

LNCS 13585

João Paulo A. Almeida
Dimka Karastoyanova
Giancarlo Guizzardi
Marco Montali
Fabrizio Maria Maggi
Claudenir M. Fonseca (Eds.)

Enterprise Design, Operations, and Computing

26th International Conference, EDOC 2022
Bozen-Bolzano, Italy, October 3–7, 2022
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

Moti Yung 

Columbia University, New York, NY, USA


More information about this series at <https://link.springer.com/bookseries/558>


João Paulo A. Almeida · Dimka Karastoyanova ·
Giancarlo Guizzardi · Marco Montali ·
Fabrizio Maria Maggi ·
Claudenir M. Fonseca (Eds.)


Enterprise Design, Operations, and Computing

26th International Conference, EDOC 2022
Bozen-Bolzano, Italy, October 3–7, 2022
Proceedings


Editors


João Paulo A. Almeida 
Universidade Federal do Espírito Santo
Vitória, Espírito Santo, Brazil

Dimka Karastoyanova 
University of Groningen
Groningen, The Netherlands

Giancarlo Guizzardi 
University of Twente
Enschede, The Netherlands

Marco Montali 
Free University of Bozen-Bolzano
Bolzano, Italy

Fabrizio Maria Maggi 
Free University of Bozen-Bolzano
Bolzano, Italy

Claudenir M. Fonseca 
University of Twente
Enschede, The Netherlands

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-17603-6 ISBN 978-3-031-17604-3 (eBook)
<https://doi.org/10.1007/978-3-031-17604-3>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2022

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

EDOC 2022 is the 26th conference in the EDOC series which provides a key forum for researchers and practitioners in the field of Enterprise Computing. EDOC addresses the full range of models, methodologies, and engineering technologies contributing to building and evolving intra- and inter-enterprise software systems. This year's conference has a special emphasis on the theme of designing and operating "Flexible Enterprises". The theme reflects an ever-changing world under the influence of disruptive events, trends and technologies, as well as the increasing role of artificial intelligence.

We are happy to announce a number of developments for EDOC. First, the EDOC acronym is now spelled-out "Enterprise Design, Operations and Computing" to reflect the broad range of aspects of interest to the conference. Second, proceedings are now published by Springer in the Lecture Notes in Computer Science (LNCS) series. Finally, EDOC is collocated for the first time with the International Conference on Cooperative Information Systems (CoopIS) now in its 28th edition. This collaboration led to a program with several joint events, including shared keynote speeches, panels, technical sessions and social functions. We look forward to a most fruitful interaction between the communities involved in the two conferences, and an exciting overall program.

These proceedings include 15 full papers selected out of 48 full papers sent for peer review (31.25% acceptance rate). All submissions were thoroughly reviewed in a single-blind process by at least three program committee members and, in the vast majority of cases, by four program committee members. The review process was led by the program committee chairs João Paulo A. Almeida and Dimka Karastoyanova and overseen by the general chairs Giancarlo Guizzardi, Marco Montali and Fabrizio Maria Maggi. The selected papers cover topical areas such as Enterprise Architecture, Enterprise Security, Business Process Mining and Discovery, Business Process Modeling and Monitoring and Development of Process-Driven Applications. We would like to show our greatest appreciation to the submitting authors and to the members of the program committee as well as additional reviewers for their hard work.

These proceedings further include abstracts which pair with the invited talks of our three renowned keynote speakers: Carliss Y. Baldwin (Harvard Business School, USA, *emerita*), Jordi Cabot (Universitat Oberta de Catalunya, Spain) and Giovanni Sartor (University of Bologna & European University Institute of Florence, Italy); and with the tutorial offered by Gerd Wagner (Brandenburg University of Technology, Cottbus, Germany.) We would like to thank them all for their generosity in joining us in Bolzano.

Companion post-conference proceedings will be published separately and will include papers selected for the EDOC forum alongside workshop papers, doctoral consortium papers and demonstration-track papers.

We would also like to thank the EDOC steering committee for entrusting us with the responsibility of organizing this year's conference. We would like to express our gratitude to all members of the organizing committee, and in particular our local organization committee. The local organization chairs Tiago P. Sales, Mattia Fumagalli, Pedro Paulo F. Barcelos and Claudenir M. Fonseca put in a lot of energy for a successful event.

Claudenir M. Fonseca should be acknowledged for his key role in our web presence, registration, and for supporting us in putting the program and these proceedings together.

We would like to thank the Free University of Bozen-Bolzano and the NOI Techpark for generously sponsoring and hosting the conference.

Finally, there can be no conference without engaged participation: we would like to express our deep gratitude to all who contributed with their insights to make our conference program interesting and all those who came to Bolzano to make EDOC lively. We were thrilled to organize the first in-person edition of EDOC after two years of online editions.

October 2022

João Paulo A. Almeida
Dimka Karastoyanova
Giancarlo Guizzardi
Marco Montali
Fabrizio Maria Maggi
Claudenir M. Fonseca

Organization

General Chairs

Giancarlo Guizzardi	Free University of Bozen-Bolzano, Italy & University of Twente, The Netherlands
Marco Montali	Free University of Bozen-Bolzano, Italy
Fabrizio Maria Maggi	Free University of Bozen-Bolzano, Italy

Program Committee Chairs

João Paulo A. Almeida	Federal University of Espírito Santo, Brazil
Dimka Karastoyanova	University of Groningen, The Netherlands

Steering Committee Chair

Selmin Nurcan	University Paris 1 Panthéon-Sorbonne, France
---------------	--

Steering Committee

Colin Atkinson	University of Mannheim, Germany
Georg Grossmann	University of South Australia, Australia
João Paulo A. Almeida	Federal University of Espírito Santo, Brazil
Marten van Sinderen	University of Twente, The Netherlands
Remco Dijkman	Eindhoven University of Technology, The Netherlands
Robert Lagerström	KTH, Sweden
Stefanie Rinderle-Ma	University of Vienna, Austria
Sylvain Hallé	Université du Québec à Chicoutimi, Canada
Zoran Milosevic	Deontik & Best Practice Software, Australia

Workshop Chairs

Tiago P. Sales	University of Twente, The Netherlands
Henderik A. Proper	Luxembourg Institute of Science and Technology & University of Luxembourg, Luxembourg

Demo Track Chairs

Massimiliano de Leoni	University of Padua, Italy
Ivan Donadello	Free University of Bozen-Bolzano, Italy
Cristine L. Griffo	Free University of Bozen-Bolzano, Italy

Industrial Chairs

Zoran Milosevic	Deontik/Best Practice Software, Australia
Flavia Santoro	Rio de Janeiro State University, Brazil

Doctoral Consortium Chairs

Felix Mannhardt	Eindhoven University of Technology, The Netherlands
Chiara Di Francescomarino	Fondazione Bruno Kessler, Italy

Proceedings Chair

Claudenir M. Fonseca	University of Twente, The Netherlands
----------------------	---------------------------------------

Website Chair

Claudenir M. Fonseca	University of Twente, The Netherlands
----------------------	---------------------------------------

Financial Chairs

Cristine L. Griffo	Free University of Bozen-Bolzano, Italy
Glenda Amaral	Free University of Bozen-Bolzano, Italy
Renata S. S. Guizzardi	University of Twente, The Netherlands

Publicity Chairs

Dominik Bork	TU Wien, Austria
Estefanía Serral	KU Leuven, Belgium

Local Organization Chairs

Tiago P. Sales	University of Twente, The Netherlands
Mattia Fumagalli	Free University of Bozen-Bolzano, Italy
Pedro Paulo F. Barcelos	Free University of Bozen-Bolzano, Italy
Claudenir M. Fonseca	University of Twente, The Netherlands

Local Organization Committee

Isadora Valle Sousa	Free University of Bozen-Bolzano, Italy
Riccardo Baratella	Free University of Bozen-Bolzano, Italy
Elena Romanenko	Free University of Bozen-Bolzano, Italy

Program Committee

Aditya Ghose	University of Wollongong, Australia
Andrea Marrella	Università di Roma, Italy
Alan Wee-Chung Liew	Griffith University, Australia
Alexander Knapp	Universität Augsburg, Germany
Alfred Zimmermann	Reutlingen University, Germany
Amin Beheshti	Macquarie University, Australia
Andreas L. Opdahl	University of Bergen, Norway
Andrew Berry	ResMed Inc., Australia
Aniruddha Gokhale	Vanderbilt University, USA
Antonio Vallecillo	Universidad de Málaga, Spain
Artem Polyvyanyy	University of Melbourne, Australia
Asif Qumer Gill	University of Technology, Australia
Axel Korthaus	Swinburne University of Technology, Australia
Barbara Weber	University of St. Gallen, Switzerland
Ben Roelens	Open University of The Netherlands, The Netherlands & Ghent University, Belgium
Benjamin Yen	The University of Hong Kong, Hong Kong, China
Carlos L. B. Azevedo	Federal Institute of Espírito Santo, Brazil
Chiara Di Francescomarino	Fondazione Bruno Kessler-IRST, Italy
Christian Zirpins	Karlsruhe University of Applied Sciences, Germany
Claudenir M. Fonseca	Free University of Bozen-Bolzano, Italy
Claudio Di Ciccio	Sapienza University of Rome, Italy
Colin Atkinson	University of Mannheim, Germany
Cristine L. Griffo	Free University of Bozen-Bolzano, Italy
Dimka Karastoyanova	University of Groningen, The Netherlands
Dominik Bork	TU Wien, Austria
Fatih Turkmen	University of Groningen, The Netherlands
Fethi Rabhi	University of New South Wales, Australia
Flavia Santoro	Rio de Janeiro State University, Brazil
Florian Matthes	Technical University of Munich, Germany
Frank Leymann	University of Stuttgart, Germany
Frederik Gailly	Ghent University, Belgium
Georg Grossmann	University of South Australia, Australia
Georg Weichhart	PROFACTOR GmbH, Austria

Giuseppe Di Lucca	University of Sannio (RCOST), Italy
Guido Governatori	CSIRO, Australia
Hans Weigand	Tilburg University, The Netherlands
Henderik A. Proper	Luxembourg Institute of Science and Technology & University of Luxembourg, Luxembourg
Hiroshi Miyazaki	Keio University, Japan
Irina Rychkova	Centre de Recherches en Informatique & University Paris 1 Pantheon-Sorbonne, France
Jaap Gordijn	Vrije Universiteit Amsterdam, The Netherlands
Jan Øyvind Aagedal	Equatex, Norway
João Moreira	University of Twente, The Netherlands
João Paulo A. Almeida	Federal University of Espírito Santo, Brazil
John Mylopoulos	University of Ottawa, Canada
José Raúl Romero	University of Cordoba, Spain
Julio C. Nardi	Federal Institute of Espírito Santo, Brazil
Julius Köpke	Alpen-Adria-Universität Klagenfurt Institute for Informatics Systems, Austria
Lam Son Lê	HCMC Tech, Vietnam
Ljiljana Brankovic	University of Newcastle, Australia
Luis Ferreira Pires	University of Twente, The Netherlands
Luise Pufahl	TU Berlin, Germany
Madhusi Bandara	University of Technology Sydney, Australia
Manfred Reichert	University of Ulm, Germany
Marco Montali	Free University of Bozen-Bolzano, Italy
Maria Teresa Gómez López	University of Seville, Spain
Maria-Eugenia Iacob	University of Twente, The Netherlands
Marten van Sinderen	University of Twente, The Netherlands
Mathias Weske	HPI & University of Potsdam, Germany
Mattia Fumagalli	Free University of Bozen-Bolzano, Italy
Michael Schrefl	University of Linz, Austria
Michael Rosemann	Queensland University of Technology, Australia
Nicolas Herbaut	Université Paris 1 Panthéon-Sorbonne, France
Oscar Pastor	Universitat Politècnica de València, Spain
Paolo Ceravolo	Università degli Studi di Milano, Italy
Peter Bernus	Griffith University, Australia
Peter F. Linington	University of Kent, UK
Pierluigi Plebani	Politecnico di Milano, Italy
Pontus Johnson	KTH Royal Institute of Technology, Sweden
Rainer Schmidt	Munich University of Applied Sciences, Germany
Rajeev Raje	IUPUI, USA

Remco Dijkman	Eindhoven University of Technology, The Netherlands
Renata Guizzardi	University of Twente, The Netherlands
Ronny Seiger	University of St. Gallen, Switzerland
Ruth Breu	Research Group Quality Engineering, Italy
Rüdiger Pryss	University of Würzburg, Germany
Sagar Sunkle	Tata Consultancy Services, India
Saïd Assar	Institut Mines-Telecom Business School, France
Schahram Dustdar	Vienna University of Technology, Austria
Selmin Nurcan	Université Paris 1 Panthéon-Sorbonne, France
Sharmistha Dey	Griffith University, Australia
Simon Hacks	University of Southern Denmark, Denmark
Stefan Tai	TU Berlin, Germany
Stefanie Rinderle-Ma	Technical University of Munich, Germany
Sylvain Hallé	Université du Québec à Chicoutimi, Canada
Tiago P. Sales	Free University of Bozen-Bolzano, Italy
Ulrich Frank	University of Duisburg Essen, Germany
Ulrik Franke	RISE, Sweden
Uwe Zdun	University of Vienna, Austria
Vinay Kulkarni	Tata Consultancy Services Research, India
Wolfgang Maass	Saarland University, Germany
Yigal Hoffner	Shenkar College of Engineering and Design, Israel
Zoran Milosevic	Deontik & Best Practice Software, Australia

Additional Reviewers

Barat, Souvik	Khorshidi, Samira
Berry, Andrew	Kuhn, Peter
Bühler, Fabian	Lichtenstein, Tom
Cremerius, Jonas	Murturi, Ilir
de Alencar Silva, Patrício	Muñoz, Paula
Dexe, Jacob	Peregrina Pérez, José Antonio
Gamage, Dimuthu	Ristov, Sasko
Guy, Ed	Roychoudhury, Suman
Gökstorp, Simon	Yilmaz, Fatih
Kaczmarek-Heß, Monika	Yussupov, Vladimir
Katsikeas, Sotirios	

Keynote Speeches

“We Are All Digital Now”: Platform Systems and Flow Processes in Modern Enterprises

Carliss Y. Baldwin

Harvard Business School, Boston, USA
cbaldwin@hbs.edu

In the last 25 years, modern enterprises have become centered on digital systems. IT applications, knit together by enterprise architectures, now pervade all business functions. But there is still doubt and confusion as to how a firm’s digital infrastructure should be managed and maintained. On the one hand, modern enterprises must be flexible, capable of generating a stream of new products and providing customers with numerous options. On the other hand, they must be efficient providing timely solutions at low cost.

These contrasting requirements are exemplified by two opposing technical paradigms: flow processes and platform systems. In the talk, I will characterize flow processes and platform systems in terms of their innate properties and organizational implications. I will also indicate when and why each patterns is more valuable. In modern establishments, the two patterns are not mutually exclusive: technologically sophisticated organizations must use both. At the end of the talk, I will speculate on where each pattern is likely to be needed and invite comments on the tensions they are likely to cause within organizations.

Smart Modeling of Smart Software

Jordi Cabot

ICREA – Universitat Oberta de Catalunya, Barcelona, Spain
jordi.cabot@icrea.cat

There is an increasing demand for embedding intelligence in software systems as part of its core set of features both in the front-end (e.g. conversational user interfaces) and back-end (e.g. prediction services). This combination is usually referred to as AI-enhanced software or, simply, smart software.

The development of smart software poses new engineering challenges, as now we need to deal with the engineering of the “traditional” components, the engineering of the “AI” ones but also of the interaction between both types that need to co-exist and collaborate.

In this talk we’ll see how modeling can help tame the complexity of engineering smart software by enabling software engineers specify and generate smart software systems starting from higher-level and platform-independent modeling primitives.

But, unavoidably, these models will be more diverse and complex than our usual ones. Don’t despair, we’ll also see how some of these same AI techniques that are making our modeling life challenging can be turned into allies and be transformed into modeling assistants to tackle the engineering of smart software with a new breed of smart modeling tools.

Modelling Ethical and Legal Norms/Explaining Compliance and Violation

Giovanni Sartor

¹ University of Bologna, Bologna, Italy
jordigiovanni.sartor@unibo.it
² European University Institute, Fiesole, Italy

Automation-supported compliance checking has become necessary in increasingly automated socio-technical contexts. AI & law research, since the 70's has addressed ways to model ethical and legal knowledge, and has developed approaches that are relevant to compliance-checking.

I will shortly review approaches to the modeling of legal content: Rules- and logic-based models; Argumentation-based models; Case-based reasoning models.

I will address some recent approaches aimed at providing logical models in a way which is understandable to non-technical people, and consider whether this idea may support developments in automated compliance checking. I will also consider the significance of argumentation-based models and ontologies to provide rationales for compliance assessments.

I will argue for the construction of human-understandable models of law and ethics, to be used for the purpose of compliance checking, also over the functioning of machine-learning based systems. How to integrate logical modeling and machine learning, in eliciting and applying normative knowledge is a challenging task for the future.

Tutorial

Business Process Modeling and Simulation with DPMN

Gerd Wagner

Department of Informatics, Brandenburg University of Technology,
Cottbus, Germany
G.Wagner@b-tu.de

The Business Process Modeling Notation (BPMN) has been successfully established as the defacto standard in Business Process (BP) Management. However, BPMN does not have a convincing formal semantics and lacks several important elements needed for BP simulation. BPMN is also not well-aligned with the Processing/Queuing Network paradigm of Operations Research (OR) and the related business process simulation paradigm pioneered by the Discrete Event Simulation (DES) languages/tools GPSS and SIMAN/Arena. The Discrete Event Process Modeling Notation (DPMN) is based on the Object Event Modeling and Simulation (OEM&S) paradigm and on Event Graphs (Schruben 1983), which capture the event scheduling paradigm of DES. DPMN supports modeling resource-constrained activities (with resource roles and resource pools) in Activity Networks and Processing Networks, as well as basic agent concepts (perception, action, communication) and agent-based BP modeling.

Contents

Enterprise Security

Computable Consent – From Regulatory, Legislative, and Organizational Policies to Security Policies	3
<i>Zoran Milosevic and Frank Pyefinch</i>	
A Multi-level Cyber-Security Reference Model in Support of Vulnerability Analysis	19
<i>Simon Hacks, Monika Kaczmarek-Heß, Sybren de Kinderen, and Daniel Töpel</i>	
Security Ontologies: A Systematic Literature Review	36
<i>Malina Adach, Kaj Hänninen, and Kristina Lundqvist</i>	

Enterprise Architecture

Model-Based Construction of Enterprise Architecture Knowledge Graphs	57
<i>Philipp-Lorenz Glaser, Syed Juned Ali, Emanuel Sallinger, and Dominik Bork</i>	
Enterprise Architecture Management Support for Digital Transformation Projects in Very Large Enterprises: A Case Study at a European Mobility Provider	74
<i>Oleg Kanin and Paul Drews</i>	
Interoperability of Digital Government Services: A Brazilian Reference Architecture Model to Promote Communication, Management, and Reuse of Solutions	91
<i>Adriana Xisto, Felipe Sommer, Marcus Vinicius Costa, José Lutiano Costa da Silva, Claudia Cappelli, and Vanessa Nunes</i>	

Business Process Modeling and Monitoring

Modeling, Executing and Monitoring IoT-Driven Business Rules with BPMN and DMN: Current Support and Challenges	111
<i>Yusuf Kirikkayis, Florian Gallik, and Manfred Reichert</i>	
Enhanced Transformation of BPMN Models with Cancellation Features	128
<i>Giorgi Lomidze, Daniel Schuster, Chiao-Yun Li, and Sebastiaan J. van Zelst</i>	

Next-Activity Prediction for Non-stationary Processes with Unseen Data
Variability 145
Amolkirat Singh Mangat and Stefanie Rinderle-Ma

Business Process Mining and Discovery

On the Origin of Questions in Process Mining Projects 165
*Francesca Zerbato, Jelmer J. Koorn, Iris Beerepoot, Barbara Weber,
and Hajo A. Reijers*

Extracting Business Process Entities and Relations from Text Using
Pre-trained Language Models and In-Context Learning 182
Patrizio Bellan, Mauro Dragoni, and Chiara Ghidini

Discovering Sound Free-Choice Workflow Nets with Non-block Structures 200
Tsung-Hao Huang and Wil M. P. van der Aalst

Shape Your Process: Discovering Declarative Business Processes
from Positive and Negative Traces Taking into Account User Preferences 217
*Federico Chesani, Chiara Di Francescomarino, Chiara Ghidini,
Giulia Grundler, Daniela Loreti, Fabrizio Maria Maggi, Paola Mello,
Marco Montali, and Sergio Tessaris*

Process-Driven Applications

Semi-automated Test Migration for BPMN-Based Process-Driven
Applications 237
Konrad Schneid, Sebastian Thöne, and Herbert Kuchen

Splitting Quantum-Classical Scripts for the Generation of Quantum
Workflows 255
Daniel Vietz, Johanna Barzen, Frank Leymann, and Benjamin Weder

Author Index 271

Enterprise Security



Computable Consent – From Regulatory, Legislative, and Organizational Policies to Security Policies

Zoran Milosevic^(✉)  and Frank Pyefinch

Best Practice Software, Brisbane, Australia
{zoran.milosevic, frank.pyefinch}@bpssoftware.net

Abstract. Consumer-facing health applications are increasingly requiring flexible approaches for expressing consumer consent preferences for the use of their health data across multiple providers, and across cloud and on-premises systems. This and the recognition of the need for clear governance and legislative rules that specify enforceable policies over how consumer data is used by the nominated and other providers, including AI vendors, increasingly require machine readable, i.e. computable consent expressions. These expressions can be regarded as additional constraints over security policies, applicable to all stakeholders, while accommodating rules from regulatory and legislative policies. Support for both kind of policies contribute to improving consumer trust in the use of their data. This is applicable to both care delivery processes but also research projects, such as clinical trials. This paper proposes a computable consent framework and positions it in the context of the new developments within Health Level Seven (HL7®) Fast Health Interoperability Resources (FHIR®) standard. The proposal is based on the use of precise policy concepts from the ISO/ITU-T RM-ODP (Reference Model for Open Distributed Processing) standard. The aim is to provide general standards-based policy semantics guidance to interoperability/solution architects and implementers involved in digital health applications. The framework is driven by consent requirements, while leveraging broader policy input from medico-legal community.

Keywords: Consent · Policy · Interoperability · Health Level Seven (HL7®) · Fast Health Interoperability Framework (FHIR®) · RM-ODP · Digital health

1 Introduction

There are increasing number of initiatives aimed at engaging consumers in active participation in their healthcare, as part of the delivery of more effective, quality and evidence-based health care. One way of doing this is through new *digital health services* such as mobile applications or portals. They allow consumers pro-active participation in health-care processes, spanning primary health care providers such as general practitioners and specialists, hospitals, and research institutions [12].

These services rely on the timely and effective access to consumer *health data* which can be shared in controlled way with relevant providers involved in delivery of health care or developing new health knowledge or solutions. Sharing of data is becoming increasingly possible due to the growing adoption of the HL7 standard, Fast Healthcare Interoperability Resources (FHIR®) [1], which allows better sharing across organizational boundaries, supporting new *interoperability* solutions at scale.

It is natural that the individual health information, with its additional confidentiality and privacy constraints, requires that consumer privacy preferences are respected, including *consumer data rights*. This is needed to ensure consumers *trust* in how their data is used, for the benefits of their health care, but also in support of clinical research. The central element here is clear understanding of the *policies* surrounding consumers *consent*, which capture their preferences for what actions are allowed (or not) when accessing or sharing their data. These policies are guided by overarching legislative, regulative, corporate and security policies, for the use and sharing of health information, as for example documented in the Royal Australian College of General Practitioners (RACGP) guidelines [13]. This complex set of rules requires increasing automation in handling policies, including consent, which are currently predominantly paper based. This paper provides a proposal for expressing such policies in a machine interpretable, i.e. computable, manner, grounded in the latest approaches to the expression of policy semantics. We use the latest FHIR consent proposal, published in the Release 5, namely FHIR Consent Resource [9], as the focus for discussion.

