail (NAICS: 44), Consumer Goods (31), and Financial Services (52), respectively, in order to normalize the amount of text representing any particular sector. Without this step, sectors with more job ads could have biased the generated skills taxonomy. We sampled job ads published by organizations for which we could confirm their industry affiliation through Dun & Bradstreet.

Resumes. Job ads provide a partial window into organizations because they do not expose the skills and roles which they already have. We used resumes to obtain a more complete understanding of the BPM professionals at three theoretically sampled organizations. From [LinkedIn.com](https://www.linkedin.com) we downloaded resumes of individuals who self-declared (as of September 2017) to work at retailer Walmart, pharma Pfizer, or investment bank Goldman Sachs. We only included those who used *business process*, *BPM*, *process improvement*, *process innovation*, or *process change* to describe their current full-time occupation. Again, we randomly sampled 1,000 resumes for each organization.

The three organizations are all large-scale in their employee size and among the most professionally-managed ones in their respective industry. From interviews with senior managers of each company, we learned that Goldman Sachs' process automation capability was constrained by that of their counterparties. Pfizer did not experience this degree of interdependence, being less dependent on external parties to complete a business process. Unlike Pfizer, Walmart and Goldman are service organizations but due to the nature of their transaction-like business operations, they have quasi-manufacturing processes.

3.2 Procedure

We generate a BPM skills taxonomy and map the skills of the professionals at Walmart, Pfizer, and Goldman Sachs into the taxonomy. First, we explore latent topics (in our case skills) across all job descriptions. Latent topics are word groups associated with a semantic context. In job ads, the semantic context can be thought of as skills, assuming that organizations can use different words to advertise the same skill. Second, we infer a skill set over each resume and cluster the skill sets for each firm, thereby exposing BPM configurations.

Modeling Skills. We use a topic model widely accepted in the computational linguistics literature [15], which is Latent Dirichlet Allocation (LDA) [5]. LDA is a generative model for classifying documents into multiple topics that are latent in their structure. A topic is a probability distribution over a lexicon. Topics are shared by all documents and the lexicon is shared among all topics. Words that have a high probability within a topic tend to co-occur across documents. LDA infers the classification scheme in form of word-topic and topic-document distributions.

The number of topics expected in the documents must be specified prior to model inference, and the optimal number can be identified by comparing their validity. We

use mean topic coherence as an indicator of internal validity, which measures how accurately high-probability words of a topic do co-occur across documents [6]. Perplexity measures external validity, the capacity of the inferred word-topic distributions to capture word co-occurrences in previously unseen documents [5]. We run three-fold cross-validation to train and test each model, obtaining a more robust score. Both mean topic coherence and perplexity are technical scores and their interpretation should not go beyond model comparison.

Mapping Skills and Roles. We infer a topic-document distribution over an imaginary document composed of a resume's current job title, description, and skills using the topic model which we identify as optimal in the previous step. To explore variation in the skill configurations, we group the topic-document distributions and run separate cluster analyses for each firm. We cluster skill sets based on their pairwise Hellinger distances using a bottom-up algorithm (the Ward method) that aims to minimize the total within-cluster variance at each level. The emerging clusters can be thought of as distinct roles, i.e., skill sets. To interpret the clusters in a particular layer, we look at the most-representative resumes which we define as those whose topic-document distributions are closest to the mean distribution, i.e., the topic-role distribution, of all resumes assigned to a particular cluster. In other words, we look at the cluster centroids.

3.3 Analysis

We performed a number of text reduction steps to the job ads [15]. The outcome was a lexicon and sentence vectors counting the frequency with which each word occurred in a sentence. Not all information provided in job ads refer to the job description [23], and hence we cleansed them.

Removing Duplicate Job Ads. We removed duplicates based on the cosine distance of their aggregated sentence vectors. The .10-threshold yielded the best F-measure (.90) in detecting duplicates as classified by two researchers (Cohen's $\kappa = .78$). The threshold was conservative, given that the average job ad contained 93 distinct words (s.d. = 46). We identified 7,860 job ads as being duplicates, leaving 17,503.

Removing Noise Sentences. We applied LDA over the sentences to remove those that obviously did not address skills but contextual information. The idea was to interpret each sentence as a distribution of domain topics and noise topics, and to remove sentences from a job ad that exhibit a high probability of being associated with a noise topic. The 40-topic model provided the best balance between internal and external validity, separating distinct themes into different topics. We treated a topic as noise when it addressed a theme that was obviously not domain-related ($\kappa = .93$). A .50-threshold yielded the best F-measure (.82, $\kappa = .74$) in detecting noise sentences, resulting into a removal of 77,700 sentences (19.6%).