1.1 Problem and Contributions

This paper addressed the problem of expressing computable consent policies reflecting patient preferences, while adopting constraints by the enterprise and security policies. The enterprise policies cover legislative, regulative and corporate rules [2]. This problem is heightened in cloud-based environments used for building FHIR enabled applications across administrative boundaries. The paper provides two main contributions:

- semantic foundations for platform-independent models for consent related policies, supporting both enterprise and security policies
- positioning of the above models in the context of distributed architecture associated with FHIR APIs and its consent information models while accommodating broader policy support for governing data sharing/use across digital health ecosystem.

Next, we present related work in support of automated consent management. Section 2 describes current FHIR Consent resource specification. Section 3 presents the generic computable policy/consent framework. Section 4 discusses the positioning of the generic framework with FHIR. Section 5 discusses future work directions.

1.2 Related Work

There are several research and standardization efforts relating to making certain aspects of consent automated and scalable.

One example are ‘dynamic consent’ approaches, which aim to facilitate more engaged and personalized communications between researchers and participants in a research study, through enabling participants to manage their consent preferences over time. One such solution was recently used in genomic research in Australia [3], which developed a web-based application tool called CTRL (control). CTRL facilitates ongoing participant-led management of their involvement in research, by allowing participants to choose from granular consent options and change consent choices over time (including for future use of their data). Participants can indicate preferences for the kinds of results they would want returned, whether they receive alerts about further research their data is shared to, and their preferred methods of contact.

Another example is a scalable consent framework for electronic health records, developed by San Diego Health Connect, funded by ONC Leading Edge Acceleration Projects in Health IT (LEAP) program [4]. Their work focused on how to use FHIR-based application programming interfaces (APIs) to allow patients to electronically document and share their consent preferences to streamline availability of information relevant to their care. This proof-of-concept project proposed a scalable and decentralized architecture for managing and enforcing patient consents. The emphasis was in supporting relatively straightforward permit or deny type of policies regarding consent, but the growing complexity and sophistication of patients’ control over their health data and their sharing across multiple providers requires more powerful computable consent framework. The solution components and available software however represent the most advanced contribution to the field, while also recognizing that further efforts are required across government and the private sector to build a scalable consent management policy and regulatory architecture.

Further, the HL7® standardization organization has recently published a Consent Management Service [5], leveraging contribution from the LEAP project but also from our earlier proposals [6]. This service is independent of any underlying digital health platform and was influenced by FHIR Consent resource developments [9].

There are also early efforts in better supporting consumers in primary health practice in expressing their consent preferences. One example is providing consent for various kind of communications to patients, such as for reminders of their appointments or clinical events, as is done with the Best Practice Premier on-premise product [16]. Another example is the profiling of the FHIR Consent resource (Version 4), for the My Script List (MySL) component of ePrescribing in Australia [11] to be discussed in Sect. 2.3.

These initiatives, and the FHIR consent standardization (see next section), demonstrate different efforts in automating consent, but do not adopt an agreed modelling framework for expressing consent preferences as computable constraints on behaviour of parties involved in handling consent. This is particularly important when consent is considered in terms of interaction with other constraints that specify a broader set of accountability, responsibility and delegation policies, arising from legislative, regulatory or security policies. This paper provides such a computable policy framework, leveraging stability and credibility of relevant parts of the ISO RM-ODP standards, in support of building systems in which parties’ behaviour can be monitored and enforced by implemented systems.

2 Towards Consent Automation – FHIR Approach

FHIR [1] provides specification of a number of modelling concepts for designing, deploying and operating digital health applications. The semantics of the modelling concepts is grounded in many years of HL7 standardization, while the adoption of the commonly used web technologies for building applications makes FHIR increasingly popular among the development community. Key interoperability features of FHIR are:

- common modelling language concepts referred to as FHIR resources, specified using UML, XML and JSON languages
- API style of application developments, relying on the modern web technologies, to support exchange of data and applications across the web
- controlled extension approaches, to reflect specific domain interests, e.g. different national requirements or application domains – known as FHIR profiles.

2.1 FHIR Consent Resource – Basic Policy and Computable Policy Expressions

FHIR standard recognises the need to have a flexible specification of consent to reflect a wide range of preferences of consumers. FHIR defines consent as [9].

– *A record of a healthcare consumer’s choices or choices made on their behalf by a third party, which permits or denies identified recipient(s) or recipient role(s) to perform one or more actions within a given policy context, for specific purposes and periods of time.*

This definition uses the general concept of an *action* performed by an agent, allowing to capture three type of uses of the Consent resource: a) privacy consent directive, being an agreement, restriction, or prohibition to collect, access, use or disclose (share) information, b) medical treatment consent directive, as consent to undergo a specific treatment (or refusal to it), and c) research consent directive, as an agreement to participate in research protocol and information sharing required. These agreements are provided by a healthcare consumer [grantor] or their personal representative, to an authorized entity [grantee] for an authorized or restricted actions with any limitations on purpose of use, and handling instructions to which the authorized entity must comply [9].

Simple Consent Form. In its simplest form, the Consent resource provides attributes to record the content and the metadata of a consent (either implicit consent as an event or an explicit consent document), enabling consent discovery by indexing, searching, and retrieval of consents based on this metadata. The key attributes are:

- Subject – reference to whom the consent applies (e.g. Patient, Practitioner)
- Grantor – reference to who is granting rights according to the policy and rules (e.g. Patient, RelatedPerson, Practitioner, CareTeam, etc.)
- Grantee - reference to who is agreeing to the policy and rules (e.g. Organisation, Practitioner, RelatedPerson, CareTeam etc.)
- DateTime - when consent was agreed to

- **Manager** – reference to a workflow consent manager (e.g. HealthService, Organisation, Patient, Practitioner etc.)
- **Enforcer** – reference to a consent enforcer
- **Source** - used to record the original consent document either in the form of a pointer to another resource or in the form of an attachment.

Note that the concepts of Patient, Practitioner, RelatedPerson and so on, are other FHIR resource concepts, capturing key properties of these information elements [1].

Support for Computable Consent. A more advanced usage of the Consent resource requires computable expression of privacy preference rules. These rules can be processed by a decision engine to decide whether the given consent permits a specific activity (e.g., sharing the patient information with a requester or enrolling the patient in a research project). There are two mechanisms for recording computable consent:

- the *provision* structure which provides a simple structure for specifying additional exception to the base policy rule (or default policy) which is about permitting or denying particular action; for example, access to patient Electronic Health Record (EHR) is generally not permitted (base rule), except when in emergency, and this hold for 7 days (exception with AND condition)
- the *policy* attribute which provides a more flexible mechanism via referencing a policy coded in a policy language of choice. FHIR does not prescribe a type of policy language to be used, with examples being XACML [7], ODRL [8] (see Sect. 3.2).

Note that each exception in the provision structure can further be refined in a hierarchical manner, but the approach does not provide ways of dealing with conflicts, such as when one exception conflicts with a higher-level exception, e.g. whether a more specific rule overrides a more general rule.

In terms of the consent enforcement options, this can be done using a mix of various access control enforcement methodologies (e.g. OAuth2.0, XACML). This enforcement includes the detailed elements of the privacy consent, such as the rules reflecting which organizational roles have access to what kind of resources (e.g. RBAC, ABAC).

2.2 Link with Smart on FHIR Architecture Pattern

We believe that the computable consent expressions, when available in the FHIR Consent resource, can be used to constrain the security policies for specifying and enforcing access to patient data on an EHR/FHIR server. For example, such policies can guide the use of OAuth2.0 security server to determine whether to issue a OAuth2 token for a client app. In fact, OAuth2 plays a central role in one approach to building FHIR applications, the so-called SMART (The Substitutable Medical Applications and Reusable Technologies) on FHIR. This is an open-source, standards-based API that leverages the OAuth 2.0 to ensure secure, universal access to EHRs [10].

The SMART on FHIR is intended to be used by developers of apps that need to access user identity information or other FHIR resources by requesting authorization

from OAuth 2.0 compliant authorization servers. The apps can be used by clinicians, patients, and other parties, and it provides a reliable, secure authorization protocol for a variety of app architectures, including apps that run on an end-user's device as well as apps that run on a secure server [10]. The SMART on FHIR process begins with a user starting an app requesting authorization from an EHR's authorization server, using *scope* parameters specifying the type of access, i.e. specific information about a patient, e.g. observations and read or write permission. If the authorization server permits this access, it returns an access token to the app, which allows the SMART app to call the FHIR server API, and access particular patient's record on the EHR's FHIR server according to the scope parameters. Smart launch also supports authorization for backend services, allowing their direct connection with an EHR when there is no user involved in the launch process, or when permissions are assigned to the client out-of-band.

Thus, the use of a computable consent policy expression to constraint scope of via consent management service with the SMART on FHIR allows linking enterprise policies from computable consent with the security enforcement approaches of SMART.

2.3 Analysis

There are recent efforts in FHIR Release 5 to improve FHIR Consent resource expressiveness to accommodate computable policies, through reference to computable policy expressions, i.e. provision structure and policy attribute, as mentioned above. New experience developed over initial deployment projects, e.g. LEAP project also suggests integrating their consent architecture with OAuth2. This, plus a broader set of policies surrounding consent, such as the expression of ownership and delegation, motivate us to apply a generic policy framework to consent, in the next section.

There were some initial attempts to use the FHIR Consent resource from an earlier FHIR version (Release 4), and specialized them for specific domain of use, specifically the Australian efforts for ePrescribing [10]. Here, My Script List (MySL) supports recording a) permissions from a patient to access their prescriptions to an Organization and b) for the MySL system to upload the patient's active prescriptions from the script exchange. This involved profiling of modelling elements such as identifier attributes for FHIR Patient resource, reflecting Australian elements such as Medicare, DVA (Department of Veteran Affairs), IHI (individual health identifier), telecom contact details, etc. It is to be noted that the earlier version of FHIR Consent resource did not support inclusion of computable consent expressions, and also used the Consent *scope* attribute to capture different type of consents, namely, the privacy, research or treatment consent, modelled using string datatype. This was certainly a modeling option available at the time, but this approach was not adopted in FHIR Release 5 Consent resource. This allows accommodating richer semantics needed for support of different type of workflows associated with different type of consent, including integration of better monitoring and enforcing of policies applicable to such workflows.

3 Computable Policy Framework

The FHIR Consent resource uses the terms of 'permit' and 'deny' which are constraints on the actions of the parties when they fill the role of grantee. In other words, they are

permissions or *prohibitions* for what the parties are allowed (or not) to do, including additional details such as for how long these conditions may be valid. It is thus possible to express consent in terms of the conditions specified in permissions and prohibitions as special type of policies.

For example, grantee's permissions are obtained through the grantor passing on their permissions, which in effect is the *authorization* for grantee, i.e. giving them ability for actions which otherwise they would not be able to do, i.e. giving them access to grantor's own health data. The authorization also places an *obligation* on the grantor, to ensure that access to the medical record is ultimately enabled, e.g. by passing security credentials to the grantee. Once the grantee has obtained permission, they would also need to satisfy other obligations, such as those arising from their medical duties (e.g. duty of care) and obligations to respect the grantor's privacy and confidentiality.

The concepts of permissions, prohibitions, obligations and authorization are regarded as fundamental types of policy constraints, each of which constrains actions of parties as they fulfill the roles to which these policies apply. Their formal expression is the subject of deontic logic [19] and these are often referred to as *deontic* constraints. They are prescribed by some combination of legislative, regulative of organizational authorities (i.e. policy context), each of which specifies rules of behaviour required to satisfy some objective, business, social or ethical.

Observe that these policy concepts are described in terms of actions, or composition of actions (behaviour), in a way that can be iteratively translated in machine executable statements, or computable expressions. This makes it possible to apply the semantics of the RM-ODP standards [2, 14], which supports formal, and thus computable expressions of such policies, developed for the purpose of building technology independent and interoperable ecosystems. These policy concepts are defined next.

3.1 Modelling Concepts for Policy Rules

The following is a list of several key policy modelling concepts, capturing the deontic and accountability constraints. Further details can be found in [6].

An *obligation* is a prescription that a particular behaviour is required. An obligation is fulfilled by the occurrence of the prescribed behaviour.

A *permission* is a prescription that a particular behaviour is allowed to occur. A permission is equivalent to there being no obligation for the behaviour not to occur.

A *prohibition* is a prescription that a particular behaviour must not occur. A prohibition is equivalent to there being an obligation for the behaviour not to occur.

Authorization is an action indicating that a particular behaviour shall not be prevented. Unlike a permission, an authorization is an empowerment.

Note that *prescription* is formally defined as an action that establishes a rule. Prescriptions provide a powerful mechanism for changing the system's business rules at runtime, enabling dynamic adaptation to respond to business changes and new needs.

The RM-ODP standard provides a pragmatic solution for translating these concepts into components that can be used in support of building enterprise distributed solutions. This is done through the concept of the *deontic token*, which has been developed to support explicit association of deontic constraints with the agent to which these constraints

apply [2, 6, 14]. These are enterprise objects which encapsulate deontic constraint assertions. The holding of the deontic tokens by parties constrains their behaviour. This is a powerful modelling approach because it provides a basis for manipulating deontic tokens, for example, passing them between parties to model delegations, and activation or de-activation of policies that apply to the parties. There are three types of deontic tokens, called *burden*, representing an obligation, *permit*, representing permission and *embargo*, representing prohibition.

In the case of a *burden*, an active enterprise object holding the burden must attempt to discharge it either directly by performing the specified behaviour or indirectly by engaging some other object to take possession of the burden and perform the specified behaviour. In the case of *permit*, an active enterprise object holding the permit is able to perform some specified piece of behaviour, while in the case of *embargo*, the object holding the embargo is inhibited from performing the behaviour [13].

The deontic concepts above serve as primitives for expressing various type of accountability, such as the concepts of delegation, commitment and rights. Further, the organizational, regulatory or legal policies are defined within their corresponding contexts, which can be formally expressed by the RM-ODP concept of *community*. A community defines how a set of participants should behave in order to achieve an objective, through the interactions between roles and the policy constraints that apply to them. These participants (or enterprise objects in RM-ODP terms) fulfill *roles* in a community, and thus accept policy constraints that apply to the roles, as stated in the contract for community. At any point in time, at most one enterprise object can fulfil a community role. A community specification may include a number of role instances of the same type, each fulfilled by a distinct enterprise object, with the constraint on the number of roles of that type that can occur, e.g. maximum number of patients in a ward.

3.2 Policy Language Options

The modelling concepts above are used as a basis for designing an architecture in support of the specification and dynamic management of policies. The form of policy rule expressions that is embedded in each of the deontic tokens and other concepts, and which would need to be referenced by FHIR Consent resources, is not prescribed by the RM-ODP standard.

In our previous work [6] we have proposed a **generic policy language**, that is informed by the RM-ODP standard and with the following form:

$\langle \text{policyContext} \rangle \langle \text{Activation} \rangle \langle \text{role} \rangle \langle \text{modality} \rangle \langle \text{eventpattern} \rangle \langle \text{targetrole} \rangle \langle \text{violation} \rangle$

$\langle \text{policyContext} \rangle$ denotes context of policy, such as legislative or organisational source of policies, for which the *community* can be used, as introduced above.

$\langle \text{Activation} \rangle$ specifies trigger *events* for dynamic activation of normative policies; these can be temporal events such as timeouts, other events such as violation of other policies, or accountability actions, e.g. prescriptions or delegations;

$\langle \text{role} \rangle$ denotes a *community role*, to which deontic modality and behavioural constraints apply (defined by the community context);

$\langle \text{modality} \rangle$ denotes *deontic modality* that applies to the party fulfilling a community role, e.g. an obligation, permission or prohibition;

<eventPattern> specifies the expected behaviour of a party in terms of their actions and other occurrences such as timeouts;
 <targetRole> denotes a community role that can be affected by the actions of the subject roles, and included as part of deontic modality;
 <violation> condition which specifies other policies which can be triggered in response to a violation of the primary deontic modality.

So, privacy consent type or template for accessing consumer record can then be:

<ConsentContext> <consentActivation> <grantor> <permission>
 <accessConsumerRecord> <grantee> <violation>

accessConsumerRecord specifies an event pattern, e.g. the start and end of an interval for which the consent was given and its purpose, for example, access to a specific IT resource. This general consent statement can be instantiated for a specific consent policy instance. Thus, the consent statement: ‘A consumer John grants permission to an emergency clinician to access his EHR record, in case of emergency’.

<EDcare> <emergencySituation> <John> <permission> <accessEHRRecord>
 <accreditedEmergencyClinician <>

This policy is activated by emergencySituation event, selected from a set of possible triggering events that can be pre-defined by a clinical provider or jurisdiction. The policy assumes the existence of patient identifier framework, for example, Individual Health Identifier in Australia, which would identify the patient John in this case. Note that no violation condition is specified here.

There are several **specific policy languages** as targets for this generic language. They are selected to reflect event-condition-action pattern (suitable for real-time monitoring) while addressing deontic constraints semantics, as introduced next.

XACML (eXtensible Access Control Markup Language), is a security language for providing a declarative fine-grained, attribute-based access control policy language. Each policy is defined in terms of rules, the evaluation of which provides Boolean permit/deny decision to a particular action or resource. XACML adopts the IETF’s architecture for policy management, with Policy Decision Point (PDP) evaluating policies against access requests provided by Policy Enforcement Points (PEP). XACML defines obligation as a directive from the PDP to the PEP on what must be carried out before or after an access is approved. XACML is suitable for expression of access control following the pessimistic style of enforcement but is not suitable for more flexible approaches to expressing optimistic enforcement options, where certain policy breaches are allowed to occur, once they are detected. Optimistic approaches allow for resolution through mediation mechanisms, such as negotiation. This means having a flexible way of dealing with violations of obligations, such as invoking other corrective policies.

Open Digital Rights Language (ODRL) to some extent address this limitation of XACML for consent management enforcement. In ODRL policies are used to represent permitted and prohibited actions over a certain asset, across two predefined roles called ‘assignee’ and ‘assigner’. There is also support for obligations, through the concept of ‘duty’. Recent experience with ODRL however reports significant limitations in dealing with the dynamics of policies [15]. For example, there is a no mechanism that

would support a patient’s revocation of their consent at any given time, there is semantic ambiguity in the concept of duty, and delegation approach using ‘transfer’ action leads to difficulty with the expression of delegation options in which grantor would allow delegating permission but still keeping it’s own permission [15].

Business Contracts Language (BCL) may best support the general language requirements [6], in part because it is grounded in the semantics of RM-ODP standard concepts, both for the behavioural and policy semantics, as presented in the previous section. As a result, BCL language would have similar structure as the general policy language above. BCL includes the concept of community template, serving as a context for the definition of roles, which specify expected behaviour of parties, including the applicable deontic constraints. BCL uses event patterns to specify triggering, behavioural and violation conditions for the policy language. BCL back-end components are implemented in Java and use contemporary software to implement interfaces, including Web-based technologies. The language can be used to specify monitoring conditions for obligations and thus support the optimistic style of enforcement. This out-of-band real-time monitoring of activities of the parties against policy rules provides many benefits, such as faster reaction to important events that might signify occurrence of medical conditions requiring action or detecting potential breaches of policies. This is typically done by a trusted third party in the role of a monitor. Once the monitor detects a breach, it can invoke discretionary or non-discretionary enforcement options.

Consider the privacy consent community introduced earlier. The snippet of the BCL below shows how the consent for cancer research can be represented:

```
CommunityTemplate: CancerResearch
ActivationSpecification: IndividualConsentDirectiveSigned
Policy: PrivacyConsentResearch
Role: Individual
Modality: Permission
TargetRole: accreditedResearcher
Condition: On CancerResearchStart [NOT MentalData]accessEHRRecord
```

The above snippet uses the guard over the EHRRecord data to ensure that access to mental health data from the patient personal health is not possible. Another option would be to specify a prohibition policy over the same data, with the same effect.

One disadvantage of BCL is that it was developed as a proof of concept, with many examples described in various publications, but there are no available open source implementations yet.

It is to be noted that policy language options introduced above are all declarative in style, which are suitable for the expressions of constraints. They are also independent of the details of widely used implementation languages that can be used to implement their functionality such as needed for event-based monitoring of policy expressions. It is expected that each deployment environment will dictate selection of the implementation languages. Further, the FHIR based DevOps environment might require its own language options, reflecting in the FHIR based tooling available.

3.3 Example – Privacy Consent

Figure 1 depicts the key roles of Grantor and Grantee in a consent community, supported by several other roles needed for consent management [6], listed next.

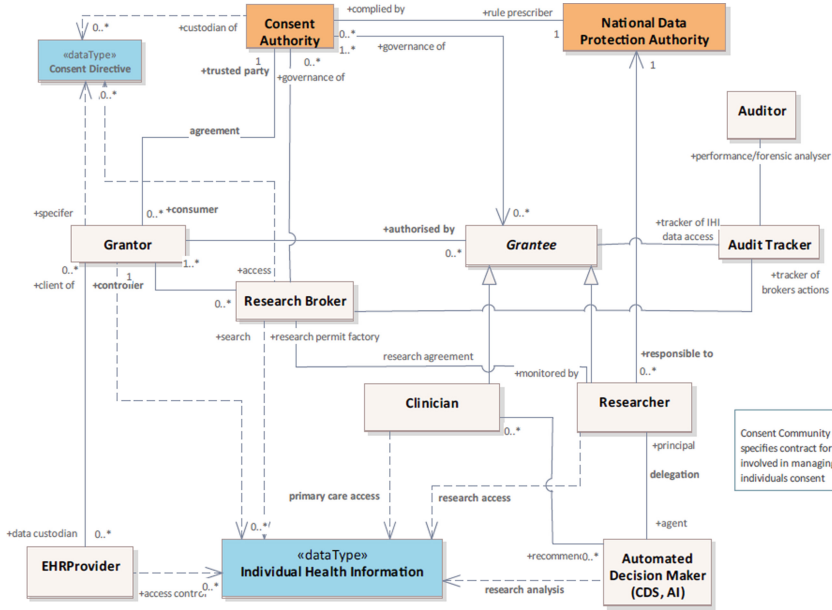


Fig. 1. Consent management community

- Grantor, to be fulfilled by any individual giving consent, possibly respecting other constraints, such as being of legal age, having normal cognitive function etc.
- Grantee, to be fulfilled by professionals with the required credentials, such as Clinician, permitted to access Grantor’s individual health information for care purposes, or Researcher, permitted to access Grantor’s de-identified health data for research purpose, and with an obligation not to perform re-identification of patient data.
- Consent Authority, a trusted party responsible for storing individuals’ consents and overseeing the consent agreement rules.
- Research Broker (Broker from here on), a legal entity authorized to search patient health and consent data to identify patients suitable for research study, e.g. cancer research. The Broker is responsible that patient preferences are enforced.
- National Data Protection Authority, responsible for defining and enforcing data protection policies.
- *Electronic Health Record (EHR) provider*, custodian of individuals’ personal health data in their EHR records. They are usually prohibited by law from releasing patient data without consent, except when a clinician is providing emergency care.

- *Automated Decision-Maker*, performing analytics, recommendations and in some cases, active decision-making, augmenting activities of clinicians or researchers; this role can be fulfilled by clinical decision support systems or AI systems.
- *Audit Tracker*, logging actions of clinicians and researchers to generate audit trails, which can be used for subsequent activity analysis, e.g. by an Auditor;
- *Auditor*, providing analysis of event traces to support performance analysis or forensic investigations, such as detecting breaches of clinicians accessing healthcare records outside of them providing care.

The scenario below illustrates a normal sequence of *actions* from the time an individual gives consent until their data is used by researchers.

1. Grantor updates their consent directive at the Consent Authority allowing their de-identified genomic data to be used for cancer research, excluding mental health data
2. Grantor permits Broker matching on their de-identified data, needed to retrieve the identifiers of those patients whose consent matches the research study parameters. It does not give them access to the health data, just search and retrieve identifiers.
3. Researcher contacts the Broker stating their interest in conducting research across all patients who have given consent for cancer research; this includes access to their medications, treatment and genomic information.
4. Broker provides a de-identified list of eligible patients to the Researcher and gives an authorization for them to access de-identified patient data from the EHR provider, with the exclusion of medication data related to mental health treatment. The authorization requires the Researcher to maintain an audit trail of all data access.
5. Researcher retrieves de-identified patient data from the EHR provider. The EHR provider filters the data as required to comply with individual patient consent directives and lodges an audit record relating to the released data with the Auditor.
6. Researcher accesses the EHR data for their research, lodging an audit record for each access with the Audit Tracker; an AI system must also lodge access, as it acts on behalf of the researcher using their authorization (a research permit token).
7. Researcher publishes the result of the research and informs all relevant parties.
8. At a later point, a patient suspects that their mental health data were used by a health insurer and then contacts the Consent Authority to lodge a complaint.
9. Consent Authority engages Auditor who accesses audit trail to perform forensic investigation of patient's data access by Grantees. Upon detection of a violation, it notifies an enforcer to apply penalty to either party (not shown in this diagram).

The following are examples of *policies* for the community roles and their actions:

- Permission of the Grantor to the Broker to search patients' data and if it satisfies researcher criteria include a link to this data in a data set for the Researcher.
- Obligation on the Audit Tracker to log data access by the Grantee reliably and on-time and provide access to the audit trail by the Auditor; the Tracker may also have an obligation to log actions of Broker which may be needed for forensic purpose.

- Authorization of the Grantor to the Researcher to access the Grantor’s individual health information, as follows.
 - Grantor first authorizes (issues permit to) the Broker for searching their data to establish whether they satisfy research question criteria.
 - Broker then issues a research permit to the Researcher which includes a list of Grantors who provided consent to access their de-identified health data. Note that the Researcher might pass this permit to an AI system, delegating computations
 - EHR provider then allows access permit to the Researcher to access health records of specific patients, provided Researcher has credentials requested by the EHR provider; this can rely on the use of Smart On FHIR backend launch (see Sect. 2.2).

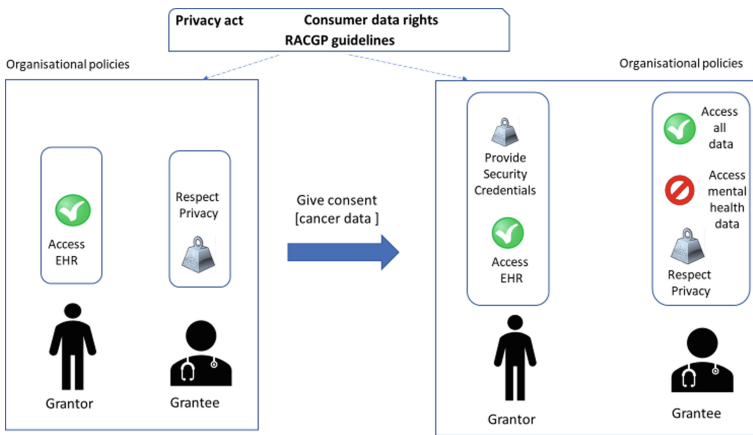


Fig. 2. The dynamics of deontic tokens before and after giving consent

Authorization is modelled using a combination of permit and burden tokens. For example, authorization of the Grantor to the Broker above, involves the permit passed from the Grantor to the Broker to search its record but also places an obligation on the Grantor itself, through the corresponding burden, to ensure that access to its record is ultimately enabled (e.g. by providing security credentials). This authorization changes the deontic state of both the Grantor and Grantee, the effect of which is that the Grantor’s permit to the Broker to search its healthcare data is passed on to the Researcher.

Figure 2 depicts how the consent action changes deontic states of Grantor and Grantee, in terms of different deontic tokens of the agents, before and after the action, while in compliance with legislation, regulatory and organizational polices. Note that data protection rules defined by a National Data Protection Authority set accountability and legal responsibility for researchers in using health data. These rules were established through *prescription* actions of the Authority, establishing obligations and permissions for all parties when accessing patient data in this community.

4 Positioning with FHIR Ecosystem

The computable consent framework, consisting of all deontic and accountability modelling components from Sect. 3.1, can be integrated with a FHIR application ecosystem, as shown in Fig. 3. This includes the integration of FHIR resources with the burden and permit objects, which are associated with the actions of community roles. Some of these deontic constraints are result of the policies prescribed by regulators or other authorities, and others are dictated by the security policy mechanisms of the undelaying platforms, such as the access token of the Oauth2.0, which can be regarded as being a special kind of Permit.

The deontic tokens representing deontic constraints, can be accessed by or transferred with the data associated with processes in a consent community. The interpretation of policy language expression fragments that they carry (depicted as PLEs in the figure), can be executed by a policy engine (i.e. consent policy engine).

It is through these deontic token objects and policy language expressions that computable policy statements can be evaluated and enforced. In the FHIR application ecosystem, FHIR Consent can be modelled as a combination of consumer permits and providers burdens, which, when embedding a computable policy language of choice, such as XACML, ODRL or BCL, can be used as the target from the policy attribute specified in the FHIR consent resource (see Sect. 2.1).

Similarly, a FHIR Contract resource stating rules for sharing data and services across partners, can be described in terms of the burdens associated with each party, reflecting the contract conditions, again described in a policy language of choice.

Recall that the policy framework above provides a solution in support of the dynamics of passing permits and burdens across parties in a system as well as creating new deontic tokens to constrain actions of the parties. This supports quite a general way of expressing accountability, ownership, creation/change of new policies, which surround consent to broader controlled data exchange. These token objects can also provide traceability to strong security mechanisms, such as for example when using Oauth2 authorization, of XACML and RBAC access control.

There are many other FHIR resources that use or are referenced by the FHIR consent. In our example, and in relation to a typical clinical trial research, a FHIR ResearchSubject resource can be used to model a party filling a grantor in the related research study (modelled as FHIR ResearchStudy resource). Here, an agreement between the EHRProvider, Broker, and any other third parties, such as the Broker or Automated Decision Maker community roles, can be specified using FHIR Contract (see Fig. 3).

The figure also depicts a generic policy editor which can be used to create consent forms, and consent templates, and the FHIR Questionnaire and QuestionnaireResponse resources can be used for this purpose.

The FHIR Provenance resource, leveraging W3C provenance specification [18], can be used to manage the tracking of the changes to the Consent. Further, FHIR DocumentReference can be used as an attachment to show the stages of consent with additional or updated document(s) attached at each stage. The Contract resource can be used like a Document Reference where, as signatures are gathered or conditions applied, the Contract can be updated and attached to the Consent. In general, the Contract resource

represents a legally enforceable, formally recorded unilateral or bilateral directive i.e., a policy or agreement [1].

FHIR AuditEvent resource can be used to support the operations associated with AuditTracker and Auditor roles in the consent community.

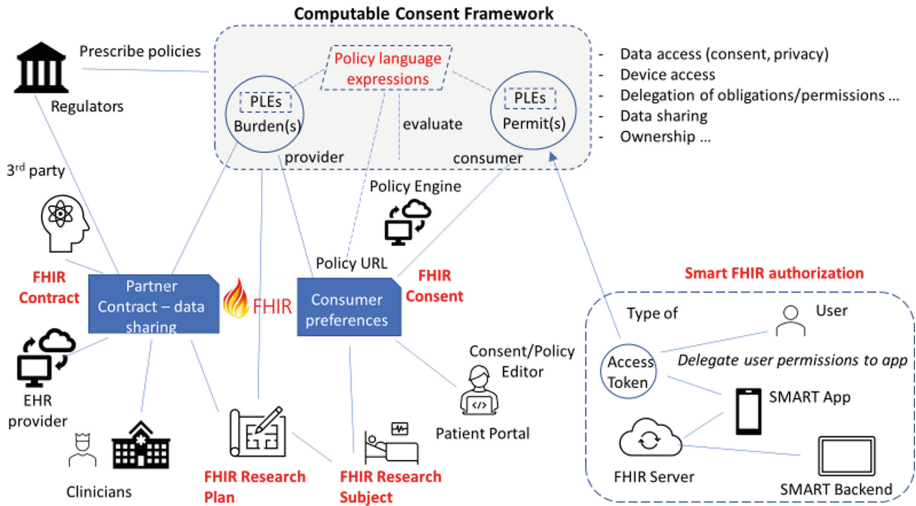


Fig. 3. Computable consent framework integration within FHIR ecosystem

5 Conclusion and Future Work

This paper has proposed a computable policy framework that can be used in cases when relevant FHIR applications may require a domain language for expressing legislative, regulative and organizational policies - in a way that can be processed by machines, interpreted, and used to invoke security policy components, such as Oauth2 authorization or role base access control. The paper focuses on the variety of policies surrounding patient consent, both privacy and research consent, including those that are defined by relevant authorities, such as the RACGP’s policies for managing health information and privacy in general practice [13].

Our future work will aim at implementing this computable policy framework in a FHIR application ecosystem such as Azure FHIR server [17], in primary health care context [12]. The first step is defining an overall architecture for consent management enforcement, making use of FHIR resources, followed by its implementation using FHIR based tools, patterns and implementation guides. The architecture could accommodate the components in Fig. 3 but also additional component such as policy editors, consent forms and integration with SMART on FHIR launch. We also plan to give a better account of actions, known as *speech acts* in the RM-ODP enterprise language [14], needed for expressing delegation, authorization and commitment. The second step is to select a policy language of choice, that best reflects the policy semantics above and investigate

its mapping into a suitable implementation language, used in the FHIR community, such as Java, C# or Python.

We also plan to discuss these issues with medico-legal practitioners to ensure that it is a legally verified approach, as well as ethics specialists to help inform building applications in which potential policy conflicts arise. Finally, we hope that this proposal may be of interest for future standardization of the FHIR Consent resource.

Acknowledgments. We would like to thank our colleagues from Best Practice Software, especially Daniel Kerridge, Gina Clement and Anthony Lee, for providing valuable input to this paper.

References

1. Health Level Seven (HL7®) Fast Health Interoperability Framework (FHIR®). <https://build.fhir.org/index.html>. Accessed 08 Aug 2022
2. Linington, P., Milosevic, Z., Tanaka, A., Vallecillo, A.: Building Enterprise Systems with ODP, An Introduction to Open Distributed Processing. Chapman Hall/CRC Press (2011)
3. Haas, M.A., Teare, H., Prictor, M., et al.: ‘CTRL’: an online, dynamic consent and participant engagement platform working towards solving the complexities of consent in genomic research. *Eur. J. Hum. Genet.* **29**, 687–698 (2021)
4. Scalable Consent Framework for the Advancement of Interoperability with FHIR-based APIs. <https://www.healthit.gov/topic/2019-leap-health-it-projects#Scalable>. Accessed 08 Aug 2022
5. HL7 International, Services Functional Model: Consent Management Service, Release 1, January 2021, HL7 STU Ballot
6. Milosevic, Z.: Enacting policies in digital health: a case for smart legal contracts and distributed ledgers? *Knowl. Eng. Rev.* **35**, E6 (2020). <https://doi.org/10.1017/S026988892000089>
7. XACML. <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>
8. ODRL Information Model 2.2. W3C Rec., 15 February 2018. Accessed 08 Aug 2022
9. FHIR Consent resource. <https://build.fhir.org/consent.html>. Accessed 08 Aug 2022
10. SMART App Launch. <http://hl7.org/fhir/smart-app-launch/>. Accessed 08 Aug 2022
11. ePrescribing API, StructureDefinition: MySLConsent. <https://fhir.medicationknowledge.com.au/dev/StructureDefinition-mysl-consent.html>. Accessed 08 Aug 2022
12. Australian Institute of Health and Welfare, Primary health care. <https://www.aihw.gov.au/reports-data/health-welfare-services/primary-health-care/overview>. Accessed 08 Aug 2022
13. The Royal Australian College of General Practitioners. Privacy and managing health information in general practice. RACGP, East Melbourne, Vic (2017)
14. ISO/IEC 15414. Information Technology: Open Distributed Processing, Reference Model, Enterprise Language, 3rd edn. (2015)
15. Kebede, M., Sileno, G., Van Engersa, T.: Critical reflection on ODRL. In: AI Approaches to the Complexity of Legal Systems XI-XII: AICOL International Workshops 2018 and 2020, Revised Selected Papers (2020)
16. Best Practice, BP Premier. <https://bpsoftware.net/bp-premier/>. Accessed 08 Aug 2022
17. FHIR Server for Azure. <https://github.com/Microsoft/fhir-server>. Accessed 08 Aug 2022
18. PROV-DM: The PROV Data Model. W3C Rec, 30 April 2013. Accessed 08 Aug 2022
19. von Wright, G.H.: Deontic logic. *Mind* **60**, 1–15 (1951)



A Multi-level Cyber-Security Reference Model in Support of Vulnerability Analysis

Simon Hacks¹ , Monika Kaczmarek-Heß² , Sybren de Kinderen³ ,
and Daniel Töpel²

¹ University of Southern Denmark, Odense, Denmark
`shacks@mmmi.sdu.dk`

² University of Duisburg-Essen, Essen, Germany
`{monika.kaczmarek-hess,daniel.toepel}@uni-due.de`

³ Eindhoven University of Technology, Eindhoven, The Netherlands
`s.d.kinderen@tue.nl`

Abstract. This paper reports on the second engineering cycle of a reference model for end-to-end cyber-security by design in the electricity sector. In our previous work, we proposed a reference model that relies on the integrated consideration of two fragmented, but complementary, reference models: NISTIR 7628 and powerLang. To align these reference models, we rely on multi-level modeling, specifically on the Flexible Meta Modeling and Execution Language (FMML^x), and integrated modeling and programming. Within this paper, we strengthen the bottom-up design of the reference model's application by integrating a semi-automated threat analysis. This enables the identification of possible points of improvement in the actual architecture design, as well as a future analysis of business-level impact of different threats. To demonstrate our approach, we rely on the well-studied Ukraine scenario from 2016.

Keywords: Vulnerability analysis · Multi-level reference model · Cyber-security by design

1 Introduction

In the era of digital transformation, inherent to the increased reliance on IT is also an increase in IT security weaknesses. Depending on the unit of analysis, such weaknesses can have a considerable effect on all affected by the digital transformation, be it organizations, industries, or regions. For instance, a well-documented example from the electricity sector, used in this paper, is a well-orchestrated cyber-security attack on the Ukrainian electricity grid, targeting the IT infrastructure of a regional electricity distribution company, causing power outages affecting approximately 225,000 households [6, 47].

Considering the increasing frequency and sophistication of attacks on IT infrastructures, it is not enough to react to security threats, but instead, a proactive approach is required [1]. To achieve this, organizations can apply *cyber-security by design*, which demands accounting for cyber-security concerns throughout an entire product life-cycle [14].

Previously [31], we proposed a reference model for cyber-security by design for smart grids, which combines the NISTIR 7628 [35] emphasizing security requirements, use cases and assets [29], with a modeling language for vulnerabilities, attacks, and countermeasures [28]. Thus, by combining relevant but fragmented security concerns, the reference model provides static support for *end-to-end* model-based security analysis. The reference model has been created with FMML^x [12], a multi-level modeling language with respective tool support [8, 11]. This allowed us (1) to naturally model domain hierarchies, e.g., covering general and domain-specific (e.g., the electricity sector) security concepts, and (2) to make the reference model amenable to computational analysis [30, 31]. Please note that a broad variety of knowledge is needed to support organizations in different stages of creating secure systems, e.g., on the power grid and its architecture, on related IT components, and on cyber-security [49]. Therefore, we provided a solution that also enables experts in one of the fields only, and just basic knowledge in the others, to design secure systems. Thus, the developed reference model provides support during all phases: starting from the identification of (security) requirements, through identification of relevant assets, risk assessment, identification of countermeasures, and finally design of a supporting architecture. Whereas the proposed reference model supports top-down security analysis and received positive feedback from domain experts [31], the support for bottom-up security analysis, such as vulnerability analysis, is limited.

As identifying vulnerabilities by security testing is widely applied to assess and improve the security of systems [54]: by conducting attack simulations, one can detect vulnerabilities and evaluate alternative security designs enabling proactive identification of threats [9], therefore, in this paper, we address this gap and aim at equipping the reference model with support for bottom-up assessment of systems' cyber-security, also by non-security experts. Therefore, we aim at (1) extending the reference model with the aspects required for the vulnerability analysis, such as assets' vulnerabilities or defenses, (2) supporting simulations and threats analysis, based on the extended model; and (3) providing initial support to analyze the business impact of the identified threats.

Our work fits squarely within design-science [19] and follows Wieringa's engineering cycle [51]. Specifically, this work covers a full engineering cycle, cf. [51], as a continuation of the outcomes of a first engineering cycle [31]. As such, it can be characterized as follows: (1) Problem identification and treatment design: we equip the earlier proposed reference model, which emphasizes top-down security design, with the capability to support vulnerability analysis. For treatment design, first, we identify a set of requirements pointing to the required extensions to the multi-level model, and, based on the related work, decide to build on

the simulation capabilities of a selected attack modeling language, icsLang [28], and its accompanying software tool, securiCAD [9]. (2) Treatment implementation: we extend the earlier version of the multi-level model, in line with the requirements and the selected modeling language, and, to capitalize upon the simulation capabilities of icsLang, we realize a tool chain between the XModeler and securiCAD. (3) Treatment validation: we perform a lightweight evaluation of our extended model and tool chain in terms of a realistic attack scenario to check the applicability of the proposed solution as well as its utility. Specifically, we use the mentioned cyber-security attack on part of the Ukrainian electricity grid, documented in, e.g., [6,47].

The paper is structured as follows. First, goals and design decisions are discussed in Sect. 2. Then, the extended version of the multi-level reference model and the tool chain are presented in Sect. 3. Next, in Sect. 4, the running scenario is described together with the obtained results. A discussion on the related work follows in Sect. 5. The paper concludes with final remarks.

2 Main Goals, Requirements and Design Decisions

As stated, our aim is to equip the designed multi-level reference model, with additional analysis possibilities, aiming at bottom-up assessment of system's cyber-security by non-security experts. Such a vulnerability analysis requires, 1) information on the system design (i.e., information on assets and associations), relevant security properties (i.e., assets' vulnerabilities and defenses), and on possible attacks (including information on vulnerabilities, their exploits and impact), and 2) the collected information needs to be processed, to uncover relevant weaknesses and possible attack paths [28]. Whereas the multi-level reference model, as presented in [31], accounts for such concepts as asset, asset associations, or attacks, and thus, constitutes a good basis for such an analysis, further extensions are required.

Requirement 1: *Populating the model with information on specific asset types and their security relevant properties.* The multi-level model should not only provide high-level concepts such as asset or vulnerability, but it should also differentiate between different types of assets, their role in the system, and their security-relevant properties. Thus, there is a need to account for, e.g., asset hierarchies and their vulnerabilities assigned to specific asset categories (e.g., vulnerabilities of hardware components, or of software applications), types (e.g., vulnerabilities of routers, or sensors), or specific software/hardware products.

Requirement 2: *Using information processing capabilities allowing to uncover weaknesses.* To identify security threats, possible attack paths need to be identified. This may be done in different ways, the most common is to generate and analyze so called attack graphs (cf. [28,32] and Sect. 5).

Requirement 3: *Accounting for "known unknowns".* The vulnerabilities analysis should also be possible in case little is known about the specific configuration of the system, e.g., when only high-level information about the main

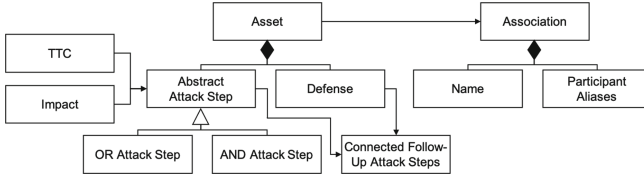


Fig. 1. The meta model of MAL

elements of the system is provided, or when it is known that some asset, such as a sensor, router or firmware is used, however no specific details regarding a specific asset instance are known. Please note that the possibility to account for known unknowns, can make such an analysis easier for non-security/non-IT experts, as less knowledge is necessary.

Requirement 4: *Accounting for the outcomes of the analysis within the model to enable further analysis.* As the conducted threat analysis should enable (1) the identification of possible points of improvement in the architecture design, as well as (2) the analysis on the business-level impact of identified threats, the information from the simulations performed should be incorporated into the multi-level model, and corresponding analyses should be defined.

Considering the requirements and that the required expertise and required data to perform such a cyber-security assessment are rather scarce, cf. Sect. 5, we have decided to capitalize on approaches and attack simulation mechanisms already available. Based on the performed analysis of existing initiatives in the security modeling and analysis field [31], we have accounted for the meta attack language (MAL) [24] in the first engineering cycle of our multi-level reference model [31], with the concrete MAL-based language, namely icsLang [17], as a basis for the required extensions. Although the XModeler is an integrated modeling and programming environment, which would allow to implement the simulation mechanisms, acquiring the needed data (e.g., regarding probabilities of attacks) would be challenging. Therefore, we decide to use icsLang’s accompanying software tool, securiCAD [9]. To be able to capitalize upon distinct features of both platforms, i.e., icsLang/securiCAD and features of multi-level modeling, we realize a tool chain between the XModeler and securiCAD. Thus, after extending the multi-level model, we provide a proof-of-concept implementation of a GraphML exporting/importing capability for the XModeler, so as to realize a two-way chain between the XModeler and the icsLang software tool. We opted for GraphML to rely on an open standard representation of conceptual models, thus easing the reuse of our toolchain also for other types of models.

3 A Multi-level Reference Model in Support of Vulnerability Analysis

In the following, we first provide a short overview of the MAL and icsLang. Then, the description of the extended multi-level model follows. Finally, an overview of the tool chaining is provided.

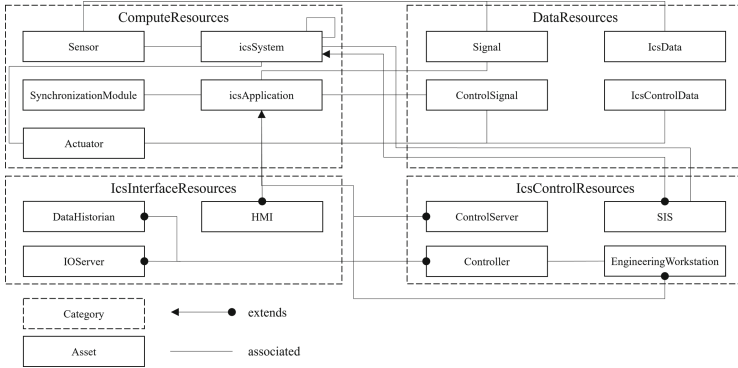


Fig. 2. Excerpt of icsLang including solely additional concepts compared to coreLang.

Meta Attack Language and icsLang: The main elements of a MAL DSL are so called **assets** (cf. Fig. 1), which describe common concepts of the domain under study. An asset is comprised of **attack steps** representing possible attacks that an attacker can perform. **Attack steps** can be connected by an attack path resulting in an attack graph, on which the attack simulation is run. Additionally, each **attack step** can be related to any combination of impact types (i.e., confidentiality, integrity, and availability) to specify the risks and **probability distributions** can be assigned to represent the effort needed to complete the related **attack step** so called time to compromise (TTC). A **defense** is an entity that prohibits connected **attack steps** to be performed if they are enabled. **Assets** have **associations** and related cardinalities between them. Inheritance between **assets** is allowed and each child **asset** inherits all the **attack steps**.

icsLang¹ is an evolution of powerLang [17] and builds upon coreLang² [28], thus accounting for all its concepts. Due to space reasons, we mention further only selected icsLang concepts, for a complete description of coreLang, cf. [28].

icsLang is organized in four different categories, pointing to which assets interact with each other (cf. Fig. 2). Assets within the category of *computing* form the main link to the concepts of coreLang. Thus, icsSystem is adding Industrial Control System (ICS) specific attack vectors to the System of coreLang, that represents resources that run Applications. Similarly, icsApplication adds ICS attack vectors to Application. More specific for the ICS-domain are the concepts of Sensor, Actuator, and SynchronizationModule. A Sensor generates data (via a signal) but may not have any associated software or host, i.e., it is running by itself. An Actuator consumes signals and causes interaction with the physical world. Finally, a SynchronizationModule represents any component

¹ <https://github.com/mal-lang/icsLang/commit/5ad91c85f6a3f45ab0c769e69d7e8cf062af4c69>.