Modeling Skills. We aggregated the remaining sentence vectors for each job ad and reran LDA. The 24-topics model generated substantive topics and provided the best

balance between internal and external validity. Two researchers independently labeled each topic and synthesized final labels with a third researcher.²

Mapping Skills and Roles. We selected the 24-topic model to infer a topic-document distribution over each resume's relevant sections which we identified by their proper HTML tags. We grouped the resumes for each firm and evaluated the cluster solutions. The best solution was identified by a distinct bent when being plotted and validated using the gap statistic which measures the goodness of a solution relative to a randomly sampled solution of the same number of clusters [25]. Both indicators suggested the five-cluster solution for Walmart and the six-cluster solution for Pfizer and Goldman Sachs, which produced simple models that exposed the fundamental structural differences between their respective BPM skill configurations. Two researchers independently inferred cluster labels from the skill configurations and job titles of the representative resumes and synthesized them with a third researcher.

4 Findings

We interpret the BPM skills as being related to either change- or operations-related tasks. Change-related BPM skills help to intentionally change the design, models, organizational structures, and software code of a business process (i.e., ostensive aspects [9]) and to translate such changes into operations (i.e., performative aspects). Operations-related BPM skills support individuals to align the actual operations of the process with its intended design to stabilize and standardize them. Together, both types of skills provide the basis for adapting and operating processes and systems.

Figures 1, 2, and 3 illustrate the skill configurations at Walmart, Pfizer, and Goldman Sachs. The dark crosses and circles represent the change- and operations-related skills ($\kappa = .94$). Their relative positioning on the map reflects the similarity of their underlying word-topic distributions generated by our LDA. To compare differences in the configurations, the opaqueness of a curve that links a skill with a cluster center is a function of whether the skill receives a cluster relevance that is within the 90th, 80th, 70th, or 60th percentile of skill attentions across all clusters.

4.1 BPM Professionals at Walmart

The professionals within Walmart's BPM function suggest a configuration into three line manager and two analyst roles (see Fig. 1). The *front-office manager* aggregates professionals skilled to perform and administer customer-facing functions such as marketing and sales. The *back-office manager* bundles those who work in enterprise- and partner-facing functions, e.g., finance and accounting. The *operations manager* comprises supply chain and logistics professionals. All three manager roles possess skills related with operations-related BPM, however, some change-related skills appear to be boundary-spanning in the sense that they integrate the roles: *business strategy*, *product development*, *business transformation*, and *change management* are shared

² A list of high-probability words per topic can be provided by the authors on request.

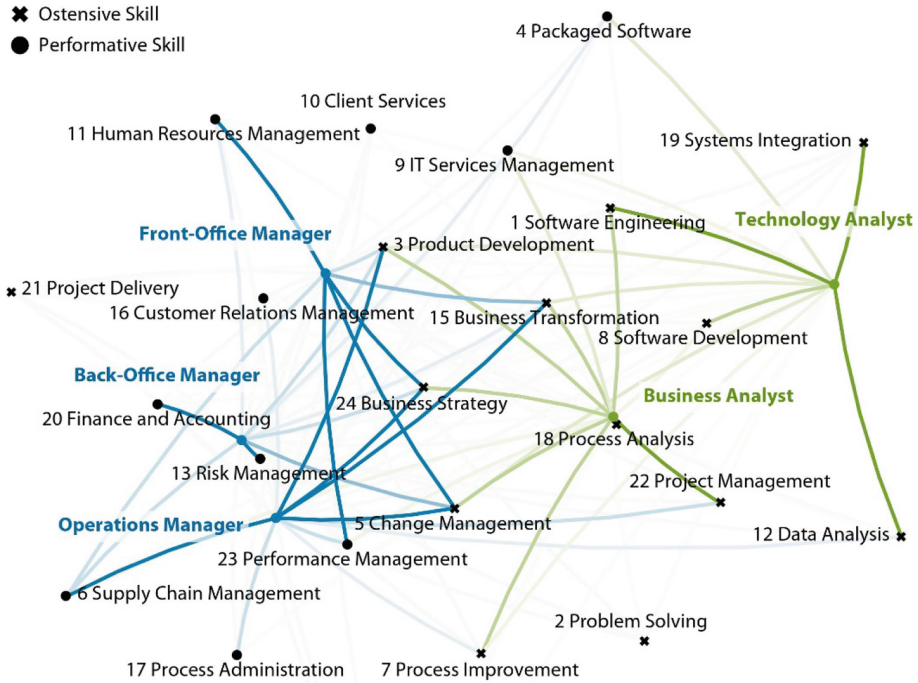


Fig. 1. BPM configuration at Walmart

change-related skills. These managers are typically responsible for the bottom-up championing and the top-down implementation of strategic programs, and hence they need the people skills necessary to execute and deliver operational changes, reflected in the *performance management* skill of *front-office* and *operations managers*. Walmart’s back-office operations are highly-automated with their SAP systems, and hence performance management is less of a concern for *back-office managers*.