² coreLang 0.4.0.

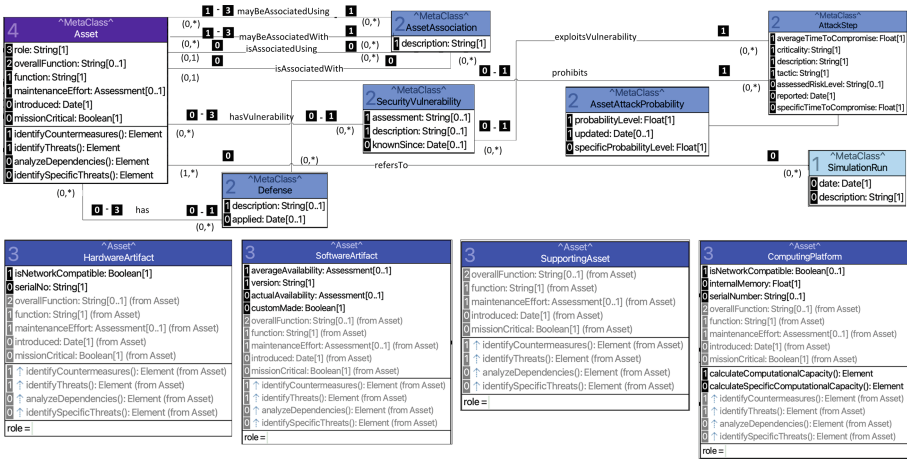


Fig. 3. Excerpt of the multi-level reference model, for overview of FMML^x's concrete syntax, cf. [12]

that provides synchronization capabilities, as for example a GPS receiver or a network-based synchronization module. To interact with each other, the aforementioned assets exchange *data* related assets with each other (i.e., *IcsData*, *IcsControlData*). In case that the medium is not of importance, *icsLang* offers the additional concepts of *Signal* and *ControlSignal*. Finally, a characterizing property of ICS is the *control* as the physical world is monitored and based on the given input parameters decisions are taken. Therefore, *icsLang* provides different assets elaborating on this aspect, such as *Controllers*, *ControlServer*, or *EngineeringWorkstation*. Besides the control functionality, ICS offer different *interfaces*, e.g., *Human-Machine Interface (HMI)*, to close the gap between human and systems, but also between operational and informational technology.

Extended Multi-level Model: Multi-level modeling (MLM) covers any modeling approach that aims to provide systematic support for representing multiple classification levels within a single body of model content [3]. This means that one can employ as many classification levels as needed for expressing the domain knowledge at hand, e.g., to naturally mirror domain hierarchies [30]. In addition, one can defer instantiation, i.e., constrain the instantiation to a model element residing at a specific, not directly proceeding, classification level [10]. Finally, one can relax the strict separation between type and instance [3], allowing one to populate and use a model with instance-level data. Accordingly, MLM comes with multiple promises, which make it particularly suitable to design reference models, cf. [30].

In our earlier work, cf. [31], based on identified requirements, we selected the FMML^x [10] with a meta modeling editor, the XModeler [10], that has an integrated language execution engine, and designed the multi-level reference model accounting for the concepts from NISTIR as well as MAL/powerLang, and

enabling top-down analysis. Now, to enable bottom-up analyses, in-line with the four high-level requirements formulated, further extensions have been performed, among others, accounting for the icsLang’s concepts or defining mappings among the concepts already present in the multi-level model and icsLang.

Figure 3 presents a small excerpt from the extended and implemented multi-level reference model using the XModeler. Please note that for readability purposes, we present only selected concepts, selected attributes and selected operations assigned to different levels of classification. For a detailed description of FMML^x, we refer to [10, 12]. Apart from the “traditional” modeling constructs such as (meta)classes (assigned to different classification levels, cf. the number standing on the left side in the class header), attributes, operations and relationships, it is possible to defer an instantiation of modeling constructs by assigning a so called level of intrinsicness (denoted as a white digit in the black box), which dictates at which level of classification a given property (attribute, operation, or association) will be instantiated.

In line with the concepts of MAL and icsLang, the multi-level reference model accounts for the characteristics and dependencies among Attack steps (**AttackStep**, Level 2 (L2)), involved Assets (**Asset**, L4)³, Countermeasures (Defenses) (L2) and Vulnerabilities (L2). The domain-specific knowledge regarding the above-mentioned aspects can be stated by instantiating the enumerated meta classes and associations between them. Please note that, as we may learn about relevant attack steps, vulnerabilities, or defenses, at different classification levels (e.g., for all **SoftwareArtifacts**, as well as those specific to, e.g., SCADA systems, cf. Sect. 4, or to specific SCADA products, or their installations), a set of associations with different intrinsicness (i.e., their instantiation has been deferred to different levels) needs to be defined, cf. Fig. 3.

While we move along the multi-level hierarchy, we deal with different types of assets and asset-specific associations. Thus, we account for different categories of assets such as hardware, software, computing platform or supporting artefacts (so taking into account mostly the role the said asset is assuming in the system), then specific types are considered on L2, and instantiated into specific types on L1. Here both generic elements of IT infrastructure/Information system may be accounted for (e.g., Routing Firewall, SCADA System), as well as specific products (e.g., a specific router offered by company X), cf. Req. 1. In addition, such a conceptual design of the model allows for addressing Req. 3, i.e., to account for known unknowns within the simulations.

Specifically, to account for known unknowns, firstly, we rely on the ability of MLM to state the relevant knowledge as soon as it is known, on a relevant level of abstraction (e.g., to assign known vulnerabilities to different categories or types of assets already, and not only on the instance level). On the other hand, we introduce on the L1 level generic concepts, which are later specialized into some further types or specific products. For instance, **Router** (L2) is instantiated into **RoutingFirewall** (L1), which is further specialized (within L1 classifica-

³ Please note that the classification level is a rather ‘technical’ term indicating the number of possible instantiation-chains, and not the importance of a given concept.

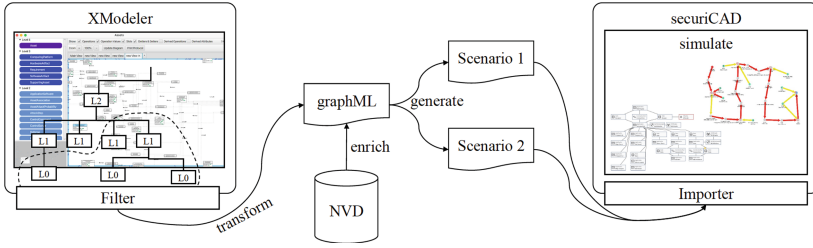


Fig. 4. Implemented prototypical tool chain

tion level) into different products offered by different companies, having some product-specific vulnerabilities and defenses. Now, while modeling the system at hand, we may face the following scenarios: (1) we point to a specific system or hardware component used, and thus instantiate it, and provide some specific information on, e.g., defenses implemented or specific vulnerabilities, (2) we point to a generic concept, thus stating, e.g., that we know that some routing firewall is part of our system, but no further specifics are known. Here we benefit from the already stated vulnerabilities or possible defenses defined on the L1 level. Here also additional information, if known, can be accounted for on the L0 level. Finally, (3) in case we have information regarding possible products used (e.g., the most popular solutions in some domain, or leading products on the market), we may state on the L0 level, that an instance of some generic asset is probably realized by some specific products. In this case, the vulnerability analysis will be conducted for different possible configurations, however, the model of the system needs to be created only once. A similar possibility is provided for while accounting for missing information on existing asset associations: we may point to some generic association type (e.g., application signal, application connection), and within analysis different possible configurations/scenarios will be considered (e.g., bi-directional, transmitted).

To provide a basis for further analysis, cf. Req. 4, and document the results of the conducted simulations, the corresponding concepts `SimulationResult` as well as `Rationale` have been defined. `Rationale` (L2) allows us to capture the reasons standing behind an assessed probability of uncovered weaknesses. The conceptualization of `Rationale` is inspired by notions important to software architecture rationalization, especially [22].

Finally, we incorporate into the model the domain-specific knowledge, among others, in the form of associations with cardinalities and well-formedness constraints. As such, when we design the system the modeling tool will identify missing elements (e.g., each hardware component needs to have a firmware, or each computing hardware needs to have some operating systems). Furthermore, the modeling tool will support us either by linking the modeled object with an already existing one, or by creating a new instance linked to it. This contributes to the higher quality and completeness of the model being created.

Prototype: To demonstrate our approach, we implemented a prototypical tool chain comprised of XModeler⁴, securiCAD⁵, and a set of python scripts⁶. The tool chain consists of five main steps, cf. Fig. 4. After the model of the specific enterprise system has been created by instantiating the concepts of the multi-level model described in the previous section (i.e., assets and their associations), a user may trigger a vulnerability analysis. To this end, first, a GraphML representation of the model that solely contains the technical aspects that can be simulated in securiCAD, is created. Apart from accounting for different asset objects (L0), their links (instantiated assets associations), also information on known vulnerabilities and defenses is accounted for (by gathering relevant information stated on different classification levels, L3–L1). In cases where it is not clear what the actual instances are (e.g., due to missing information) or different scenarios should be compared, possible instances and their generic type will be exported (cf. the notion of generic asset types, and their possible realizations as presented in the previous subsection). This enables the generation of different scenarios that can be simulated separately. Next, we scan the National Vulnerability Database (NVD)⁷ for known vulnerabilities to the concrete instances in the GraphML, that are not already included in the original model, and enrich the GraphML with this information. Based on the enriched GraphML, we create the different scenarios that shall be simulated with securiCAD. Basically, we generate for each possible combination of configurations a single scenario that is then provided to securiCAD to be simulated. In other words, if there are two possible realizations of a meta-concept in the multi-level model (e.g., we do not certainly know if a computer is running on Windows or Linux), we would generate two scenarios, representing both configurations. Having reached the necessary level of detail to perform simulations in securiCAD, we are able to translate the graph representation into a format that is readable by securiCAD. Therefore, we rely on the securiCAD Model SDK⁸ that not only ensures that the model is readable by securiCAD, but also ensures that the created model follows a certain MAL DSL, i.e., icsLang in our case. Finally, we conduct the analysis of the different scenarios in a batched mode by facilitating the securiCAD Enterprise SDK⁹.

4 Demonstration

In this section, we perform a lightweight demonstration of our extended multi-level model and the tool chain in terms of a realistic attack scenario. We use the scenario to gauge the simulation capabilities inherent to our tooling chain.

Modeled Scenario: The Ukrainian attack scenario consisted of, cf. [6, 47]: (i) a primary attack, aimed at causing power disruptions. This primary attack

⁴ XModeler 2.0.5, <https://www.wi-inf.uni-duisburg-essen.de/LE4MM/>.

⁵ securiCAD Enterprise v1.11.2.

⁶ <https://github.com/simonhacks/MLM2securiCAD>.

⁷ <https://nvd.nist.gov/>.

⁸ <https://github.com/foreseeti/securicad-model-sdk>.

⁹ <https://github.com/foreseeti/securicad-enterprise-sdk>.

concerned the use of malware for obtaining credentials to access a Supervisory control and data acquisition (SCADA) system [56] via a VPN to subsequently use the obtained SCADA system access to remotely shut down circuit breakers in electricity substations, thus causing power outages; (ii) amplifying attacks, designed to maintain the power outages caused during the primary attack [47]. We focus on an amplifying attack, where the attackers targeted serial-to-ethernet gateways to prevent SCADA-substation communication [47] by replacing the legitimate firmware remotely. These serial-to-ethernet gateways mediate between the ethernet-based communication used by the SCADA system and legacy substation devices, which use serial communication. As such, by attacking the serial-to-ethernet gateways the attackers were successful in disrupting SCADA-substation communication and preventing remote access to substation equipment and the restoration of power. The attackers exploited different vulnerabilities stemming from substation design to deal with the hostile environment (extreme temperature fluctuations, voltage spikes, etc.): redundancy, minimum of moving parts, error-correcting memory, etc. [41]. Thus, substations often have an unsegmented network [47] and gateways that expose legacy devices previously (indirectly) protected simply by virtue of relying on serial communication protocols [39, 50].

Results: Based on the scenario, we modeled the corresponding system in the XModeler (by instantiating relevant concepts from higher classification levels) and accounted for uncertainty about the used routers between the networks. Namely, we solely know that there is a router deployed, and we assume that it is either a *Mikro Tik RB4011* or an *openurt*. Next, we generate two scenarios, cf. Fig. 5, with either router in place, and query the NIST NVD for known vulnerabilities for all assets within the model. This enables us to perform the attack simulations in securiCAD that result in different attack graphs, cf. Fig. 6, and probability distributions, cf. Fig. 7, for each attack step that provide information about how likely an attacker is to be successful within how many days.

Utility Assessment of the Proposed Artifact: Whereas our previous work emphasized a top-down approach to securely design systems, in this engineering cycle, we extend this reference model with specific asset types and their security relevant properties (cf. Req. 1). Firstly, system analysts or non-experts may benefit from the knowledge already accounted for within the model, its semantic richness and completeness. This knowledge facilitates the modeling process of enterprise system as well as supports security analyses. Secondly, the implemented tool chaining provides means to analyze the model for possible weaknesses (Req. 2). Here, we (re)use the created model of the enterprise system together with knowledge embedded in the multi-level model, to semi-automatically conduct attack simulations. As the scenario considered falls short when it comes to some specific details, there is some uncertainty about the concrete assets in place. Relying on the capability of MLM to express knowledge as soon as it becomes known (cf. Sect. 2), we are able to express this uncertainty. Particularly, we can generate different scenarios accounting for different realizations based on knowledge expressed on a higher level of abstraction which,

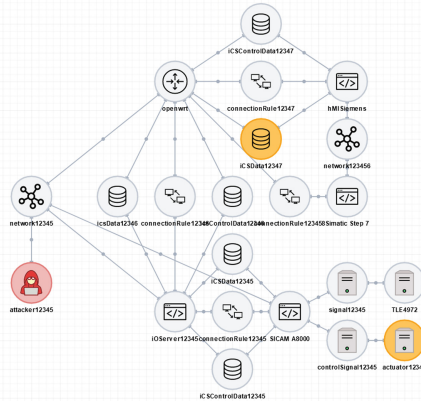


Fig. 5. Generated threat model for *open-wrt-scenario*.

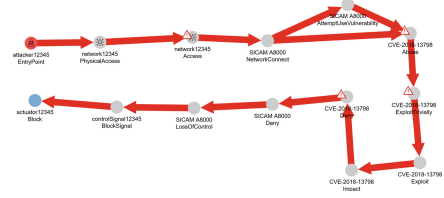


Fig. 6. Identified attack path to block actuator.

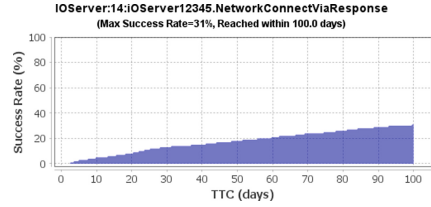


Fig. 7. Computed probability distribution to connect to IOserver.

then, can be simulated (Req. 3). Thus, we account for not known system's configuration, whereas using traditional approaches either we would not be able to conduct the simulation at all, or different design options would need to be manually created.

5 Related Work

Our work addresses the design of secure systems in its different phases, i.e., gathering security requirements, analyzing existing risks, and designing secure systems. For instance, when it comes to the latter, to identify security requirements one can rely on STS-ml, an actor- and goal-oriented requirements modeling language for socio-technical systems [38], or ModelSec [43] to relate security requirements to assets, threats, and contingency plans. However, the identification of security requirements, even given dedicated modeling languages, is not trivial to conduct and requires substantial knowledge on the domain under study and cyber-security concepts [34]. Here, reference models and standards focusing on cyber-security step in, such as the most widely-known reference model for smart grids cyber-security: the NIST reference model for cyber-security, NISTIR 7628 [35], which we have also used in our previous work [31]. NISTIR offers concepts, cyber-security requirements, and guidelines that are specific for the energy sector in terms of, e.g., considered actors, and IT infrastructure types. Yet another example for a source of threat intelligence is MITRE ATT&CK, which provides a taxonomy and instance knowledge for adversary tactics and

techniques based on real-world observations [48], or based on it a domain-specific language (DSL) [55], which we also accounted for in our multi-level model [31].

When it comes to *analyzing existing risks*, e.g., vulnerability analysis, different efforts have been conducted to reuse existing information instead of facilitating reference models within organizations and enrich it with security information. [23] used models describing the power infrastructure and performed attack simulations on them. The results of the simulations were then used to suggest an optimized infrastructure for the reduction of lost energy in case of an attack. [33] enhanced ArchiMate to xArchiMate to perform simulations, experiments, and analyze Enterprise Architectures (EA) by an extension to the ArchiMate meta-model. Similarly, [53] used EA repositories to predict effects of failing components on the entire architecture, without making actual changes to the used notation. In addition, process models can be used for similar purposes. For instance, [57] leveraged the extension mechanism provided in BPMN 2.0 to model threat-based security requirements. Contrary, [42] proposed a non-compliant BPMN meta-model extension including predefined set of high-level requirements, enabling business analysts to express their security needs.

Taking a more general point of view, model-driven security engineering offers a large number of DSLs (e.g., UMLsec [26, 27], SecureUML [4, 5], and SECTET [2, 18]), which partly also provide capabilities to perform simulations. Often, these approaches rely on the concept of attack trees [44, 45]. On this basis, many attack-graph-based approaches have been presented [32], resulting in a plethora of attack graph-based tools. These tools often have in common that they actively set out to collect information about some existing system and automatically simulate attack graphs based on this data.

For example, MulVal [21, 37] derives logical attack-graphs by facilitating vulnerability scans to determine the probability that an attacker is able to exploit them successfully. In contrast, in NAVIGATOR [7] vulnerabilities are considered to be directly exploitable by the attacker, given that they have access to the vulnerable system. The TVA tool [36] can be used to model networks in terms of security conditions and it uses a database of exploits to describe the transitions between these security conditions. A step further is taken by NetSecuritas [15] by creating additional security recommendations.

A subdomain of attack graph modeling and simulation elaborates on probabilities, in particular employing Bayesian networks. [13] translate TVA attack graphs into dynamic Bayesian networks, while relying on CVSS scores for the probabilities. Similarly, [52] rely on CVSS to model uncertainties in the attack structure, attacker’s actions, and alerts triggering. [40] use Bayesian attack graphs to estimate the security risk on network systems and produce a security mitigation plan using a genetic algorithm. Other works combine attack graphs and system models, e.g., CySeMoL [46], P2CySeMoL [20], pwnPr3d [25], and securiCAD [9]. The common idea for all of this work is that probabilistic attack graphs are automatically generated and calculated from a given system specification, devised in the respective frameworks’ separate language.

Our work contributes to the before presented related work in different ways. The main contribution of our approach relies in the simulation of different configuration scenarios, so dealing with uncertainty in different parts of the model (e.g., because a concrete version of a software is unknown). All previous approaches assume that it is known how the architecture exactly looks like. Our approach allows to overcome this assumption. In addition, due to embedding the domain-specific knowledge into the multi-level model, both when it comes to the design of information system, as well as security properties, possible threats, etc., such security analysis is made possible also for non-experts. Finally, the approach that we followed together with the application of multi-level modeling enables a first step towards the analysis of the business impact that is imposed by different vulnerabilities and reasoning about the influence of a single vulnerability.

6 Conclusions

In this paper, we focus on the application of a multi-level cyber-security reference model to support bottom-up analysis, especially vulnerability analysis of information systems. To this end, we extend our previous work, and capitalize on selected languages and tools offering a scarce cyber-security knowledge. To show the utility of our approach, we presented a scenario based on the cyber-attack on the Ukrainian electricity grid.

The presented extensions should be considered as a starting point only. On the one hand, we are benefiting from the multi-level reference model's prescriptive (e.g., security related knowledge, system properties, asset associations) and descriptive features (e.g., possibility to model the actual system we want to analyze), on the other hand, thanks to the integrated modeling and programming environment, we can run the required analysis using the model as a starting point. Nevertheless, a few limitations and challenges need to be mentioned: (1) For now, there is no tight integration between the tools used and the communication between the tools requires exporting/importing data between two platforms. (2) Due to the lack of semantic clarity of some icsLang concepts, some pragmatic decisions regarding classification of those concepts needed to be made. Whereas this lack of semantic clarity offers some flexibility to users and allows them to account for their perspective while using those concepts, our design of the multi-level model requires more specificity, as otherwise the reliability of the simulation in the multi-level modeling setting may be seriously affected. (3) Although FMML^x has proved to be a suitable instrument for reference modeling, a need for additional constructs, among others, support for contingent-level associations as well as a possibility to specialize associations became apparent. Therefore, some workarounds were necessary to model the required information.

When it comes to our future work, there are a couple avenues that we consider. Firstly, the analysis of the business-impact currently needs to be done manually by navigating through the model and reasoning on its content. To support semi-automated analysis of the business impact we are working on how

the results of the simulation could be propagated to the higher levels, as well as on what metrics could be used to capture this business impact. Secondly, the extent to which we capitalize on the semantic richness of the concepts modelled can be increased. To this end, we will investigate what properties or their values may provide us additional information regarding the security level of the system. For instance, there is a possibility to derive vulnerabilities and active defenses based on the stated values of security-relevant properties of some assets (on different classification levels). Next, as accounting only for IT assets vulnerabilities is not enough to secure an enterprise system, but human actors and their behavior need to be considered as well, cf. [16], further extensions to the model and the way the security analysis is conducted will be investigated. Finally, the suggested approach needs further validation and evaluation within real world settings.