The *business analyst* role links to typical process project-related skills such as *project management*, *process analysis*, and *software engineering*, whereas professionals aggregated under the *technology analyst* role are skilled in *software development* and *systems integration* but also *data analysis*. Both roles link mostly to change-related BPM skills with the *business analyst* occupying a central position within the BPM configuration as an interface between business and technology. This configuration mirrors the common understanding of analyst role in the literature as a boundary spanner and knowledge broker. This role appears to be more assigned to a process on a project basis in order for its optimization or change.

The skills of *technology analysts* reflect Walmart’s IT strategy to have a commercial ERP system for the standardization of non-differentiating processes, while engineering merchandizing and supply chain systems more on their own as a measure to better integrate e-commerce and department-store operations. The responsibilities for new software developments reside in Walmart’s digital business unit Global

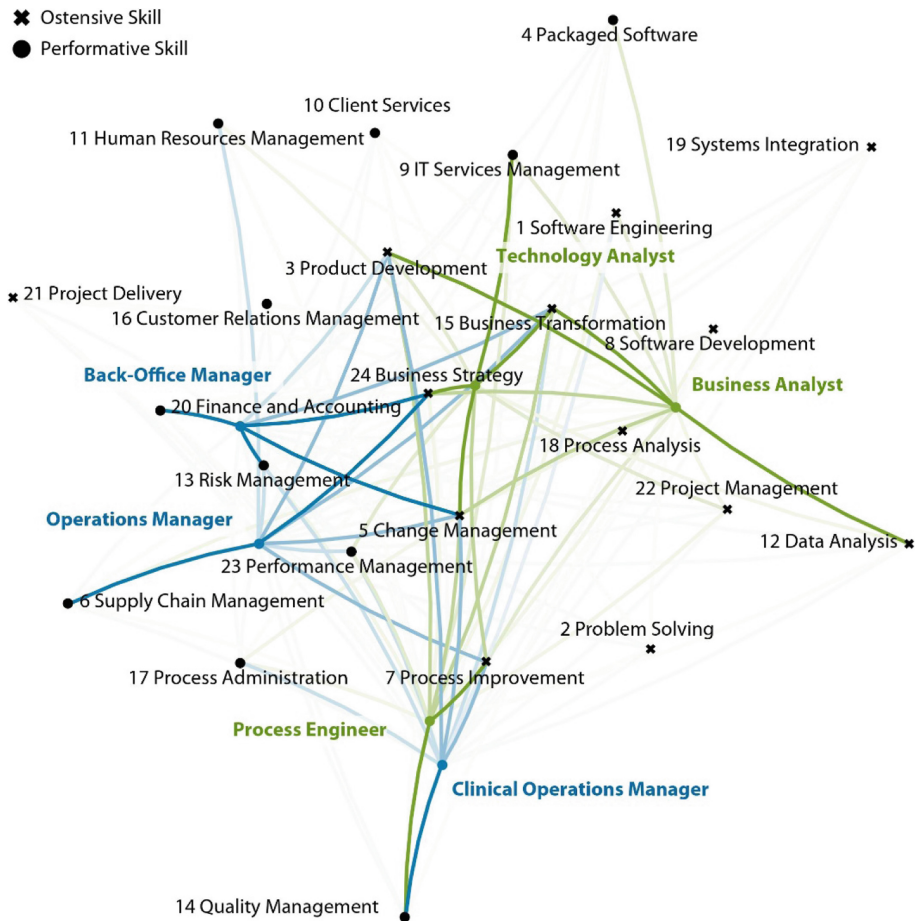


Fig. 2. BPM configuration at Pfizer

eCommerce, and hence it is plausible that Walmart's *technology analysts* are mostly skilled in the integration of backbone application systems.

4.2 BPM Professionals at Pfizer

The professionals within Pfizer's BPM function are grouped into three manager and three analyst roles (see Fig. 2). The *back-office manager* and *operations manager* comprise professionals with skills similar to their peers at Walmart which makes sense due to the similarity of their business models of selling physical products. Pfizer sells its products to commercial businesses rather than end consumers, which can explain the missing *front-office manager* role because out-bound processes are under the auspices of operations managers. Instead, a *clinical operations manager* role bundles the professionals skilled to monitor and control the new drug and medicine development processes. Professionals in this role possess operations-related BPM skills related to

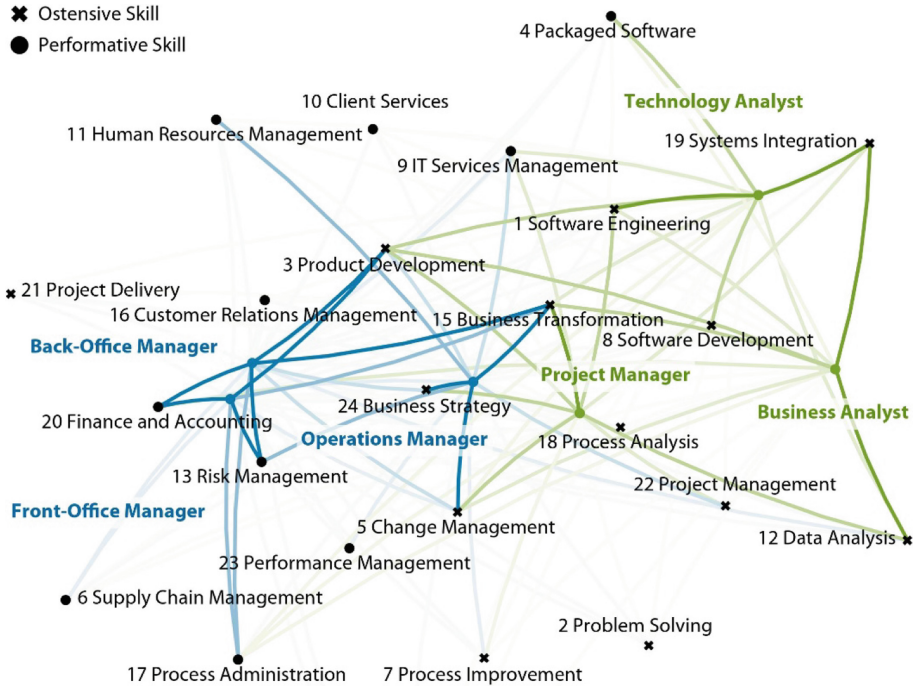


Fig. 3. BPM configuration at Goldman Sachs

process administration, risk management, and quality management, and change-related skills such as product development, process improvement, and change management.

Pfizer’s manufacturing operations add the *process engineer* role that comprises professionals with change-related skills relevant for the design of physical production processes. These professionals have similar skills as *clinical operations managers*, however, *process engineers* are those who can engineer reliable manufacturing processes, according to their resumes. *Process engineers* are skilled to analyze, design, and improve the high-volume processes whose performances are highly automated within the firm. The *clinical operations manager* monitors and controls processes such as clinical trials which involve different actors from within and outside of the firm and which are less standardizable in their performance. The *business analyst* role summarizes skills akin to its peer at Walmart. Pfizer’s *technology analysts* are more strategic-oriented towards change, as reflected in the links to *business strategy*, *product development*, *business transformation*, and *change management*. Neither the *business analyst* nor the *technology analyst* exposes software development and engineering skills, suggesting a skills chasm between the firm’s BPM and IT function.

4.3 BPM Professionals at Goldman Sachs

The professionals within Goldman Sachs’ BPM function are grouped into four manager and two analyst roles (see Fig. 3). The *front-office* and *back-office managers* comprise

professionals with skills mostly related to operations-related BPM such as *risk management*, *process administration*, and *finance and accounting*, however, the development of new innovative financial products is also part of their skill sets. This suggests that these professionals see their role more focused on identifying, monitoring and controlling (new ways of) revenue-generating client interactions and transactions in alignment with the firm's risk appetite. The *operations manager* bundles the professionals skilled in change-related BPM, according to the links to *business strategy*, *product development*, *business transformation*, and *change management*. *Risk management* and *IT services management* being operations-related skills seem to be also their concern.

The emphasis of skills of Goldman Sachs' managers on emphasizing risk, compliance, and administration compared with a relatively technical orientation of their *business analysts* arguably necessitates the existence of the *project manager* role that comprises professionals with design and change skills to coordinate between business and technology. These professionals possess a skill set comparable to Walmart's business analysts.