References

1. Abraham, C., Chatterjee, D., Sims, R.R.: Muddling through cybersecurity: insights from the U.S. healthcare industry. *Bus. Horiz.* **62**(4), 539–548 (2019)
2. Alam, M., Breu, R., Hafner, M.: Model-driven security engineering for trust management in SECTET. *JSW* **2**(1), 47–59 (2007)
3. Atkinson, C., Kühne, T.: The essence of multilevel metamodeling. In: Gogolla, M., Kobryn, C. (eds.) *UML 2001. LNCS*, vol. 2185, pp. 19–33. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45441-1_3
4. Basin, D., Clavel, M., Egea, M.: A decade of model-driven security. In: 16th ACM Symposium on Access Control Models and Technologies, pp. 1–10. ACM (2011)
5. Basin, D., Doser, J., Lodderstedt, T.: Model driven security: from UML models to access control infrastructures. *ACM Trans. Softw. Eng. Methodol. (TOSEM)* **15**(1), 39–91 (2006)
6. Defense Use Case: Analysis of the cyber attack on the Ukrainian power grid. Electricity Information Sharing and Analysis Center (E-ISAC), vol. 388, pp. 1–29 (2016)
7. Chu, M., Ingols, K., Lippmann, R., Webster, S., Boyer, S.: Visualizing attack graphs, reachability, and trust relationships with navigator. In: Proceedings of the 7th International Symposium on Visualization for Cyber Security, pp. 22–33. ACM (2010)
8. Clark, T., Willans, J.: Software language engineering with XMF and XModeler. In: Mernik, M. (ed.) *Formal and Practical Aspects of Domain-Specific Languages: Recent Developments*, pp. 311–340. IGI Global (2013). <https://eprints.mdx.ac.uk/9560/>
9. Ekstedt, M., Johnson, P., Lagerström, R., Gorton, D., Nydrén, J., Shahzad, K.: securiCAD by foreseeti: a CAD tool for enterprise cyber security management. In: *Enterprise Distributed Object Computing Workshop*, pp. 152–155. IEEE (2015)
10. Frank, U.: Multilevel modeling - toward a new paradigm of conceptual modeling and information systems design. *BISE* **6**(6), 319–337 (2014)
11. Frank, U.: Designing models and systems to support it management: a case for multilevel modeling. In: *MULTI@MoDELS*, pp. 3–24. CEUR-ws.org (2016)
12. Frank, U.: The flexible multi-level modelling and execution language (FMMLx). version 2.0: analysis of requirements and technical terminology. Technical report 66, ICB-Research Report (2018)




13. Frigault, M., Wang, L., Singhal, A., Jajodia, S.: Measuring network security using dynamic Bayesian network. In: 4th ACM Workshop on Quality of Protection. pp. 23–30. ACM (2008)
14. Geismann, J., Gerking, C., Boddien, E.: Towards ensuring security by design in cyber-physical systems engineering processes. In: International Conference on Software and System Process, pp. 123–127 (2018)
15. Ghosh, N., Chokshi, I., Sarkar, M., Ghosh, S.K., Kaushik, A.K., Das, S.K.: Net-Securitas: an integrated attack graph-based security assessment tool for enterprise networks. In: Conference on Distributed Computing and Networking, p. 30. ACM (2015)
16. Hacks, S., Butun, I., Lagerström, R., Buhaiu, A., Georgiadou, A., Michalitsi Psarrou, A.: Integrating security behavior into attack simulations. In: International Conference on Availability, Reliability and Security, ARES 2021. ACM (2021)
17. Hacks, S., Katsikeas, S., Ling, E., Lagerström, R., Ekstedt, M.: powerLang: a probabilistic attack simulation language for the power domain. *En. Inf.* **3**(1), 1–17 (2020)
18. Hafner, M., Breu, R., Agreiter, B., Nowak, A.: SECTET: an extensible framework for the realization of secure inter-organizational workflows. *Internet Res.* **16**(5), 491–506 (2006)
19. Hevner, A.R., March, S.T., Park, J., et al.: Design Science in Information Systems Research. *MIS Q.* **28**(1), 75–105 (2004)
20. Holm, H., Shahzad, K., Buschle, M., Ekstedt, M.: P²CySeMoL: predictive, probabilistic cyber security modeling language. *IEEE Trans. Dependable Secure Comput.* **12**(6), 626–639 (2015)
21. Homer, J., et al.: Aggregating vulnerability metrics in enterprise networks using attack graphs. *J. Comput. Secur.* **21**(4), 561–597 (2013)
22. Jansen, A., Bosch, J.: Software architecture as a set of architectural design decisions. In: Conference on Software Architecture, pp. 109–120. IEEE (2005)
23. Jiang, Y., Jeusfeld, M., Atif, Y., Ding, J., Brax, C., Nero, E.: A language and repository for cyber security of smart grids. In: International Enterprise Distributed Object Computing Conference, pp. 164–170 (2018)
24. Johnson, P., Lagerström, R., Ekstedt, M.: A meta language for threat modeling and attack simulations. In: International Conference on Availability, Reliability and Security. ACM, New York (2018)
25. Johnson, P., Vernotte, A., Ekstedt, M., Lagerström, R.: pwnPr3d: an attack-graph-driven probabilistic threat-modeling approach. In: 2016 11th International Conference on Availability, Reliability and Security (ARES), pp. 278–283. IEEE (2016)
26. Jürjens, J.: UMLsec: extending UML for secure systems development. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 412–425. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45800-X_32
27. Jürjens, J.: Secure Systems Development with UML. Springer, Heidelberg (2005). <https://doi.org/10.1007/b137706>
28. Katsikeas, S., et al.: An attack simulation language for the IT domain. In: Eades III, H., Gadyatskaya, O. (eds.) GramSec 2020. LNCS, vol. 12419, pp. 67–86. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62230-5_4
29. de Kinderen, S., Kaczmarek-Heß, M.: Multi-level modeling as a language architecture for reference models: on the example of the smart grid domain. In: Conference on Business Informatics, pp. 174–183. IEEE (2019)
30. de Kinderen, S., Kaczmarek-Heß, M.: Making a case for multi-level reference modeling – a comparison of conventional and multi-level language architectures for

- reference modeling challenges. In: Ahlemann, F., Schütte, R., Stieglitz, S. (eds.) WI 2021. LNISO, vol. 48, pp. 342–358. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86800-0_24
31. de Kinderen, S., Kaczmarek-Heß, M., Hacks, S.: Towards cybersecurity by design: a multi-level reference model for requirements-driven smart grid cybersecurity. In: 30th European Conference on Information Systems, ECIS 2022, Timisoara (2022)
 32. Kordy, B., Piètre-Cambacédès, L., Schweitzer, P.: Dag-based attack and defense modeling: don't miss the forest for the attack trees. *Comp. Sci. Rev.* **13**, 1–38 (2014)
 33. Manzur, L., Ulloa, J.M., Sánchez, M., Villalobos, J.: xArchiMate: enterprise architecture simulation, experimentation and analysis. *Simulation* **91**(3), 276–301 (2015)
 34. Morikawa, I., Yamaoka, Y.: Threat tree templates to ease difficulties in threat modeling. In: Conference on Network-Based Information Systems, pp. 673–678 (2011)
 35. NIST Smart Grid Cybersecurity Panel: NISTIR 7628-guidelines for smart grid cyber security, vol. 1–3 (2010)
 36. Noel, S., Elder, M., Jajodia, S., Kalapa, P., O'Hare, S., Prole, K.: Advances in topological vulnerability analysis. In: Conference For Homeland Security, pp. 124–129 (2009)
 37. Ou, X., Govindavajhala, S., Appel, A.W.: Mulval: a logic-based network security analyzer. In: USENIX Security (2005)
 38. Paja, E., Dalpiaz, F., Giorgini, P.: Modelling and reasoning about security requirements in socio-technical systems. *Data Knowl. Eng.* **98**, 123–143 (2015)
 39. Pliatsios, D., Sarigiannidis, P., Lagkas, T., Sarigiannidis, A.G.: A survey on SCADA systems: secure protocols, incidents, threats and tactics. *IEEE Commun. Surv. Tutor.* **22**(3), 1942–1976 (2020)
 40. Poolsappasit, N., Dewri, R., Ray, I.: Dynamic security risk management using Bayesian attack graphs. *Trans. Dependable Secure Comput.* **9**(1), 61–74 (2012)
 41. Rayees, A.B.S.M.: Substation automation techniques and future trends. In: 2007 Innovations in Information Technologies (IIT), pp. 412–416 (2007)
 42. Rodriguez, A., Fernandez-Medina, E., Piattini, M.: A BPMN extension for the modeling of security requirements in business processes. *IEICE Trans. Inf. Syst.* **90-D**(4), 745–752 (2007)
 43. Sánchez, O., Molina, F., Garcia-Molina, J., Toval, A.: ModelSec: a generative architecture for model-driven security. *Univ. Comput. Sci.* **15**(15), 2957–2980 (2009)
 44. Schneier, B.: Attack trees. *Dr. Dobb's J.* **24**(12), 21–29 (1999)
 45. Schneier, S.: *Lies: Digital Security in a Networked World*, vol. 21, pp. 318–333. Wiley, New York (2000)
 46. Sommestad, T., Ekstedt, M., Holm, H.: The cyber security modeling language: a tool for assessing the vulnerability of enterprise system architectures. *IEEE Syst. J.* **7**(3), 363–373 (2013)
 47. Stellos, I., Kotzanikolaou, P., Psarakis, M., Alcaraz, C., Lopez, J.: A survey of IoT-enabled cyberattacks: assessing attack paths to critical infrastructures and services. *IEEE Commun. Surv. Tutor.* **20**(4), 3453–3495 (2018)
 48. Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G., Thomas, C.B.: MITRE ATT&CK: design and philosophy. Technical report, The MITRE Corporation (2018)
 49. Sun, C.C., Hahn, A., Liu, C.C.: Cyber security of a power grid: state-of-the-art. *Int. J. Electr. Power Energy Syst.* **99**, 45–56 (2018)

50. Taylor, J.M., Sharif, H.R.: Security challenges and methods for protecting critical infrastructure cyber-physical systems. In: MoWNeT, pp. 1–6 (2017)
51. Wieringa, R.J.: Design Science Methodology for Information Systems and Software Engineering. Springer, Heidelberg (2014). <https://doi.org/10.1007/978-3-662-43839-8>
52. Xie, P., Li, J.H., Ou, X., Liu, P., Levy, R.: Using Bayesian networks for cyber security analysis. In: Dependable Systems and Networks, pp. 211–220. IEEE (2010)
53. Xiong, W., Carlsson, P., Lagerström, R.: Re-using enterprise architecture repositories for agile threat modeling. In: EDOCW, pp. 118–127 (2019)
54. Xiong, W., Lagerström, R.: Threat modeling-a systematic literature review. *Comput. Secur.* **84**, 53–69 (2019)
55. Xiong, W., Legrand, E., Åberg, O., Lagerström, R.: Cyber security threat modeling based on the MITRE enterprise ATT&CK matrix. *SoSyM* **21**, 157–177 (2021)
56. Yadav, G., Paul, K.: Architecture and security of SCADA systems: a review. *Crit. Infrastruct. Protect.* **34**, 100433 (2021)
57. Zareen, S., Akram, A., Ahmad Khan, S.: Security requirements engineering framework with BPMN 2.0. 2 extension model for development of information systems. *Appl. Sci.* **10**(14), 4981 (2020)



Security Ontologies: A Systematic Literature Review

Malina Adach^(✉), Kaj Hänninen, and Kristina Lundqvist

School of Innovation, Design and Engineering, Mälardalen University,
Västerås, Sweden

{malina.adach,kaj.hanninen,kristina.lundqvist}@mdu.se

Abstract. Security ontologies have been developed to facilitate the organization and management of security knowledge. A comparison and evaluation of how these ontologies relate to one another is challenging due to their structure, size, complexity, and level of expressiveness. Differences between ontologies can be found on both the ontological and linguistic levels, resulting in errors and inconsistencies (i.e., different concept hierarchies, types of concepts, definitions) when comparing and aligning them. Moreover, many concepts related to security ontologies have not been thoroughly explored and do not fully meet security standards. By using standards, we can ensure that concepts and definitions are unified and coherent. In this study, we address these deficiencies by reviewing existing security ontologies to identify core concepts and relationships. The primary objective of the systematic literature review is to identify core concepts and relationships that are used to describe security issues. We further analyse and map these core concepts and relationships to five security standards (i.e., NIST SP 800-160, NIST SP 800-30 rev.1, NIST SP 800-27 rev.A, ISO/IEC 27001 and NISTIR 8053). As a contribution, this paper provides a set of core concepts and relationships that comply with the standards mentioned above and allow for a new security ontology to be developed.

Keywords: Security ontology · Concepts · Relationships · Security standards · Ontologies

1 Introduction

An ontology presents knowledge in a structured way and supports communication, organization, and knowledge reusability [1]. The main goals of an ontology are to describe reality with the concepts and relationships thereof, share vocabulary, and to provide a formal description of terms to avoid language ambiguity. Many security ontologies have been proposed over the past decade, but they only cover some aspects of the security domain. Several questions related to security ontologies still remain, for example:

Q1. Which core concepts and relationships can be used to adequately comprehend security issues?

Q2. Which of these core concepts and relationships should be included in a security ontology?

Q3. Which core concepts are compliant with security standards?

In this study, we conduct a systematic literature review of existing security ontologies. Since different ontologies may propose different definitions to explain concepts and relationships, a systematic literature review can indicate and facilitate the extraction of common core concepts and relationships that should be included in a security ontology [2]. The need for security is fundamental and includes many concepts and relationships, so engineering a security ontology is a considerable, but worthwhile challenge [3]. The concepts and relationships delineated in a security ontology should be detailed, and the concepts should be mapped to existing security standards to reduce ambiguities in the development thereof (e.g., differences in definitions used, incomplete ontologies). This paper presents a systematic literature review that offers an overview of research on existing security ontologies to identify the core concepts and relationships and map them to the following five security standards: NIST SP 800-160 [4], NIST SP 800-30 rev.1 [5], NIST SP 800-27 rev.A [6], ISO/IEC 27001 [7], and NISTIR 8053 [8]. These standards were selected because they facilitate the exchange of knowledge by ensuring a common understanding of concepts and definitions.

Developing a System-of-System with compliance standards improves security, and this was the reason for mapping identified concepts to common security standards. The mapping of security standards can help ontologies to be optimized by identifying concepts compatible with security standards and removing redundant security measures to meet those standards. Existing security ontologies can be used to simplify the mapping of more than two security standards. Since security ontologies cover a wide range of areas, they play a significant role in mapping. Consequently, their concepts must be detailed and described to comply with security standards. The contribution of this study differs from previous efforts in the following ways:

- Core concepts and relationships that capture security issues were identified
- Already-developed security ontologies were reused without being redefined
- Core concepts were mapped to existing security standards
- Security knowledge was considered to reuse and expand previously collected knowledge.

The reason we studied these ontologies to find common themes is that we need to identify security concepts and relationships that can be mapped to security standards. The main contribution of this paper is a proposal of core concepts and relationships that complies with the above-mentioned standards and can be used to develop a novel security ontology. The remainder of this paper is structured as follows: Sect. 2 describes background on security ontologies and studies that are related to our work. The process of the systematic literature review and analysis of the results are detailed in Sect. 3. Section 4 presents conclusions and further research directions.

2 Background and Related Work

This section introduces the necessary background on existing security ontologies.

2.1 Security Ontologies

Security-related issues are critical in all contexts related to the exchange of personal data and confidential information [9]. For a System-of-Systems (SoS), as an example, there are features of significant concern related to multiple iterations between humans, autonomous vehicles, and technology, and the heterogeneity of different autonomous vehicles that relate to various forms of technology. In these situations, it is crucial to verify the security and privacy of services and applications to ensure that the SoS function properly. There are potentially several and wide-ranging problems, especially when security concepts are misunderstood or misinterpreted [10]. For this reason, an ontology can be broadly used to organize a specific area of interest.

Several ontologies have been proposed in literature to resolve security-related issues; each ontology varies according to the complexity of the specific problem, the amount of detail needed and the area that the ontology is intended to cover. For instance, one of the earliest works related to the security domain described the concepts of an information system and proposed a language, Telos, for the information system knowledge. The authors of this study emphasized that Telos can also be used for the purpose of security specification [11]. Landwehr et al. provided a taxonomy of different security flaws in computer programs [12]. A broad and abstract taxonomy describing the security concepts that includes the idea of faults, fault tolerance techniques, fault modes, and verification approaches has also been proposed [13]; this taxonomy is not exhaustive and is somewhat restricted in terms of its ability to classify actual attacks due to limited relationships among the different classes [14]. Many studies have highlighted the need for a security ontology, rather than a taxonomy of the security domain [15]. Blanco et al. listed several security ontologies in their work [16]; some of these only focused on one area of the information-security domain, while others provided an overview of information security, but nothing that is specific enough for this purpose.

Among the general ontologies that are relevant to this discussion is the proposal for the Web Ontology Language based ontology for information security [17], which provides an expandable ontology for the information-security domain that consists of domain-specific terminology and general concepts (e.g., top-level concepts, such as assets, threats, vulnerabilities, and countermeasures) and domain-specific technical vocabulary. Similarly, Blanco et al. suggested an ontology that models a larger portion of the information-security domain and includes non-core concepts like organizational infrastructure [16]; this ontology includes high-level concepts, such as assets, control, organization, threats, and vulnerabilities. While both ontologies are interesting, neither of them is exhaustive or sufficiently comprehensive; the former provides a clear and simple ontology

that explains threat concepts, and the latter proposes a more complex ontology to explain asset-related concepts. This lack of specificity is covered by other security-domain-specific ontologies, e.g., [18, 19].

It is advisable to consider reusing existing ontologies to develop a more complete ontology that is capable of covering multiple security-related issues [20].

2.2 Existing Systematic Literature Review of Security Ontologies

Ontologies are used in the security domain to obtain, manage, and share information and security knowledge and can be divided into two categories: general and security-specific [21]. The goal of security ontologies is to develop common, unambiguous semantic models of security domain concepts that reduce language ambiguity, while at the same time providing a means for easy expansion and usability of relevant knowledge in research [22].

Souag et al. [21] conducted a systematic literature review to identify existing research on ontologies and the requirements and security issues thereof. They proposed eight categories according to which security ontologies could be classified: theoretical basis, security taxonomies, general, specific, risk-based, web-oriented, requirements-related, and modeling. The authors only found a few studies related to security ontologies that offered different methods to cover security issues; each ontology was analyzed for the way it covered a specific issue and to determine whether it could be used to define security requirements. This analysis revealed a gap between ontology and security-engineering domains. Nguyen [23] presented a basic review of ontology as it relates to security information systems. The aim of this research was to investigate the literature and identify areas of interest for further research. The author concluded that at that time, there were no ontologies for use in the modeling and security of computer networks. Blanco et al. [16] performed a systematic literature review to identify, analyze, and extract the main security ontologies related to the information security domain. They only considered titles, keywords, and abstracts when analyzing these papers, and they concluded that the literature could be classified into three groups: seventeen were general and specific security ontologies, nine were semantic web-oriented ontologies, and four were theoretical papers. The authors discovered that existing security ontologies do not exhaustively define concepts, do not use appropriate descriptive language for descriptions and cannot be extended or reused. Three years after publishing this review, Blanco et al. [24], conducted an extended systematic literature review that included their earlier analysis and a comparison of the security ontologies detailed therein. The aim of this research was to identify and classify the purpose of each study; titles, keywords, and abstracts were analyzed and delineate relationships between ontological concepts used in security domains, but security standards were not considered in this analysis. The investigation resulted in eight general and 20 security-specific ontologies, and three theoretical papers. The authors concluded that these ontologies contributed to the security domain, but only provided a partial solution, rather than an integrated security ontology. They also deter-

mined that successfully implementing an integrated ontology was a complex task that required more in-depth study.

While these studies classified, analyzed, and reviewed several existing security ontologies, they did not cover the entire spectrum of security knowledge; we will therefore include as many security-knowledge resources as we can in this study in order to identify the core concepts and relationships thereof. Moreover, because these studies focused on information system security, rather than general security, our goal is not to compare different security ontologies, but rather to integrate existing ontologies to create an appropriate new security ontology. The aforementioned reviews were related to the security aspects of application-specific domains, and they did not include the security standards we use for ontology creation. In contrast, our approach, considers various security ontologies and is therefore general enough to be applicable to any IT system. Even though the cited research did not examine any ontological concepts mapped to security standards, we were able to use these studies to identify the core concepts and relationships for various security issues and map them to five security standards.

3 The Systematic Literature Review

In this section, the procedure for conducting the systematic literature review is explained. The systematic literature review was based on the original guidelines proposed by Kitchenham [25] and was divided into three stages:

- 1. Planning:** Questions that need to be answered by the systematic literature review were formed, and a review protocol was defined that sets out the main procedures to be followed during the review.
- 2. Conducting:** Secondary sources and studies were selected, inclusion and exclusion criteria were defined, and all the relevant papers were extracted. All duplicate search results were removed, then the results were screened through inclusion/exclusion criteria.
- 3. Reporting:** Data synthesis was performed (i.e., the studies were classified), and the questions formed in the first stage were answered.

The scope of this review has been limited to identifying the core concepts and relationships that:

- (i) can be utilized to adequately comprehend security issues,
- (ii) should be included in a security ontology, and
- (iii) are compliant with security standards.

This study focuses on identifying and gathering concepts and relationships that can be used to develop a novel security ontology.

3.1 Planning the Systematic Literature Review

Formulating the Systematic Literature Review Questions

The formulation of the questions serves to introduce the systematic literature

review methodology [25]. Therefore, we formed the following three questions to identify the core security concepts and relationships that were presented in the literature:

- Q1. Which core concepts and relationships can be used to adequately comprehend security issues?*
- Q2. Which of these core concepts and relationships should be included in a security ontology?*
- Q3. Which of these core concepts are compliant with security standards?*

Defining the Review Protocol

According to Kitchenham [25], the review protocol should define the methods for how the following activities are to be conducted in a systematic literature review, such as the creation of a research strategy, the selection of primary studies as well as the inclusion and exclusion criteria, the quality of the assessment criteria, data extraction, and data synthesis. In Sect. 3.2, we will describe how we defined and performed each activity of this protocol.

3.2 Conducting the Systematic Literature Review

The research strategy and the selection of primary studies are presented at this stage. The research strategy’ goal is to find as many studies as possible that are related to the questions posed in Sect. 3.1. The research process includes the selection of the literature sources, the definition of the search string, the specification of inclusion and exclusion criteria, and the conducting of the research.

Selection of Literature Sources

The search for peer reviewed literature was conducted in the three major online databases, IEEE Xplore [26], Scopus [27] (includes: IEEE, ACM, and Elsevier, Wiley, and Springer), and Web of Science (includes: IEEE, ACM, and Elsevier) [28]. Selected databases provide access to preview and download the abstract and full text papers. The overlapping between the IEEE databases and ACM publications is covered by Scopus and Web of Science. This allows us to reduce the risk of omitting some papers of interest. Sources from the security domain were collected, and publications related to security ontologies were selected. The selection criteria for identifying security-related concepts and the relationships among them were based on the existing definitions and descriptions of these concepts and relationships.

Search String

Following [25], we derived the primary search string from the questions. Specifically, we used “Boolean AND” to link the primary search string and “Boolean OR” to include alternative synonyms of such a search string. We used a wildcard via an asterisk (*) in the search string for multiple character searching (e.g., ontolog*, securit*, cyber*). We searched for ontologies developed both for security and cybersecurity. We divided the search string into two parts (1 and 2, differences are highlighted in bold text) in the IEEE database because the

number of wildcards is limited to 7 per search. Papers published as conference articles, journal papers, early access or book chapters in the computer science domain between January 1988 and April 2022 were selected. We used the following search strings to define the titles and abstracts in each database, and the number of papers found in the search.

IEEE 1: ((*“Document Title”: Ontolog**) AND (*“Document Title”: Securit** OR *“Document Title”: threa** OR *“Document Title”: vulnerability* OR *“Document Title”: privacy* OR *“Document Title”: attack* OR *“Document Title”: confidentiality* OR *“Document Title”: integrity* OR *“Document Title”: asset* OR *“Document Title”: countermeasure* OR *“Document Title”: control* OR *“Document Title”: consequence* OR *“Document Title”: cyber**)) OR ((*“Abstract”: Ontolog**) AND (*“Abstract”: Securit** OR *“Abstract”: threa** OR *“Abstract”: vulnerability* OR *“Abstract”: privacy* OR *“Abstract”: attack* OR *“Abstract”: confidentiality*)) - returned **1127** results.

IEEE 2: ((*“Document Title”: Ontolog**) AND (*“Document Title”: Securit** OR *“Document Title”: threa** OR *“Document Title”: vulnerability* OR *“Document Title”: privacy* OR *“Document Title”: attack* OR *“Document Title”: confidentiality* OR *“Document Title”: integrity* OR *“Document Title”: asset* OR *“Document Title”: countermeasure* OR *“Document Title”: control* OR *“Document Title”: consequence* OR *“Document Title”: cyber**)) OR ((*“Abstract”: Ontolog**) AND (*“Abstract”: integrity* OR *“Abstract”: asset* OR *“Abstract”: countermeasure* OR *“Abstract”: control* OR *“Abstract”: consequence* OR *“Abstract”: cyber**)) returned **1809** results.

Scopus: (ABS(((ontolog*) AND ((securit*) OR (threa*) OR vulnerability OR privacy OR attack OR confidentiality OR integrity OR asset OR countermeasure OR control OR consequence OR (cyber*)))) AND TITLE (((ontolog*) AND ((securit*) OR (threa*) OR vulnerability OR privacy OR attack OR confidentiality OR integrity OR asset OR countermeasure OR control OR consequence OR (cyber*)))) AND PUBYEAR > 1988 AND PUBYEAR < 2022 AND (LIMIT-TO (SUBJAREA, “COMP”) AND (LIMIT-TO (DOCTYPE, “cp”) OR LIMIT-TO (DOCTYPE, “ar”) OR LIMIT-TO (DOCTYPE, “ch”)) AND (LIMIT-TO (LANGUAGE, “English”)) returned **698** results.

Web of Science: (SU = Computer Science AND (((TI = Ontolog*) AND (TI = (Securit*) OR TI = (asset) OR TI = (threa*) OR TI = (privacy) OR TI = (attack) OR TI = (confidentiality) OR TI = (control) OR TI = (integrity) OR TI = (countermeasure) OR TI = (vulnerability) OR TI = (cyber*) OR TI = (consequence))) OR ((AB = (Ontolog*)) AND (AB = (Securit*) OR AB = (threa*) OR AB = (vulnerability) OR AB = (privacy) OR AB = (attack) OR AB = (asset) OR AB = (integrity) OR AB = (confidentiality) OR AB = (countermeasure) OR AB = (control) OR AB = (consequence) OR AB = (cyber*)))))) returned **3921** results.

Research Process

The research process was carried out in two steps.

First, we used the aforementioned electronic databases and only selected papers with titles and abstracts that were deemed relevant according to the search string. In the second step, we applied the inclusion and exclusion criteria to selected papers. Our review was conducted manually, and each author participated in the entire screening process. There are many security ontologies and ontological approaches represented by UML (Unified Modelling Language) class models or OWL (The Web Ontology Language), which can only be manually interpreted. Our preliminary search resulted in a total of **7,555** papers, and selection criteria were applied to these papers to obtain the final group of relevant papers. The results of searches are shown in [29]. Based on our review, we have not identified any paper strengths or weaknesses, merely focusing on our inclusion and exclusion criteria.

Primary Selection - Inclusion and Exclusion Criteria

The research with the selected databases returned 7,555 relevant papers from which we removed **2,442** duplicate search results. We focused on analyzing titles and abstracts of the returned papers to discover how the concepts relate to security. Then, we applied the primary selection criteria (shown in Table 1) to the remaining **5,113** papers.

Table 1. Inclusion and exclusion criteria

Inclusion criteria	Exclusion criteria
Papers are published in the English language	Papers are published in languages other than English
Papers present the development or extension of ontology/(ies) or an ontology-based approach related to security and has already been used at least once	Papers present an ontology or an ontology-based approach that is not related to security
Papers present comparison/reviews/surveys of ontology/(ies) related to security	Gray literature papers, short papers or posters
Papers are published from January 1988 to April 2022	Work-in-progress papers
Papers are published and available in scientific databases or printed versions	Papers are not available
Papers are related to any of the questions from Sect. 3.1	Papers are not related to any of the questions from Sect. 3.1
Complete versions of papers	Multiple versions of the same papers
Papers have already approval by the scientific community	Papers shorter than 3 pages

This process has been performed manually with the possibility to evaluate each paper with one of three options:

1. The paper is accepted because it presents the development, extension, or comparison/ reviews/ surveys of ontology/ies or ontology-based approaches that cover different aspects related to security.
2. The paper presents an ontology or an ontology-based approach that is not related to security.
3. The paper is rejected because it does not meet criterion in 1 or 2.

After manually applying the above-mentioned three options and the inclusion and exclusion criteria presented in Table 1 to the paper’s titles and abstracts, **4,914** irrelevant papers were excluded, and **199** relevant papers were analyzed in the next step.

Quality Assessment

The quality assessment criteria were applied to the **199** papers obtained from the aforementioned steps. Furthermore, these criteria were applied to **22** additional papers that were identified through the snowballing step. To identify the relevant papers that could be used to answer questions from Sect. 3.1, we formed the three following quality assessment (QA) questions:

QA1. Are the presented concepts and relationships clearly defined and described?

QA2. Do the papers present an appropriate way for the concepts and relationships to deal with security issues?

QA3. Have the concepts and relationships been justified by sufficient analysis or examples?

The above quality assessment criteria were applied to the full texts of **221** papers. To assess the paper’s completeness and relevance, each QA had only two possible answers, “Yes” or “No”. If the answer is “No” to any one of the quality assessments questions, the paper is excluded.

3.3 Reporting the Systematic Literature Review

In the final stage, the summary of the results is included. This consists of three steps:

1. Data synthesis
2. Results and analysis
3. Answers to the questions from Sect. 3.1.

Data Synthesis – Classification of Studies

The data related to QA1 was extracted directly from the list of selected papers presented in Sect. 3.2. To answer QA2, the contents of the 8 selected papers were further analyzed to identify the core concepts and relationships. The collected

core concepts and relationships are presented in Table 3. In addition, we identified the core concepts and relationships that should be included in a security ontology, and described them in Sect. 3.3. To answer QA3, the core concepts shown in Table 4 were mapped to the definitions proposed in the security standards.

Results and Analysis

The results of the systematic literature review are summarized below and presented in Fig. 1. The search in three databases returned total of **7,555** papers from which **2,442** duplicate search results were removed. For the review process, we have developed inclusion and exclusion criteria that we can refer to when either including or excluding a paper. The inclusion and exclusion criteria were designed to select papers that address our main research questions. Therefore, these criteria determine the scope of our review.

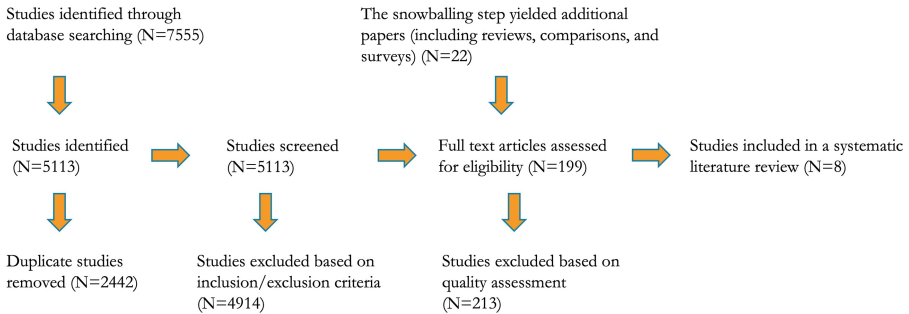


Fig. 1. Paper screening process

During title and abstract review of **5,113** papers, **4,914** articles were excluded based on inclusion/exclusion criteria. A total of **199** papers were assessed for eligibility in the systematic literature review. After applying the inclusion and exclusion criteria, the final papers were selected for quality assessment. The snowballing step yielded **twenty-two** additional papers (including reviews, comparisons, and surveys). This approach provided us with **221** papers, which were included in the quality assessment step. Based on the quality assessment from Sect. 3.2, **213** papers were excluded, and only **8** papers that met the criteria were included in the systematic literature review.

As a result, **8** eligible papers were selected as relevant for addressing our question from Sect. 3.1. The results of the QA of the papers are presented in Table 2. Below is a short summary of the 8 papers identified as relevant:

1. Schumacher [30] proposed a security ontology with nine concepts and 12 relationships for maintaining the security pattern repositories using a theoretical search engine to locate security patterns. Consequently, the author has focused on identifying a small number of core security concepts and limited the scope to first level of abstraction.

2. Dritsas et al. [31] proposed a specialized security ontology with seven core concepts and nine relationships for the e-poll domain and presented how the ontology can help developers working in software projects to deal with a wide range of security issues.

3. Fenz and Ekelhart [32] proposed a security ontology with 11 concepts and 15 relationships that provides a unified and formal knowledge for the information security domain. Their ontology was integrated with ISO/IEC 27001 [7] standard ontology and applied to quantitative risk assessment.

Table 2. The results of the quality assessment of the papers

Author (year)	Title	Number of concepts	Number of relationships
Schumacher (2003) [30]	<i>Towards a security core ontology</i>	9	12
Dritsas et al.(2005) [31]	<i>Employing ontologies for the development of security critical applications</i>	7	9
Herzog et al. (2007) [17]	<i>An ontology of information security</i>	6	7
Fenz and Ekelhart (2009) [32]	<i>Formalizing information security knowledge</i>	11	15
Wang and Guo (2010) [33]	<i>OVM: an ontology for vulnerability management</i>	6	10
Pereira et al. (2012) [34]	<i>An ontology approach in designing security information systems to support organizational security risk</i>	8	16
Ramanauskaitė et al. (2013) [36]	<i>Security ontology for adaptive mapping of security standards</i>	5	7
Agrawal (2016) [37]	<i>Towards the ontology of ISO/IEC 27005:2011 risk management standard</i>	11	16

4. Herzog et al. [17] proposed a Web Ontology Language based ontology of information security overview to model security concepts, such as assets, counter-measures, threats, vulnerabilities, and their relationships. This ontology includes six core concepts and seven relationships, and can be used for reasoning about the relationships between concepts and can help determine threats that might be compromising the assets.

5. Wang and Guo [33] proposed the ontology for vulnerability management (OVM) with six concepts and ten relationships, which captures the core concepts of information security and focuses on software vulnerabilities. The authors utilized the NVD (National Vulnerability Database) to populate their ontology with descriptions of some common vulnerabilities.

6. Pereira et al. [34] proposed a security ontology with eight concepts and 16 relationships to support organizations in dealing with the many security

information issues and implementing appropriate management to facilitate their security decision-making needs. This ontology aims to unify the concepts and terminology of information security according to the ISO/IEC_JTC1 [35].

7. Ramanauskaite et al. [36] proposed a security ontology with five concepts and seven relationships that maps various security standards (e.g., ISO 27001 [7], ISSA 5173 [38], NISTIR 7621 [39], and PCI DSS [40]). These standards are mapped to optimize the use of multiple security standards in organizations and minimize the complexity of mapping.

8. Agrawal [37] proposed an ontology that defines the concepts of ISO 27005 [41], including risk management standards and relationships. This ontology includes 11 concepts and 16 relationships, and enables a better understanding and identification of the core concepts of ISO 27005 [41].

Table 3 presents core concepts and relationships that have been gathered from the eight above mentioned papers.

Table 3. Security concepts and relationships used to capture security issues in the identified papers

Author	Concepts	Relationships
Schumacher [30]	Asset, attack, attacker, countermeasure, risk, security objective, stakeholder, threat, vulnerability	<i>Address, carry out, cause harm to, exploits, express, has, implements, increases, place value on, protect against, realizes, reduces</i>
Dritsas et al. [31]	Asset, attacker, deliberate attack, countermeasure, objective, stakeholder, threat	<i>Address, damages, defines, implements, protects, protects, realizes, threatens, uses</i>
Herzog et al. [17]	Asset, countermeasure, defense strategy, security goal, threat, vulnerability	<i>EnabledBy, has, protects, protects, protects, threatens, threatens</i>
Fenz and Ekelhart [32]	Asset, control, control type, organization, security attribute, severity scale, standard control, threat, threat origin, threat source, vulnerability	<i>Affects, correspondsTo, gives rise to, has, has, has, isExploitedBy, isImplementedBy, isMitigatedBy, isOwnedBy, of, on, requires, requires, threatens</i>
Wang and Guo [33]	Attack, attacker, consequence, countermeasure, IT_Product, vulnerability	<i>Attack, attackConsequence, isExploitedBy, causes, conducts, has, has, hasRelated, mitigates, protects</i>
Pereira et al. [34]	Asset, attack, CIA, control, event, incident, threat, vulnerability	<i>areEffectedBy, detects, detects, effects, explores, has, has, isMadefrom, lostOf, materialized, protects, protects, protects, reduces, responds, towards</i>
Ramanauskaite et al. [36]	Asset, countermeasure, organization, threat, vulnerability	<i>Eliminates, existsIn, existsIn, exploits, has, has, mitigates</i>
Agrawal [37]	Asset, CIA, consequence, control, event, likelihood, objective, organization, risk, threat, and vulnerability	<i>Affects, affects, causes, contains, exploits, harms, has, has, has, has, isRealizedBy, leadsTo, mitigates, modifies, modifies, owns</i>

The above 8 papers were identified as relevant in answering questions from Sect. 3.3. Answers were presented in the next step.

Answers to the Review Questions

This section includes the answers and the findings of this systematic literature review. First, we answered the questions in Sect. 3.1. Then, we presented the findings of this systematic literature review.

Q1. Which core concepts and relationships can be used to adequately comprehend security issues?

We thoroughly analyzed each of the 8 selected papers to identify any concepts and relationships that could be used to capture any security issues. The results include the concepts and relationships described in Sect. 3.3 that have been identified in each selected paper. As a result, a total of **63** concepts and **92** relationships were identified, among which **27** unique concepts and **51** relationships were distinguished.

Q2. Which of these core concepts and relationships should be included in a security ontology?

Among the 27 identified concepts and 51 relationships, we have selected **12** core concepts and **35** relationships that should be included in a security ontology. Each of the selected concepts and relationships was selected based on the following three criteria:

- its relevance for capturing security issues;
- limitation to a high-level of abstraction (e.g., system-level concepts), and
- the frequency of its appearance in the selected papers.

The following **concepts and relationships** (frequency of appearance) were selected:

Core Concepts: Asset (7), Attack (3), Attacker (3), Consequence (2), Control (3), Countermeasure (5), Event (2), Incident (1), Organization (3), Security Goal (1), Threat (7), Vulnerability (7).

Core Relationships: *Affects* (3), *attackConsequence* (1), *causes* (2), *conducts* (1), *detects* (2), *eliminates* (1), *exists in* (2), *exploits* (3), *gives raise to* (1), *has* (16), *is exploited by* (2), *is implemented by* (1), *is made from* (1), *isMitigatedBy*, *is owned by* (1), *materialized* (1), *mitigates* (4), *modifies* (2), *owns* (1), *protects* (8), *realizes* (2), *reduces* (2), *requires* (2), *responds* (1), *threatens* (4), *towards* (1).

Q3. What are the core concepts that are compliant with security standards?

We answered this question by comparing the definitions of the concepts collected from the selected papers with the definitions proposed in the security standards. In this step, from the 8 relevant papers core concepts were extracted and duplicates were removed. Only **12** core concepts extracted could be mapped to the definitions described in the security standards. As the definitions described in the standards are more detailed, a mapping of definitions from the standards to the core concepts collected from the analyzed papers was needed. The concept mapping with security standards is presented in Table 4.

The concepts of Asset, Consequence, Control, Countermeasure, Event, Incident, Organization, and Vulnerability are mapped to the standards ISO/IEC 27001 [7] and NIST SP 800-160 [4]. The concept of Attack is mapped to the standards ISO/IEC 27001 [7] and NIST SP 800-30 rev.1 [5].

Table 4. Definitions of the core concepts mapped to security standards

Definitions of core concepts	Security standard
An asset is any resource (i.e., a tangible (furniture) or intangible (data)) that has importance and value to the owner, which may be the target of a security incident. It can exhibit some weaknesses that make assets susceptible to exploitation	NIST SP 800-160
An attack is an unauthorized access to or use of an asset, or an attempt to expose, destroy, disable, alter, gain, or steal an asset that an attacker can take by exploiting any vulnerability and producing security events	NIST SP 800-30 ISO/IEC 27001
An attacker is anyone or anything that attempts to expose, destroy, disable, alter, gain, or steal an asset by exploiting any vulnerability and producing some security events	NISTIR 8053
A consequence is the possible outcome of an attack or an event (e.g., data modification, denial of services), affecting the properties (CIA) of an asset or a security incident caused by an attacker	NIST SP 800-160 ISO/IEC 27001
A control is a mean of managing risk (e.g., policies), which can be of an administrative, technical, managerial, or legal nature. An attribute assigned to an asset reflects its relative importance or necessity in achieving or contributing to stated goals	NIST SP 800-160 ISO/IEC 27001
A countermeasure is a prevention mechanism that detects an incident/event, reduces or avoids a threat/an incident's effects, and/or protects an asset and its properties. It can be an action/approach that mitigates or prevents the risk and impacts of an attack or a measure that modifies risk and mitigates defined vulnerabilities by implementing physical or organizational measures	NIST SP 800-160
An event is an occurrence or change of a particular set of circumstances	NIST SP 800-160 ISO/IEC 27001
An incident is an anomalous or unexpected event, set of events, a condition, or situation at any time during the life-cycle of a project, product, service, or system	NIST SP 800-160 ISO/IEC 27001
An organization is a group of people and facilities with responsibilities, authorities, and relationships	NIST SP 800-160 ISO/IEC 27001
A security goal includes confidentiality, availability, integrity, accountability, assurance, anonymity, authentication, authorization, correctness, identification, non-repudiation, policy compliance, privacy, secrecy, and trust	NIST SP 800-27
A threat is a potential cause of an unwanted incident which can harm a system/organization/asset. It includes the types of dangers against a given set of security properties (CIA) and can be classified as passive, active, natural, accidental, and intentional	NISTIR 8053 ISO/IEC 27001
A vulnerability is any weakness of an asset or the system that can be exploited by a threat (e.g., security flaws). It can be influenced directly (intentionally malicious) or indirectly (an unintentional mistake) by human behavior	NIST SP 800-160 ISO/IEC 27001

The concepts of Attacker and Threat are mapped to the standard NISTIR 8053 [8]. A concept of Security Goal is mapped to the standard NIST SP 800-27 Rev.A [6].

Based on the systematic literature review results, we identified a set of core concepts and relationships among them that were used to capture security issues and should be included in a security ontology. We mapped the collected security concepts to the definitions proposed by the security standards. The obtained 12 core concepts - **asset, attack, attacker, consequence, control, countermeasure, event, incident, organization, security goal, threat, and vulnerability** - and their relationships can be used to develop a new security ontology.

4 Conclusions and Future Work

We conducted a systematic literature review of the existing security literature to identify the core concepts for capturing security issues and the relationships thereof. Overall, we included **221** papers in this review, and we examined all of these with three quality assessment criteria questions in mind. The selection process has been based on titles, abstracts, and full-text reading. As a result, **8** eligible papers were selected as relevant to the questions from Sect. 3.1 and used for further analysis, and we presented the selected data. Effective presentation of the set of selected data from the relevant papers was made using tables. Based on the results of our review, we conclude that the existing ontologies are not complete or consistent, lack the core concepts, and do not fully comply with existing security standards. We then identified a set of core concepts and relationships that capture security issues. The definitions of these **12** core concepts were mapped to security standards. The aim of this paper was to review and analyze selected security ontologies and to extract core concepts and relationships that capture security issues. The reason we studied these ontologies to find a common theme was that we needed to identify security concepts and relationships that could be mapped to security standards and compared with safety concepts and relationships. To the best of our knowledge this is the first study that maps the core concepts and relationships with common security standards. The main contribution of this paper proposes the core concepts and relationships that comply with the above-mentioned standards and allow the development of a new security ontology that can be evaluated and compared to other ontologies.

Acknowledgment. This work is supported by the projects: Serendipity - Secure and dependable platforms for autonomy, grant nr: RIT17-0009, funded by the Swedish Foundation for Strategic Research (SSF) and by the DPAC - Dependable Platform for Autonomous Systems and Control, grant nr: 20150022, funded by the Knowledge foundation (KKS).

References

1. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? *Int. J. Hum.-Comput. Stud.* **43**(4–5), 907–928 (1995)
2. Kang, W., Liang, Y.: A security ontology with MDA for software development. In: *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 67–74. IEEE, Beijing (2013)
3. Tsoumas, B., Gritzalis, D.: Towards an ontology-based security management. In: *20th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 985–992. IEEE, Vienna (2006)
4. Ross, R.S., McEvelley, M., Oren, J.C.: NIST SP 800-160, *Systems Security Engineering Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems*. NIST, US Department of Commerce, Gaithersburg, MD, USA, Technical report, NIST SP (2016)
5. Ross, R.S.: NIST SP 800-30 REV.1: guide for conducting risk assessments. <https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final>. Accessed 14 Aug 2022
6. Stoneburner, G., Hayden, C., Feringa, A.: NIST SP 800–27 Rev. A. *Engineering principles for information technology security (a baseline for achieving security)*, NIST (2017)
7. ISO/IEC 27001:2013 - Information security management system - requirements, ISO, Technical report (2013)
8. Garfinkel, S.L.: NISTIR 8053: de-identification of personal information, NIST (2015)
9. Maxwell, T.A.: Information policy, data mining, and national security: false positives and unidentified negatives. In: *38th Annual Hawaii International Conference on System Sciences*, pp. 134c–134c (2005). <https://doi.org/10.1109/HICSS.2005.317>
10. Jurisica, I., Mylopoulos, J., Yu, E.: Ontologies for knowledge management: an information systems perspective. *Knowl. Inf. Syst.* **6**(4), 380–401 (2004). <https://doi.org/10.1007/s10115-003-0135-4>
11. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: representing knowledge about information systems. *ACM Trans. Inf. Syst.* **8**(4), 325–362 (1990)
12. Landwehr, C.E., Bull, A.R., McDermott, J.P., Choi, W.S.: A taxonomy of computer program security flaws. *ACM Comput. Surv.* **26**(3), 211–254 (1994)
13. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secure Comput.* **1**(1), 11–33 (2004)
14. Howard, J., D., Longstaff, T.: *A common language for computer security incidents*. Sandia National Laboratories, pp. 1–25 (1998)
15. Donner, M.: Toward a security ontology. *IEEE Secur. Priv.* **1**(3), 6–7 (2003)
16. Blanco, C., Lasheras, J., Valencia-Garcia, R., Fernández-Medina, E., Alvarez, J., Piattini, M.: A systematic review and comparison of security ontologies. In: *3rd International Conference on Availability, Reliability and Security (ARES)*, pp. 813–820. IEEE, Barcelona (2008)
17. Herzog, A., Shahmehri, N., Duma, C.: An ontology of information security. *Int. J. Inf. Secur. Priv.* **1**(4), 1–23 (2007)
18. Undercoffer, J., Joshi, A., Pinkston, J.: Modeling computer attacks: an ontology for intrusion detection. In: Vigna, G., Kruegel, C., Jonsson, E. (eds.) *RAID 2003*. LNCS, vol. 2820, pp. 113–135. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45248-5_7



19. Geneiatakis, D., Lambrinouidakis, C.: An ontology description for SIP security flaws. *Comput. Commun.* **30**(6), 1367–1374 (2007)
20. Noy, N.F., McGuinness D.L.: *Ontology development 101: a guide to creating your first ontology*, pp. 1–25 (2001)
21. Souag, A., Salinesi, C., Comyn-Wattiau, I.: Ontologies for security requirements: a literature survey and classification. In: Bajec, M., Eder, J. (eds.) *CAiSE 2012*. LNBP, vol. 112, pp. 61–69. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31069-0_5
22. Boinski, T., Orłowski, P., Szymanski, J., Krawczyk, H.: Security ontology construction and integration. In: Filipe, J., Dietz, J.L.G. (eds.) *International Conference on Knowledge Engineering and Ontology Development (KEOD)*, pp. 369–374. INSTICC, Paris (2011)
23. Nguyen, V.: *Ontologies and information systems: a literature survey*. DSTO-TN-1002, Defence Science and Technology Organisation, Edingubrg, SA, pp. 66–92 (2011)
24. Blanco, C., Lasheras, J., Fernández-Medina, E., Valencia-García, R., Toval, A.: Basis for an integrated security ontology according to a systematic review of existing proposals. *Comput. Stand. Int.* **33**, 372–388 (2011)
25. Kitchenham, B.: Procedures for performing systematic reviews. *Keele UK Keele Univ.* **33**(2004), 1–26 (2004)
26. IEEE Xplore. <https://www.ieee.org>. Accessed 14 Aug 2022
27. Scopus. <https://www.scopus.com/search/form.uri?display=basic>. Accessed 14 Aug 2022
28. Web of Science. <https://www.webofscience.com/>. Accessed 14 Aug 2022
29. Adach, M., Hänninen, K., Lundqvist, K.: Search results of security ontologies 1988–2022, Technical report, MDU, Västerås. https://www.es.mdh.se/pdf_publications/6424.pdf. Accessed 14 Aug 2022
30. Schumacher, M.: 6. Toward a security core ontology. In: Schumacher, M. (ed.) *Security Engineering with Patterns*. LNCS, vol. 2754, pp. 87–96. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45180-8_6
31. Dritsas, S., et al.: Employing ontologies for the development of security critical applications. In: Funabashi, M., Grzech, A. (eds.) *I3E 2005*. IIFIP, vol. 189, pp. 187–201. Springer, Boston, MA (2005). https://doi.org/10.1007/0-387-29773-1_13
32. Fenz, S., Ekelhart, A.: formalizing information security knowledge. In: *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS)*, pp. 183–194. ACM, New York (2009)
33. Wang, J.A., Guo, M.: OVM: an ontology for vulnerability management. In: *5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies (CSIIRW)*, Oak Ridge Tennessee, USA, pp. 1–4 (2009)
34. Pereira, T., Santos, H.: An ontology approach in designing security information systems to support organizational security risk knowledge. In: *International Conference on Knowledge Engineering and Ontology Development (KEOD)*. SSEO, vol. 1, pp. 461–466, ScitePress, Barcelona (2012)
35. ISO/IEC_JTC1 27005:2008: information technology - security techniques - information security risk management, ISO, Technical report (2008)
36. Ramanauskaitė, S., Olifer, D., Goranin, N., Cenys, A.: Security ontology for adaptive mapping of security standards. *Int. J. Comput. Commun.* **8**(6), 813–825 (2013)
37. Agrawal, V.: Towards the ontology of ISO/IEC 27005: 2011 risk management standard. HAISA, Frankfurt, Germany, pp. 101–111 (2016)

38. ISSA-UK. Information security for small and medium-sized enterprises, Information System Security Association, Technical report (2011)
39. Paulsen, C., Toth, P.: NISTIR 7621 small business information security: the fundamentals. NIST, US Department of Commerce (2016)
40. Payment card industry data security standard (PCIDSS), PCI-Security Standard Council, Technical report (2006). https://www.commerce.uwo.ca/pdf/PCI_DSS_v3-2-1.pdf. Accessed 14 Aug 2022
41. ISO/IEC 27005:2011 - information technology—security techniques—information security risk management, ISO, Technical report (2011)

Enterprise Architecture



Model-Based Construction of Enterprise Architecture Knowledge Graphs

Philipp-Lorenz Glaser¹, Syed Juned Ali¹, Emanuel Sallinger²,
and Dominik Bork¹

¹ Business Informatics Group, TU Wien, Vienna, Austria

{philipp-lorenz.glaser,syed.juned.ali,dominik.bork}@tuwien.ac.at

² Database and Artificial Intelligence Group, TU Wien, Vienna, Austria
emanuel.sallinger@tuwien.ac.at

Abstract. Enterprise Architecture offers guidelines for the coherent, model-based design and management of enterprises. EA models provide a layered, integrated, and cohesive representation of the enterprise, enabling communication, analysis, and decision making. With the increasing size of EA models, automated analysis becomes essential. However, advanced model analysis is neither incorporated in current EA methods like ArchiMate nor supported by existing EA tools like Archi. Knowledge Graphs (KGs) can effectively organize and represent knowledge and enable reasoning to utilize this knowledge, e.g., for decision support. This paper introduces a model-based Enterprise Architecture Knowledge Graph (EAKG) construction method and shows how starting from ArchiMate models, an initially derived EAKG can be further enriched by EA-specific and graph characteristics-based knowledge. The introduced EAKG entails new representation and reasoning methods applicable to EA knowledge. As a proof of concept, we present the results of a first Design Science Research Cycle aiming to realize an Archi plugin for the EAKG that enables analysis of EA Smells within ArchiMate models.