5 Discussion

We asked how organizations configure their BPM professionals to attend to the changes caused by regulation. Our three cases suggest that organizations whose processes are regulated by an unstable regime separate change- from operations-related BPM tasks if the processes require extensive manual monitoring and control by managers and analysts. If operations-related BPM tasks are automated by BPM technology, less manual operational attention is needed. A higher BPM technology capability could allow end-to-end workflow automation with minimum exceptions, substituting human by algorithmic attention. Eliminating humans from a process can reduce variation between intended and actual process operations because processes are executed as specified by the model and executable script. This allows managers and analysts to attend more to change-related BPM tasks and regulatory adaptation. This configuration is reflected in the BPM professionals of an organization that is regulated by a stable regime. We thus distinguish between a *bilateral* and a *trilateral BPM configuration* and conjecture that the configuration of an organization operating under an unstable regulatory regime also depends on its BPM technology capability.

5.1 Bilateral and Trilateral BPM Configuration

The intended (ostensive) and actual (performative) operation of a process are in constant friction. That means the performative process can become the ostensive process when being performed repeatedly in deviation from its intended design, and the ostensive process may change even though this change does not materialize in the actual performance of the process [9]. Regulation affects an organization's ostensive processes, followed by the performative processes. That said, regulatory compliance is assessed against the performative processes, not the ostensive ones.

The *ostensive misfit* between external regulatory requirements and internal ostensive processes and the *performative misfit* between ostensive and performative processes impose pressures on an organization and demand attention. Both ostensive and performative misfits represent “exceptions,” and BPM professionals have finite attention to address and solve all exceptions given finite time. Depending on the frequency of regulatory changes triggering ostensive misfits and the provenance of performative misfits, we propose that organizations rely either on a bilateral or trilateral configuration of their BPM professionals to maintain the necessary absorptive capacity to attend to these exceptions and their resolution (see Fig. 4).

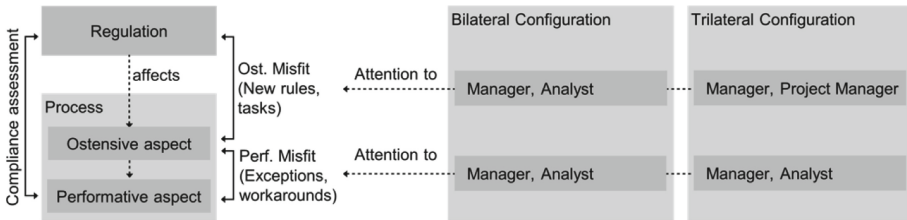


Fig. 4. Bilateral and trilateral BPM configuration

If the regulatory regime is stable, then exceptions are mostly caused by performative misfits. A bilateral BPM configuration allows to align the performance of processes to their intended designs, and to execute and implement intended changes because professionals combine the necessary BPM skills for design and execution. Performative misfits are either caused by individuals or technology performing process activities. Managers can coordinate and solve human misalignments whereas algorithmic misalignments are solved by analysts. Managers monitor and control everyday process operations to maintain performative fit, applying methods such as Lean Management to alter ostensive processes and Six Sigma to translate such changes into measurable performative improvements. These managers have the skills to identify performance problems and their root causes, and take action for their resolution. Business and technology analysts address algorithmic misalignments, and resolving such misfits is an ostensive rather than performative problem because it requires changes to executable scripts. The bilateral interactions between the professionals required to change ostensive and performative processes are characteristic of the bilateral BPM configuration.

Regulation is published as a complex web of textual non-machine-readable documents and are often written with a significant amount of ambiguity. Implementing new regulations is not a straight-forward task. It imposes significant amounts of sense-making time on human individuals. This sense-making involves understanding the documents and their relations to others, assessing implications on processes, and searching for an optimal way to implement requirements and change policies and procedures. This adaptation demands special attention [26].

Martinez-Moyano et al. [19] explain how agency problems in investment banks such as Goldman Sachs cause profit maximization pressures that run against compliance, increasing rule violations that in the absence of intervention derail the business. Pfizer faces similar tensions between maximizing output and ensuring regulatory compliance of its production processes. Its higher BPM technology capability though requires managers and analysts to spend less of their time with monitoring and control the compliancy of their operations, reflecting a bilateral BPM configuration. These processes have a higher straight-through processing rate to benefit from scale economies, allowing managers and analysts focus more on engineering rather than administration. It is therefore not surprising that these professionals are more engineers than classical analysts, indicating the comprehensive tasks performed by these professionals.