Keywords: Enterprise Architecture · Knowledge Graph · Modeling tool · ArchiMate · Archi

1 Introduction

The transformation of information systems triggered enterprises to adopt enterprise architecture (EA) as a means to manage the complexity and evolution of the enterprise [8]. EA enables comprehensive management and decision-making based on the models of the organization. An enterprise is typically described through multiple EA layers such as *Business*, *Application*, and *Technology*. EA models are graphical representations that provide valuable support, e.g., integrated IT and business decision-making [12], planning future states of the enterprise, and improving the business and IT alignment [19]. To support all these functions, EA models need to be analyzed efficiently. Such EA analysis involves

querying models to evaluate various properties [38]. However, holistic EA models grow in size and complexity, thereby hampering manual human analysis, while advanced and automated analysis of EA models is surprisingly underrepresented in research and EA tooling so far [50].

EA modeling tools do not take full advantage of the several structural properties of EA models represented as graphs, such as the differentiation of relations between elements, the discovery of paths, clusters, or graph metrics. Current approaches are often tied to a concrete EA approach, offering a limited set of visualization techniques. EA modeling tools offer different features based on the supported EA approach and the analytical capabilities provided and thus restrict the kind of analysis they support [37]. The need for proper tool support was pointed out in the past as one EA [47] and business information systems modeling [21] research gap. We instantiate this gap in the following by substantiating a need for a generic and advanced EA analysis tool that utilizes the full potential of the graphical structure of EA models.

Knowledge Graphs (KG) represent interlinked descriptions of entities - objects, events, and concepts. Recently, the use of KGs in conceptual modeling, model-driven software engineering, and EA has been explored (cf., [11, 34, 44, 46, 49]). KGs can organize and represent knowledge to ease advanced reasoning (e.g., rule-based and machine learning-based) [15] and to provide question answering, recommendation, and information retrieval solutions [54].

In the context of EA, graph-based formalisms have been applied for the representation and reasoning of EA models [46, 50]. However, these works are merely constrained to the explicit knowledge encoded by the EA model (i.e., no further knowledge enrichment) and basic model analysis (i.e., no KG reasoning). We propose the model-based construction and enrichment of Enterprise Architecture Knowledge Graphs (EAKGs) to exploit the benefits of KG-based representation and reasoning in EA. EAKGs enable AI-based applications for EA model analysis. We further report on developing an EAKG plugin for the Archi toolkit. The plugin visualizes and analyses the EAKG and supports the EAKG knowledge enrichment. The EAKG provides a generic and unified intermediary representation of EAs, making our approach easily extensible for integrating other graph-based EA analysis tools. Our main contributions thus include (i) model-based construction and enrichment of EAKG, (ii) development of an Archi plugin for analysis and visualization of the EA models, and (iii) feasibility evaluation using a case-based approach.

This work reports Design Science Research (DSR) [27]. In particular, we build and evaluate the EAKG plugin for Archi that implements our conceptual contribution, the *model-based construction of EAKGs*.

In the remainder of this paper, first, Sect. 2 presents the relevant backgrounds and related works on EA Management, KGs, and their combination. We propose an approach for model-based construction of EAKGs in Sect. 3. The developed EAKG Archi plugin is presented in Sect. 4. Section 5 reports the results of a case-based evaluation before we finally providing a conclusive discussion of this paper in Sect. 6.

2 Background

Enterprise Architecture Management (EAM) is a “*management practice that establishes, maintains and uses a coherent set of guidelines, architecture principles and governance regimes that provide direction for and practical help with the design and the development of an enterprise’s architecture in order to achieve its vision and strategy*” [1]. The most used modeling language, standardized by the Open Group, is ArchiMate [32,39]. ArchiMate adopts a layered view of an enterprise depicted by the ArchiMate Framework, where the core entities of an enterprise are categorized along *layers* and *aspects*. A strength of ArchiMate is the ability to cover relevant aspects of an enterprise in a holistic and integrated manner. Shortcomings of ArchiMate are its limited semantic specificity [41] and the limited processing of the modeled information [13]. One of the most widely used EA modeling tools is Archi¹.

2.1 Enterprise Architecture Analysis

EA analysis concerns using EA models to analyze selected properties to provide decision support. Barbosa et al. [4] defined a taxonomy to classify EA research according to their analysis concerns, analysis techniques, and modeling languages employed to ease value extraction from EA models. A comprehensive survey of research on EA analysis techniques is presented by Buckl et al. [10]. The authors indicate a lack of automated analysis techniques that also scale well. A more recent survey yielded that “*Modern analysis approaches should combine interactive visualizations with automated analysis techniques*” [33]. The study by Santana et al. [47] reveals the need to develop proper tooling for EA analysis. Närman et al. [37] present a framework based on the ArchiMate metamodel for assessing four properties: application usage, system availability, service response time, and data accuracy. Florez et al. [19] present a catalog of automated analysis methods for enterprise models in a standardized modeling language and further implement the methods in a modeling tool. Domain ontologies have been applied for the representation, domain-knowledge enrichment, and analysis of EA models [14].

2.2 Graph-Based Analysis of EA Models

Aside from the previously presented approaches that base the analysis on a specific EA framework or modeling language, we focus on the following approaches that utilize graph-based representation analysis of EAs. With increasing model size and complexity, ArchiMate models can get difficult to comprehend by humans. Graph visualizations can be compelling as they further abstract the different ArchiMate elements to the two basic concepts, i.e., *nodes* and *edges*. Graph visualizations can be easily customized. Furthermore, storing a graph in a graph database enables the efficient execution of complex queries over large graphs.

¹ <https://www.archimatetool.com/>, last accessed: 15.08.2022.

Transforming EA models into graphs [4] or Linked Data [31,42] to enable semantic analysis is not new. Such EA analysis focuses on quantitative graph theory, which measures (i.e., quantifies) structural aspects of graphs. Many quantitative graph measures exist like *PageRank* and *Betweenness* (cf. [17]). Caetano et al. [14] map the conceptual schemas of EA models to an (upper-level) ontology and present their further analysis through logical inference or graph analysis. Several works use graphs for maintaining and optimizing EAs. Giakoumakis et al. [23] replace existing services with new services while aiming not to disrupt the organization using multi-objective optimization on a graph representation of the EA model. Similarly, Franke et al. [22] use a binary integer programming model to optimize the relation between IT systems and processes. Prediction based on EA models, represented as graphs, has been proposed by MacCormack et al. [35] using Design Structure Matrices to analyze the coupling between EA components and Hacks and Lichter [26] using a probabilistic approach that considers different scenarios. Holschke et al. [29] perform failure impact analysis with Bayesian Belief Networks, and Buschle et al. [13] adapt ArchiMate by fault trees to analyze the availability of EA components. Plataniotis et al. [43] present decision design graphs to analyze, e.g., how the decisions taken on the business level affect decisions on the technology level.

2.3 Knowledge Graphs and Enterprise Architecture

Knowledge Graphs (KGs) have, since their popularization by Google in 2012, seen widespread adoption in academia and industry. They have been used to derive “world knowledge graphs” such as Google’s KG, or DBpedia [2], but also “enterprise knowledge graphs” that represent more specific application domains [5]. In the case of this paper, the domain is EA itself.

KGs have been applied to represent different kinds of conceptual models like genomic datasets [7]. The Resource Description Framework (RDF) is the most common representation of KGs, and transformations from conceptual models to RDF have been proposed [52]. KG-based representation of models also enables reasoning methods for KGs, including logic-based and machine learning-based reasoning. Examples include supporting the analytic process [46] and more general reasoning contexts for conceptual modeling [36]. Bakhshadeh et al. [3] proposed the transformation of ArchiMate models into a Web Ontology Language (OWL) representation that enables consistency and completeness analysis of the EA models. However, OWL-based reasoning does not utilize the graph-based structural properties of EA models for analysis.

More recently, the first works proposed transforming EA models into KGs [49,50] with initial experimental results toward using the KG for EA Smell detection [51]. This current paper extends this stream of research by first proposing a generic model-based Knowledge Graph construction process (see Sect. 3). While previous works concentrated on transforming the syntactic nature of EA

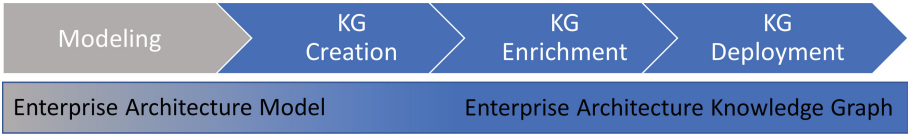


Fig. 1. Model-based knowledge graph construction process

models into equivalent KGs, in the proposed process, we emphasize how to augment knowledge of the KG from graph algorithms and EA Smells analysis by following an approach that comprises *knowledge creation*, *knowledge enrichment*, and *knowledge deployment* phases. As a final extension, we implement tool support for our approach using an Archi plugin (see Sect. 4).

3 Model-Based EA Knowledge Graph Construction

Various approaches exist for structuring the creation and the life cycle of KGs (cf. [6, 40, 48]). While no definitive procedure exists yet, many of the referred approaches focus on some form of *creation*, *evolution/enrichment*, and *deployment/use*. In this work, we introduce a **Model-based Knowledge Graph Construction Process** (see the upper part of Fig. 1), which is applied to the EA domain in order to construct an Enterprise Architecture Knowledge Graph (EAKG) (see the lower part of Fig. 1). The process structures the transition from EA modeling toward constructing a KG that enables representation, enrichment, and reasoning of EA knowledge. Initially, knowledge is extracted and created from the EA models in the **KG Creation** phase. Further knowledge is enriched, and additional inferences are made using that knowledge in the **KG Enrichment** phase. Finally, **KG Deployment** tailors the resulting KG to specific applications and facilitates different reasoning and representation approaches.

3.1 Knowledge Graph Creation

In the KG Creation stage, the EA model’s relevant information is extracted to create the initial EAKG. Conceptual models follow a schema and provide metadata (e.g., naming and classification) for objects, relations, and properties most often specified by metamodels [9]. This information can facilitate the KG creation stage by mapping it to specific nodes, edges, and properties in the EAKG. The specificity of models and the expressiveness of the used modeling language here clearly plays a significant role in the quality and richness of the initial KG. The transformation of languages that already conform to a graph structure (such as ArchiMate) into a KG structure is straightforward, while for other languages, a deeper investigation of meaningful mappings is necessary.

In our work, we use ArchiMate models and map the ArchiMate metamodel to the KG metamodel (see Fig. 2). The knowledge graph metamodel is inspired from [16]. The ArchiMate metamodel consists of different kinds of elements that

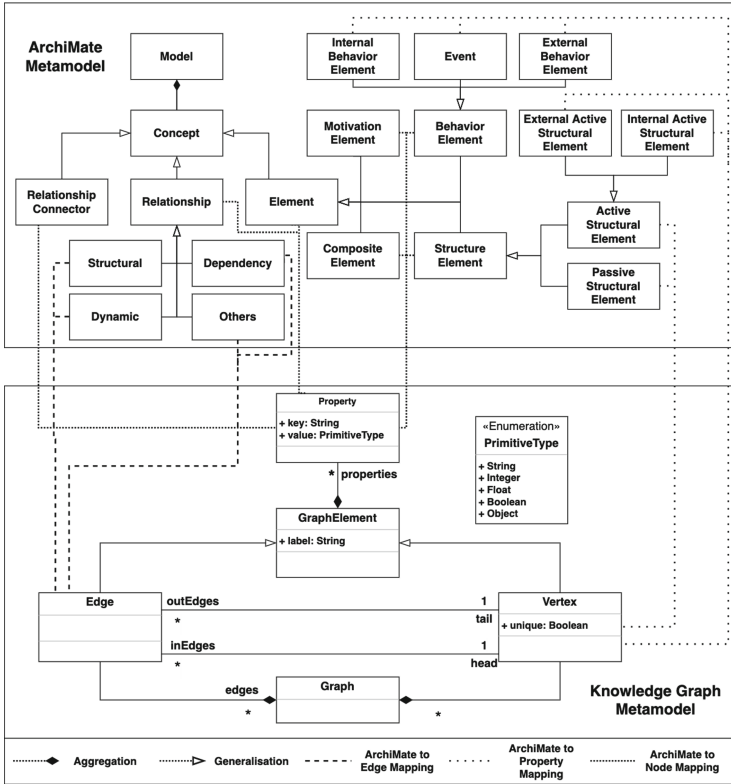


Fig. 2. ArchiMate behavioral and structural elements to KG metamodel mapping (KG metamodel inspired from [16])

are structurally categorized. A relationship is divided into a *structural*, *dependency*, *dynamic*, or *others* category. An EA relationship is mapped to the *edge* of the KG metamodel. A relationship connector is a property of an EA relationship; therefore, it is mapped to the property of the KG metamodel. The concrete behavioral and structural elements are mapped to the *node*. The abstract details of elements are mapped to the *property* of KG metamodel. For example, an *Active Structural Element* will be mapped to a node of KG metamodel, whereas it inherits the properties of a *Structure Element* and therefore, *Structure Element* is mapped to the property of KG metamodel. EAKG, using this mapping, thereby captures the semantics of EA model elements in the properties, and the structural elements are mapped to the nodes and edges of the KG.

Hoeffler [28] introduced the notion of *type semantics* and *inherent semantics* as two contributing aspects to derive the “full semantic description of model elements”. Type semantics is defined by the metamodel concepts and their properties themselves. Therefore, this kind of semantics is invariant to all instantiations and applies transitively to them. Once modelers create a model by instan-

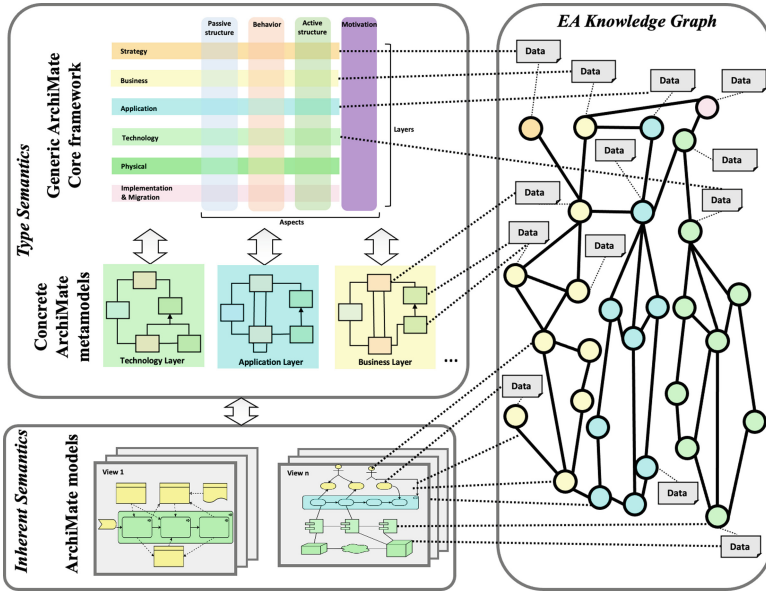


Fig. 3. EAKG creation

tiating metamodel concepts, they define the inherent semantics. These kinds of semantics are contingent on the modeler and the modeled case. The ArchiMate generic framework comprises different layers for representing different enterprise viewpoints. Each layer comprises active, passive, behavioral, and motivational aspects. Each aspect is formed by different elements shown in the ArchiMate metamodel in Fig. 2. Figure 3 visualizes our approach of transforming ArchiMate models into a property graph-based EAKG using the mappings from Fig. 2. The EA to KG mapping thereby, incorporates the type and inherent semantics from EA models. From the generic ArchiMate framework, we can transform meta-data like ArchiMate *layers* and *aspects* (shown in different colors in Fig. 3) into properties of nodes and edges of the EAKG. We can further derive type semantics from the concrete ArchiMate layer-specific metamodels like the *Application Layer* metamodel. We derive the inherent semantics from the concrete ArchiMate models, like the elements' names and connections.

3.2 Knowledge Graph Enrichment

The next step of the KG construction process is focused on knowledge enriching through general and domain-specific knowledge. Enriching such knowledge results in additional labels, properties, and edges in the EAKG. In this work, we enrich the initial EAKG by *Graph characteristics* (i.e., general graph-theoretical knowledge) and *EA Smells* (i.e., domain-specific EA knowledge). Hacks et al. introduce the concept of EA Smells [25] analogous to Code Smells [20] in the software engineering domain. EA Smells signify bad modeling practices and allow

architects to discover possible flaws in their models. A catalog of EA Smells was published [45], and the website² serves as a knowledge base, currently listing 63 EA smells.

Graph Characteristics. Graph characteristics describe quantitative aspects of the KG regarding structural characteristics (e.g., *centralities* and *communities*). The initial graph expands with new properties for nodes, representing the score of specific graph algorithms. The score property can then be used for new KG representations, e.g., *node size* to highlight centralities or *color* to differentiate communities (exemplified in Sect. 5).

EA Smells. An approach for KG-based EA Smell Detection has been introduced in [51]. We adopt some of the proposed smell detection queries and use them during the knowledge enrichment step to enrich the EAKG with EA Smell-specific knowledge. Detected smells expand the EAKG structure with labels on affected nodes or by adding relationships when multiple objects are affected by a smell (e.g., introducing a new edge with the label ‘duplication’ between two nodes in case the *Duplication* smell detected duplicate nodes in the EAKG).

3.3 Knowledge Graph Deployment and Application

This step focuses on deploying the EAKG and allowing it to power various applications and use cases. The complexity of the preceding steps is entirely concealed to the user, i.e., to enterprise architects. The deployment takes the entire EAKG and provides functionality to explore, represent, and reason the EA knowledge efficiently. The architecture and implementation details are provided in the tool paper [24].

Still, the user can identify the provenance of each node, property, or edge or limit the scope of the EAKG to individual parts of the enterprise architecture. Thus, the primary focus of the deployment step is to make the EAKG easily accessible while providing features to work with the knowledge enrichment (e.g., only represent EA elements of a particular layer, only show parts of an EA that are affected by an EA Smell). We describe these (and more) features in great detail throughout the remainder of this paper, but in particular in Sect. 5.

4 An EAKG Archi Plugin

The result of the first iteration of the DSR life cycle is an initial prototype, employing the discussed concepts of Sect. 3. In particular, the prototype realizes automated reasoning and representation of EA knowledge based on the previously introduced *Model-based Knowledge Graph Construction Process* as a plugin for the Archi modeling toolkit. The plugin aims to make Knowledge Graph-based EA analysis available to enterprise architects, i.e., an audience that not necessarily has graph-theoretic knowledge.

² <https://swc-public.pages.rwth-aachen.de/smells/ea-smells/>, accessed: 11.05.2022.

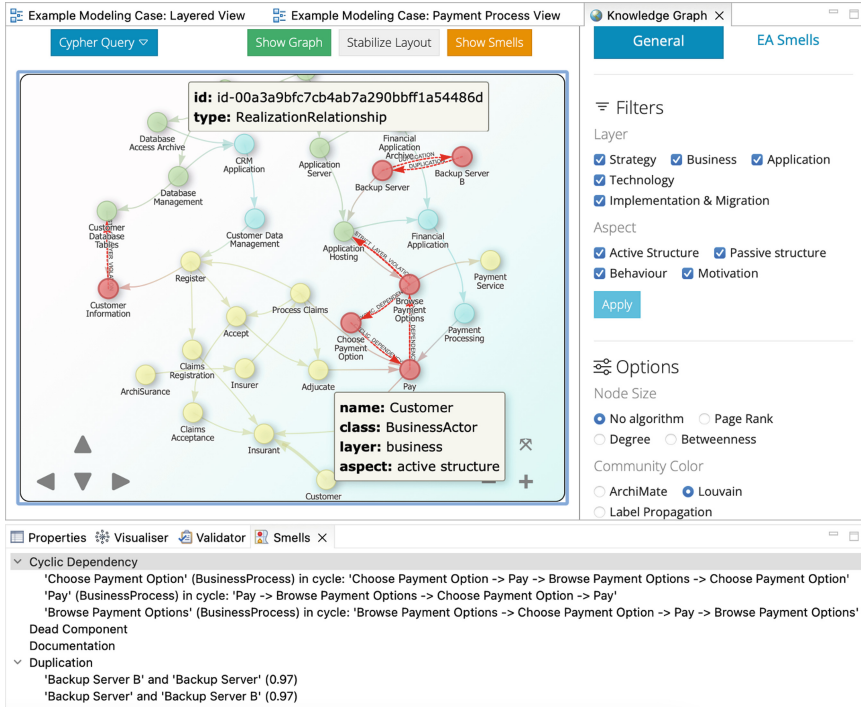


Fig. 4. EAKG plugin in Archi with graph view and smells report view (Color figure online)

Figure 4 shows the integration of the plugin within the Archi application, containing both the *Graph View*, and the *Smells Report View*. The node colors in the EAKG are derived from the ArchiMate core framework based on the EA layer. The used EA model is a modified version of the ArchiSurance case study – for details, see Sect. 5.

Knowledge Graph Visualization. The main view in the center visualizes the EAKG. Nodes denote ArchiMate elements, while edges denote ArchiMate relationships. The colors of nodes derived from the ArchiMate core framework. Further properties are exposed by hovering over nodes and edges, as exemplified by the *Customer* element in the figure and the relationship at the *Database Access Archive* and *CRM Application*. The toolbar at the top allows the execution of custom cypher queries on the EAKG.