If the regulatory regime is unstable *and* processes *can* be reliably automated by BPM technology to cause minimum performative misfit that requires human intervention, ostensive misfits caused by differences between an organization's regulatory requirements and ostensive processes become the major source requiring attention. Under such a condition, managers and analysts can focus on regulatory realignment because BPM systems control and perform processes and activities. Ostensive changes to executable scripts directly translate into their automated performance.

Unlike Pharmaceuticals, Financial Services is a heavily vertically-disintegrated industry with by many specialized firms buying and selling financial products both electronically and over the counter. Goldman Sachs processed more than one million trades per day with a straight-through processing rate of about 96% at the time of our data collection, with a significant amount of time of their operations managers and analysts being consumed by reconciling errors and interruptions. While Pfizer has more control of its unstably-regulated processes because it can buffer them from upstream and downstream partners, the capability of financial services providers to perform with minimal exceptions also depends on the technology capability of their counterparties because financial services production and delivery processes are heavily inter-organizational. The capability algorithmically buffer internal operations from, and reconcile, counterparty-caused exceptions depends on a firm's ability to develop automatable ostensive scripts that can algorithmically catch and handle such issues. The complexity in market interactions makes such engineering efforts inherently difficult.

If the regulatory regime is unstable *and* performative processes *cannot* be reliably automated because BPM technology has a lower capability, the organization simultaneously faces pressures from ostensive and performative misfits. Under such a condition, we propose that the organization relies on a trilateral BPM configuration because managers and analysts are more concerned with reconciling and resolving performative misfits to ensure that processes achieve their desired outcome. Therefore, a project manager addresses ostensive misfits, being in charge of redesign and adaptation. The trilateral interactions between the professionals required to change ostensive and performative processes allow an organization to maintain the requisite absorptive capacity to cope with an unstable regulatory regime when operational attention cannot be shifted to BPM technology.

Not all processes of an unstable-regulated organization experience regulatory instability. For example, regulatory requirements of typical support processes such as

human resources and payroll are often stable. Different unstable-regulated processes typically also have a different BPM technology capability. We conjecture that organizations change such processes by applying a bilateral approach, coordinated by analysts rather than project managers because such processes require less attention to ostensive misfits. The bilateral and trilateral BPM configurations are not necessarily mutually-exclusive but can coexist within the same organization. Rather, these organizations situationally switch between a bilateral and trilateral approach to process management within their trilateral BPM configuration. We find support for this conjecture in Goldman Sachs' *analyst* and *project manager* clusters, which absorb process engineers in their periphery. These engineers have skills akin to Pfizer's process engineers – for digital rather than physical processes.

5.2 Implications

While our analysis is motivated by the bottom-up clustering and aggregation of skills of human individuals into collective skill configurations at three industry-leading firms, in practice it is more plausible that these organizations take a decompositional approach by assessing their regulatory regime and BPM technology capability first, and by specifying the roles and their required skills accordingly. Regardless of whether such specifications are based on macro-to-micro decomposition principles or micro-to-macro aggregation principles, or a mix of both, our cross-sectional analysis cuts levels of organizational analysis. We show how macro-level effects about regulatory instability and technology can relate with micro-level skill configurations of an organization's BPM practice. These configurations represent a human resources-based perspective on the microfoundations of an organization's ambidextrous process management capability that addresses the ostensive and performative issues that require attention [20]. Arguably because of a shortage of analysis techniques and datasets, the linkage between microfoundations (individual, skills) and macrofoundations (organization, capability) is mostly a theoretical debate [3, 10]. Our contributions here are method-wise, and our approach to uncover skills and configurations can help other academics better understand the linkages between individual and organizational aspects affecting human resources and organizational performance both within and outside of BPM [7].

On the practical side, our discovered configurations can serve as a basis for professionals in charge of a BPM practice better assess their current BPM target operating model, linking roles, skills, and capability. Digital and information technologies reduce vertical layers of management hierarchy [11, 13]. However, BPM technology may also alter the horizontal structuring of managers when processes become digitized and their operations automated. BPM technology capability creates options for Chief Process Officers to rethink the skill sets required of their staff to better enable and support the digital transformation of their organization over time. As organizations continue to automate their routine and non-routine tasks with artificial intelligence and machine learning, robotic process automation, cognitive agents, and distributed ledger technology, among other technologies, these digital transformations turn operations-related BPM tasks inherently into engineering problems. Chief Process Officers should keep an eye on their BPM technology capability and put in place plans for the enterprise-wide education and reskilling professionals involved in the operational management and

administration of processes to allow for a shift from a potential trilateral configuration to a bilateral one. We speculate that digitally-transformed organizations can allow their managers and analysts to attend more to the innovation of minimum viable products and services. However, our analysis does not consider performance measures of both configurations, and hence future research should try to link both configurations with organizational performance implications.

5.3 Limitations

While our study provides important insights into the skills and configurations of BPM professionals, our findings are subject to a limitation that relates to the nature of job ads and self-reported resumes as a research dataset. For example, an organization may announce a presumed vacancy less with the motivation to fill this role but to be recognized as a legitimate business partner for (potential) customers. Unstable-regulated industries may be more likely to write about governance, risk, and compliance in their job ads than organizations operating under a more stable regulatory regime, ultimately paying less attention to other skills and tasks relevant to the role.

We tried to mitigate this concern by generating a common skills taxonomy across the Retail, Consumer Goods, and Financial Services industries rather than an industry-specific skills taxonomy, and by mapping the resumes into the common taxonomy. Similar concerns may exist for the skills and job descriptions reported in resumes. Professionals certainly use [LinkedIn.com](https://www.linkedin.com) to connect and advertise themselves to increase their visibility and attractiveness for other employers, and hence they will likely select words in their resumes that add to their employability. We addressed this potential self-reporting bias methodologically by selecting a randomly-sampled and sufficiently large amount of resumes, and by averaging the skills reported in each research, looking at the cluster centroids rather than peripheries. The findings provide insights into the *means* and not ends of the BPM configurations of three theoretically-sampled organizations. Future research should test the BPM configurations and conjectures to a broader set of organizations, industries, and aim to establish a link between BPM skill configurations and performance implications of these organizations.

6 Conclusion

An organization's BPM technology can play an important mediating function between its regulatory regime and the configuration of its BPM professionals. Investment bank Goldman Sachs is an example of how a lower process automation capability can be associated with a trilateral BPM configuration that is used to maintain absorptive capacity to attend to the ostensive and performative misfits caused by regulation and operation. Pfizer is an organization that manages to maintain a higher automation capability within an unstable regulatory context and appears to be able to transfer performative BPM tasks from its professionals to the technology, reflecting a bilateral BPM configuration. This configuration mirrors that of an organization that operates under a stable regulatory regime, exemplified by our analysis of Walmart.

References

1. Ashby, W.R.: *An Introduction to Cybernetics*. Chapman & Hall, London (1956)
2. Baldwin, R., Scott, C., Hood, C.: *A Reader on Regulation*. Oxford University Press, Oxford (1998)
3. Barney, J., Felin, T.: What are microfoundations? *Acad. Manag. Perspect.* **27**, 138–155 (2013)
4. Benner, M.J., Tuschman, M.L.: Exploitation, exploration, and process management: the productivity dilemma revisited. *Acad. Manag. Rev.* **29**, 238–256 (2003)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn.* **3**, 993–1022 (2003)
6. Boyd-Graber, J.L., Mimno, D., Newman, D.: Care and feeding of topic models: problems, diagnostics, and improvements. In: *Handbook of Mixed Membership Models and their Applications*. CRC Press (2014)
7. vom Brocke, J., Zelt, S., Schmiedel, T.: On the Role of Context in Business Process Management. *Int. J. Inf. Manag.* **36**, 486–495 (2016)
8. Executive Office of the President: *National Industry Classification System* (2017)
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. Felin, T., Foss, N.J., Heimeriks, K.H., Madsen, T.L.: Microfoundations of routines and capabilities: individuals, processes, and structure. *J. Manag. Stud.* **49**, 1351–1374 (2012)
11. Frey, C.B., Osborne, M.A.: *The future of employment: how susceptible are jobs to computerization*. University of Oxford (2013)
12. Haveman, H.A., Russo, M.V., Meyer, A.D.: Organizational environments in flux: the impact of regulatory punctuations on organizational domains. *Organ. Sci.* **12**, 253–273 (2001)
13. Hickson, D.J., Pugh, D.S., Pheysey, D.C.: Operations technology and organization structure: an empirical reappraisal. *Adm. Sci. Q.* **14**, 378–397 (1969)
14. Iliev, P.: The effect of SOX Section 404: costs, earnings quality, and stock prices. *J. Finance* **65**, 1163–1196 (2010)
15. Jurafsky, D., Martin, J.H.: *Speech and Language Processing: An Introduction to Natural Language Processing*. Prentice Hall, Upper Saddle River (2009)
16. Krishnan, J., Rama, D., Zhang, Y.: Costs to comply with SOX Section 404. *Audit. J. Pract. Theory* **27**, 169–186 (2008)
17. Lohmann, P., zur Muehlen, M.: Business process management skills and roles: an investigation of the demand and supply side of BPM professionals. In: *13th International Conference on Business Process Management* (2015)
18. Martin, J.A.: Dynamic managerial capabilities. *Organ. Sci.* **22**, 118–140 (2011)
19. Martinez-Moyano, I.J., McCaffrey, D.P., Oliva, R.: Drift and adjustment in organizational rule compliance: explaining the “regulatory pendulum” in financial markets. *Organ. Sci.* **25**, 321–338 (2017)
20. Molloy, J.C., Barney, J.B.: Who captures the value created with human capital? A market-based view. *Acad. Manag. Perspect.* **29**, 309–325 (2015)
21. Nelson, R.R., Winter, S.G.: *An Evolutionary Theory of Economic Change*. Harvard University Press, Cambridge (1982)
22. Ocasio, W.: Towards an attention-based view of the firm. *Strateg. Manag. J.* **18**, 187–206 (1997)
23. Rafaeli, A., Oliver, A.L.: Employment ads: a configurational research agenda. *J. Manag. Inq.* **7**, 342–358 (1998)

24. Röglinger, M., Schwindenhammer, L., Stelzl, K.: How to put organizational ambidexterity into practice – towards a maturity model. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNBIP, vol. 329, pp. 194–210. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98651-7_12
25. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. *J. Roy. Stat. Soc. B* **63**, 411–423 (2001)
26. Woodward, R.E.: Task complexity: definition of the construct. *Organ. Behav. Hum. Decis. Process.* **37**, 60–82 (1986)

Author Index

- Amyot, Daniel 102
Artale, Alessandro 139
Augusto, Adriano 268
- Baarir, Souheib 52
Bandara, Wasana 338, 370
Barnes, Janne 338
Bauer, Martin 179
Beerepoot, Iris 402
Braberman, Victor 35
Burattin, Andrea 86
- Calvanese, Diego 157
Camargo, Manuel 286
Carmona, Josep 86, 250
Colombo Tosatto, Silvano 119
- D'ippolito, Nicolás 35
Damiani, Ernesto 232
Dinter, Barbara 354
Dumas, Marlon 268, 286
- Ghasemi, Mahdi 102
Ghilardi, Silvio 157
Gianini, Gabriele 232
Gianola, Alessandro 157
Gibson, Jeremy 338
Goel, Kanika 338, 370
González-Rojas, Oscar 286
Governatori, Guido 119
Grisold, Thomas 23
Gross, Steven 23
- Hoogendoorn, Mark 198
Houhou, Sara 52
- Klinkmüller, Christopher 322
Kollwitz, Christoph 354
Kovtunova, Alisa 139
Kramer, Jeff 35
- La Rosa, Marcello 268
Ladleif, Jan 69
- Leotta, Francesco 16
Lohmann, Patrick 419
Lu, Xixi 198
- Marrella, Andrea 16
Mecella, Massimo 16
Mizouni, Rabeb 232
Montali, Marco 86, 139, 157
Mühlhäuser, Max 216
Müller, Richard 322
- Nahabedian, Leandro 35
Nolle, Timo 216
- Padró, Lluís 86, 250
Poizat, Pascal 52
- Quéinnec, Philippe 52
- Reijers, Hajo A. 198, 402
Rivkin, Andrey 157
Röglinger, Maximilian 23
Rosemann, Michael 305, 370
- Sadiq, Shazia 10
Sánchez Guinea, Alejandro 216
Sánchez-Ferreres, Josep 86
Seeliger, Alexander 216
Shafagatova, Aygun 386
Stelzl, Katharina 23
- Tabatabaei, Seyed Amin 198
Tello, Ghalia 232
- Uchitel, Sebastián 35
- van Beest, Nick 119
van de Weerd, Inge 402
van der Aa, Han 179
van der Aalst, Wil M. P. 3, 139

Van Looy, Amy [386](#)
vom Brocke, Jan [23](#)

Weber, Ingo [69](#), [322](#)
Weidlich, Matthias [179](#)

Weske, Mathias [69](#)
Wurm, Bastian [370](#)
Wynn, Moe Thandar [10](#)

zur Muehlen, Michael [419](#)