Graph Characteristics Visualisation. The right-hand sidebar includes a filter and options menu. Here, enterprise architects can easily filter the displayed elements from specific layers or aspects of ArchiMate. The option menu on the bottom right offers configurations for the *Graph characteristics Knowledge Graph Enrichment* introduced in Sect. 3. Graph centrality measures are reflected via the node size, whereas community measures are reflected via node color.

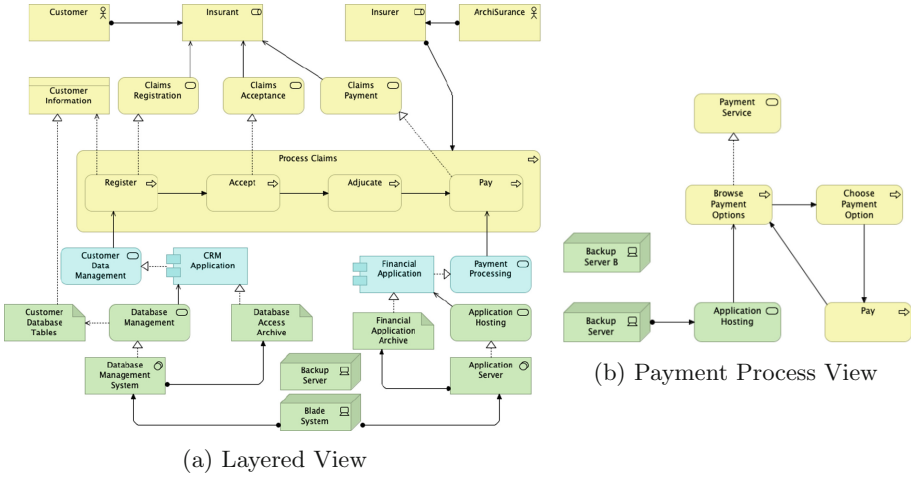


Fig. 5. Excerpts of the original EA model used during the case study (color of layers for model elements derived from the ArchiMate core framework) (Color figure online)

EA Smells Detection. The Report view at the bottom lists all EA Smells and the affected elements in the model. The EA Smells view shows the affected elements highlighted in red and references to other elements of the smell represented as dashed, red edges. The EA Smells tab in the sidebar provides information about each EA Smell, including a visualization, a description, and a solution (for examples, see Sect. 5).

5 Case-Based Evaluation

In order to evaluate the feasibility of constructing the EAKG (see Fig. 3) and using it as means of reasoning and representing EA knowledge, including EA Smells, we present a case-based evaluation of an ArchiMate model based on the popular ArchiSurance case study [30]. The model consists of multiple viewpoints, three of which are visualized in Fig. 5a: the *Layered Viewpoint*, which is reused from the ArchiSurance model and in the Fig. 5b *Payment Process Viewpoint*. Different colors in the models denote the various ArchiMate layers. The case study extends the ArchiSurance case by the requirements and realisation of online shopping and portfolio management of insurance products.

Figure 4 already shows the resulting initial graph structure and how the ArchiMate elements and relationships are mapped to nodes and edges in the EAKG. The transformation maps the properties related to layers (e.g., *Strategy*, *Business*) and aspects (e.g., *Active Structure*, *Passive Structure*) in the original EA model to the properties of the nodes in the resulting EAKG. The relationship type (e.g., *realisation*, *assignment*, *association*) are stored in the properties of the relationships in the EAKG. Further information about the relationships is derived and stored as edge properties in EAKG; for example,

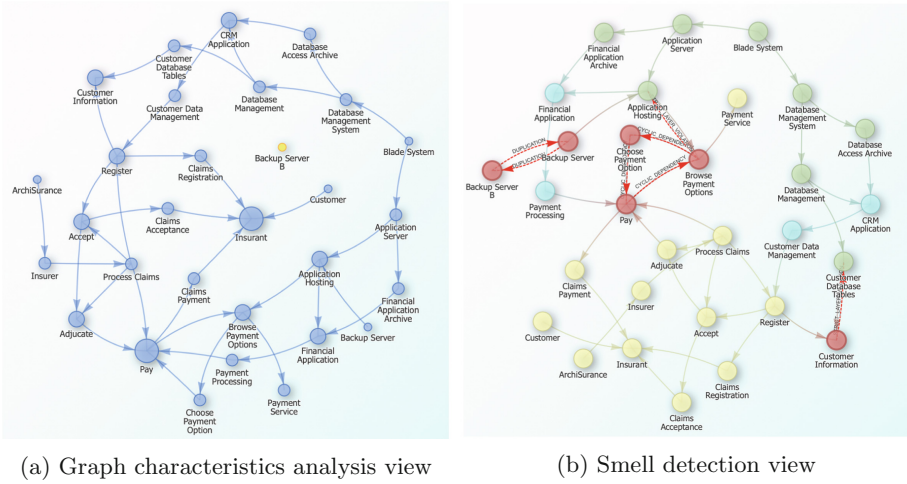


Fig. 6. KG-based EA analysis representations in EAKG (Color figure online)

realisation relationship is a structural relationship, and association relationship is a dependency relationship. Therefore, information about the relationship like *structural* or *dependency* is derived from the relationship type and stored as an edge property. In this way, the structural aspects are stored as nodes and edges and the semantic aspects of the EA elements (related to layers and aspects and further details about model elements) are stored as properties of nodes and edges in the EAKG. Next, we show the enrichment and analysis of the EAKG using Fig. 6.

Graph-Based Analysis. After the KG construction, the EAKG can be further analyzed and enriched by applying graph algorithms. Figure 6(a) visualises the resulting graph after executing the *Weakly Connected Components* graph algorithm. *Node Size* is set to *Degree* and the *Community Color* to *Weakly Connected Components*. The degree denotes the number of connections, and, as can be seen, the size of nodes increases with the amount of incoming and outgoing edges. The Weakly Connected Components algorithm helps identify disconnected sub graphs by assigning each node that is part of the same sub graph with the same color. In our case, the graph consists of two disconnected subgraphs with different colors each. The degree of the node denotes the importance of each node. In our case, the Insurant can be seen as an important node because of its biggest size and provides insights to the modelers about checking the incoming and outgoing edges.

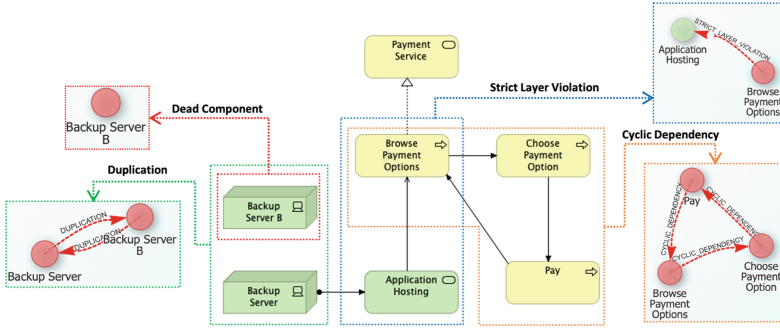


Fig. 7. EA smells

EA Smell Detection. The EA model of our case was designed to include different EA Smells their representation in the enriched KG. We, e.g., added a *Long Documentation* text to the ArchiSurance element (in the Layered Viewpoint). Throughout the case we further added *Dead Component*, *Strict Layer Violation*, and *Cyclic Dependencies* smells. As can be seen in Fig. 6(b), nodes that are part of a detected EA Smell are highlighted in red, while edges to other nodes (that are also part of the detected EA Smell) are represented by a newly introduced dashed red edge with the name of the detected EA Smell as a label. The present smells in the viewpoints are displayed in Fig. 7, with individual representations mapping for each smell to the source model elements. Note that the EA Smells and the KG-based EA Smell detection have been introduced previously. We refer the interested reader to the dedicated literature [45, 51]. We contribute here a much richer representation of EA Smells that again uses a Knowledge Graph instead of a textual analysis proposed in previous research.

6 Conclusive Discussion

Our approach and the toolkit enable many possibilities for generalization (e.g., to other modeling languages) and extension (e.g., to incorporate further EA Smells or other knowledge). It is important to note that the metadata of an EA model captured by EAKG (e.g., metamodel level properties of the elements and relationships) are not fully utilized in the presented analysis; however, our EAKG transformation enables the possibility of applying KG-based techniques to analyze and process the EA models and further support the modelers. Ontologies can be integrated into the EAKG; therefore, foundational or domain ontologies can be linked to the EAKG for knowledge enrichment of the EA models and further apply reasoning. The conceptual schema of the EA models can be mapped to an ontology for knowledge enrichment and annotation for e.g., a health domain ontology can enrich the semantics of EA models of a hospital.

EAKG enables the KG-based AI applications for the semantic processing of EA models to support reasoning (e.g., inference-based, machine learning-based),

integration (e.g., ontology mapping). EAKG provides the feasibility of applying machine learning techniques. The semantic relationships between model instance data and the models, along with the labels and the metadata (metamodel labels), can use NLP to predict links from a model element to an ontology element. Models can be mapped to common ontologies to further support interoperability. Graph Neural Networks (GNN) support applications such as node, edge, and even graph classification, link prediction between entities [53]. GNNs has been applied for UML model completion [18]. Similarly, GNNs with NLP techniques can be applied on EAKGs to support element recommendation or model auto-completion during modeling. GNNs can further transform an EAKG into a vector space with encoded specific domain information, enabling a domain-specific EA model semantic search.

In order to cope with the increasing complexity of maintaining and analyzing overarching enterprise architecture models, in this paper, we proposed an approach for model-based Enterprise Architecture Knowledge Graph (EAKG) construction and means to enrich type semantics, inherent semantics, general graph knowledge, and domain-specific enterprise architecture knowledge into the resulting EAKG. To evaluate our approach's feasibility and make it available to enterprise architects, we developed a first tool prototype, an EAKG plugin to the widely used Archi toolkit. Our approach enables full automation for the entire EAKG construction process and provides an efficient and intuitive GUI to explore and analyze the enterprise architecture knowledge graph.

The most innovative contribution we make with this paper is the not trivial enrichment of the EAKG from multiple sources and using the KG for EA analysis and representing EA knowledge using, e.g., the added nodes and relationships for EA Smells. Consequently, we propose to not only use KGs for automated analysis of overarching EA models, but also to improve human understandability by appropriate graph visualizations.

In this paper, we report on the results of a first Design Science Research cycle that aims at integrating all relevant sources into a fully-packed plugin archive. Future prototypes will emphasize a more lightweight plugin that interconnects the EA Smells catalog [45]. Instead of hard-coding the smell detection queries, such integration would enable us to always use the latest set of EA Smells together with their detection queries. Such a distributed system, of course, requires adequate infrastructure and latency, so we intended not to go along that path in developing the prototype. Moreover, instead of integrating a neo4j database, the plugin could easily connect to an existing neo4j instance.

A limitation of this research is the fact that we present a single case in this paper – previous works showed the scalability of the Graph-based EA Smell detection [51]. Further, as we aim to support enterprise architects with our approach, too, we need to engage in empirical evaluations to test the hypothesis on the perceived usefulness, ease of use, and intention to use EAKG in practice. Indeed, we are currently in discussions with a German-based international company on using and evaluating our approach.

As we know that showing larger models or graphs in a paper format might limit comprehensibility, we also created a demo video showcasing the EAKG Archi plugin in action. The video shows the core functionality and the case study example in detail and is accessible via: <https://youtu.be/gcXiAWDJDes>. The implementation of the EAKG plugin is open source³ and we aim to list it on the Archi plugins page for researchers, teachers and EA practitioners⁴.

Acknowledgements. This work has been partially funded through the Erasmus+ KA220-HED project “Digital Platform Enterprise” (DEMO) with the project number: 2021-1-RO01-KA220-HED-000027576, the project “Enterprise Architecture Knowledge Graphs” funded by a Career Grant of TU Wien, and the Austrian Research Promotion Agency (FFG) via the Austrian Competence Center for Digital Production (CDP) under the contract number 854187.

References

1. Ahlemann, F., Stettiner, E., Messerschmidt, M., Legner, C.: Strategic Enterprise Architecture Management: Challenges, Best Practices, and Future Developments. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-3-642-24223-6>
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
3. Bakhshadeh, M., Morais, A., Caetano, A., Borbinha, J.: Ontology transformation of enterprise architecture models. In: Camarinha-Matos, L.M., Barrento, N.S., Mendonça, R. (eds.) DoCEIS 2014. IAICT, vol. 423, pp. 55–62. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54734-8_7
4. Barbosa, A., Santana, A., Hacks, S., Stein, N.V.: A taxonomy for enterprise architecture analysis research. In: 21st International Conference on Enterprise Information Systems, vol. 2, pp. 493–504. SciTePress (2019)
5. Bellomarini, L., Fakhoury, D., Gottlob, G., Sallinger, E.: Knowledge graphs and enterprise AI: the promise of an enabling technology. In: 35th IEEE International Conference on Data Engineering, pp. 26–37. IEEE (2019)
6. Bellomarini, L., Sallinger, E., Vahdati, S.: Chapter 2 Knowledge graphs: the layered perspective. In: Janev, V., Graux, D., Jabeen, H., Sallinger, E. (eds.) Knowledge Graphs and Big Data Processing. LNCS, vol. 12072, pp. 20–34. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53199-7_2
7. Bernasconi, A., Canakoglu, A., Ceri, S.: From a conceptual model to a knowledge graph for genomic datasets. In: Laender, A.H.F., Pernici, B., Lim, E.-P., de Oliveira, J.P.M. (eds.) ER 2019. LNCS, vol. 11788, pp. 352–360. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33223-5_29
8. Bork, D., et al.: Requirements engineering for model-based enterprise architecture management with ArchiMate. In: Pergl, R., Babkin, E., Lock, R., Malyzhenkov, P., Merunka, V. (eds.) EOMAS 2018. LNBIP, vol. 332, pp. 16–30. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00787-4_2

³ EAKG Github repository: <https://github.com/borkdominik/archi-kganalysis-plugin>.

⁴ Archi plugins: <https://www.archimatetool.com/plugins/>, accessed 02.05.2022.

9. Bork, D., Karagiannis, D., Pittl, B.: A survey of modeling language specification techniques. *Inf. Syst.* **87**, 101425 (2020). <https://doi.org/10.1016/j.is.2019.101425>
10. Buckl, S., Matthes, F., Schweda, C.M.: Classifying enterprise architecture analysis approaches. In: Poler, R., van Sinderen, M., Sanchis, R. (eds.) *IWEI 2009. LNBIP*, vol. 38, pp. 66–79. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04750-3_6
11. Burgueño, L., Kessentini, M., Wimmer, M., Zschaler, S.: 3rd workshop on artificial intelligence and model-driven engineering. In: *International Conference on Model Driven Engineering Languages and Systems Companion*, pp. 148–149 (2021)
12. Buschle, M., Holm, H., Sommestad, T., Ekstedt, M., Shahzad, K.: A tool for automatic enterprise architecture modeling. In: Nurcan, S. (ed.) *CAiSE Forum 2011. LNBIP*, vol. 107, pp. 1–15. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29749-6_1
13. Buschle, M., Johnson, P., Shahzad, K.: The enterprise architecture analysis tool - support for the predictive, probabilistic architecture modeling framework, pp. 3350–3364 (2013)
14. Caetano, A., et al.: Representation and analysis of enterprise models with semantic techniques: an application to archimate, e3value and business model canvas. *Knowl. Inf. Syst.* **50**(1), 315–346 (2017)
15. Chen, X., Jia, S., Xiang, Y.: A review: knowledge reasoning over knowledge graph. *Expert Syst. Appl.* **141**, 112948 (2020)
16. Daniel, G., Sunyé, G., Cabot, J.: UMLtoGraphDB: mapping conceptual schemas to graph databases. In: Comyn-Wattiau, I., Tanaka, K., Song, I.-Y., Yamamoto, S., Saeki, M. (eds.) *ER 2016. LNCS*, vol. 9974, pp. 430–444. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46397-1_33
17. Dehmer, M., Emmert-Streib, F., Shi, Y.: Quantitative graph theory: a new branch of graph theory and network science. *Inf. Sci.* **418–419**, 575–580 (2017)
18. Di Rocco, J., Di Sipio, C., Di Ruscio, D., Nguyen, P.T.: A GNN-based recommender system to assist the specification of metamodels and models. In: *International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pp. 70–81 (2021)
19. Florez, H., Sánchez, M., Villalobos, J.: A catalog of automated analysis methods for enterprise models. *Springerplus* **5**(1), 1–24 (2016). <https://doi.org/10.1186/s40064-016-2032-9>
20. Fowler, M.: *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, Boston (2018)
21. Frank, U., Strecker, S., Fettke, P., vom Brocke, J., Becker, J., Sinz, E.J.: The research field “modeling business information systems” - current challenges and elements of a future research agenda. *Bus. Inf. Syst. Eng.* **6**(1), 39–43 (2014)
22. Franke, U., Holschke, O., Buschle, M., Narman, P., Rake-Revelant, J.: It consolidation: an optimization approach. In: *International Enterprise Distributed Object Computing Conference Workshops*, pp. 21–26 (2010)
23. Giakoumakis, V., Krob, D., Liberti, L., Roda, F.: Technological architecture evolutions of information systems: trade-off and optimization. *Concurr. Eng.* **20**(2), 127–147 (2012)
24. Glaser, P.L., Ali, S.J., Sallinger, E., Bork, D.: Exploring enterprise architecture knowledge graphs in Archi: the EAKG toolkit (2022). Under review
25. Hacks, S., Höfert, H., Salentin, J., Yeong, Y.C., Lichter, H.: Towards the definition of enterprise architecture debts. In: *2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW)*, pp. 9–16. IEEE (2019)

26. Hacks, S., Lichter, H.: A probabilistic enterprise architecture model evolution. In: International Enterprise Distributed Object Computing Conference, pp. 51–57 (2018)
27. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* **28**(1), 75–105 (2004)
28. Höfferer, P.: Achieving business process model interoperability using metamodels and ontologies. In: Österle, H., Schelp, J., Winter, R. (eds.) European Conference on Information Systems, ECIS 2007, pp. 1620–1631 (2007)
29. Holschke, O., Närman, P., Flores, W.R., Eriksson, E., Schönherr, M.: Using enterprise architecture models and Bayesian belief networks for failure impact analysis. In: Feuerlicht, G., Lamersdorf, W. (eds.) ICSOC 2008. LNCS, vol. 5472, pp. 339–350. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01247-1_35
30. Jonkers, H., Band, I., Quartel, D.: The ArchiSurance case study. The Open Group, pp. 1–32 (2012)
31. Karagiannis, D., Buchmann, R.A.: Linked open models: extending linked open data with conceptual model information. *Inf. Syst.* **56**, 174–197 (2016)
32. Lankhorst, M.M.: Enterprise Architecture at Work - Modelling, Communication and Analysis. The Enterprise Engineering Series, 2nd edn. Springer, Heidelberg (2009)
33. Lantow, B., Jugel, D., Wißotzki, M., Lehmann, B., Zimmermann, O., Sandkuhl, K.: Towards a classification framework for approaches to enterprise architecture analysis. In: Horkoff, J., Jeusfeld, M.A., Persson, A. (eds.) PoEM 2016. LNBIP, vol. 267, pp. 335–343. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48393-1_25
34. Maass, W., Storey, V.C.: Pairing conceptual modeling with machine learning. *Data Knowl. Eng.* **134**, 101909 (2021)
35. Maccormack, A.D., Lagerstrom, R., Baldwin, C.Y.: A methodology for operationalizing enterprise architecture and evaluating enterprise it flexibility. Harvard Business School Working Paper Series# 15-060 (2015)
36. Medvedev, D., Shani, U., Dori, D.: Gaining insights into conceptual models: a graph-theoretic querying approach. *Appl. Sci.* **11**(2), 765 (2021)
37. Naranjo, D., Sánchez, M., Villalobos, J.: PRIMROSe: a graph-based approach for enterprise architecture analysis. In: Cordeiro, J., Hammoudi, S., Maciaszek, L., Camp, O., Filipe, J. (eds.) ICEIS 2014. LNBIP, vol. 227, pp. 434–452. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22348-3_24
38. Närman, P., Buschle, M., Ekstedt, M.: An enterprise architecture framework for multi-attribute information systems analysis. *Softw. Syst. Model.* **13**(3), 1085–1116 (2012). <https://doi.org/10.1007/s10270-012-0288-2>
39. OMG: ArchiMate® 3.1 Specification. The Open Group (2019). <https://pubs.opengroup.org/architecture/archimate3-doc/>
40. Pan, J.Z., Vetere, G., Gómez-Pérez, J.M., Wu, H. (eds.): Exploiting Linked Data and Knowledge Graphs in Large Organisations. Springer, Heidelberg (2017). <https://doi.org/10.1007/978-3-319-45654-6>
41. Pittl, B., Bork, D.: Modeling digital enterprise ecosystems with ArchiMate: a mobility provision case study. In: ICServ 2017. LNCS, vol. 10371, pp. 178–189. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61240-9_17
42. Pittl, B., Fill, H.: Transforming enterprise models to linked data via semantic annotations. In: Schaefer, I., Karagiannis, D., Vogelsang, A., Méndez, D., Seidl, C. (eds.) Modellierung 2018. LNI, pp. 55–70. Gesellschaft für Informatik (2018)

43. Plataniotis, G., de Kinderen, S., Proper, H.A.: Relating decisions in enterprise architecture using decision design graphs. In: 2013 17th IEEE International Enterprise Distributed Object Computing Conference, pp. 139–146. IEEE (2013)
44. Reimer, U., Bork, D., Fettke, P., Tropmann-Frick, M.: Preface of the first workshop models in AI. In: Companion Proceedings of Modellierung 2020 Short, Workshop and Tools & Demo Papers, pp. 128–129. CEUR Workshop Proceedings (2020)
45. Salentin, J., Hacks, S.: Towards a catalog of enterprise architecture smells. In: Gronau, N., Heine, M., Krasnova, H., Poustcchi, K. (eds.) Internationalen Tagung Wirtschaftsinformatik, Community Tracks, pp. 276–290. GITO Verlag (2020)
46. Santana, A., Fischbach, K., de Moura, H.P.: Enterprise architecture analysis and network thinking: a literature review. In: Bui, T.X., Jr., R.H.S. (eds.) 49th Hawaii International Conference on System Sciences, pp. 4566–4575. IEEE (2016)
47. Santana, A., Simon, D., Fischbach, K., de Moura, H.: Combining network measures and expert knowledge to analyze enterprise architecture at the component level. In: 2016 IEEE EDOC Conference, pp. 1–10. IEEE (2016)
48. Simsek, U., et al.: Knowledge graph lifecycle: building and maintaining knowledge graphs (2021)
49. Smajevic, M., Bork, D.: From conceptual models to knowledge graphs: a generic model transformation platform. In: International Conference on Model Driven Engineering Languages and Systems Companion, pp. 610–614 (2021)
50. Smajevic, M., Bork, D.: Towards graph-based analysis of enterprise architecture models. In: Ghose, A., Horkoff, J., Silva Souza, V.E., Parsons, J., Evermann, J. (eds.) ER 2021. LNCS, vol. 13011, pp. 199–209. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-89022-3_17
51. Smajevic, M., Hacks, S., Bork, D.: Using knowledge graphs to detect enterprise architecture smells. In: Serral, E., Stirna, J., Ralyté, J., Grabis, J. (eds.) PoEM 2021. LNBIP, vol. 432, pp. 48–63. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91279-6_4
52. Tong, Q., Zhang, F., Cheng, J.: Construction of RDF (S) from UML class diagrams. *J. Comput. Inf. Technol.* **22**(4), 237–250 (2014)
53. Zhou, J., et al.: Graph neural networks: a review of methods and applications. *AI Open* **1**, 57–81 (2020)
54. Zou, X.: A survey on application of knowledge graph. In: *Journal of Physics: Conference Series*, vol. 1487, p. 012016. IOP Publishing (2020)



Enterprise Architecture Management Support for Digital Transformation Projects in Very Large Enterprises: A Case Study at a European Mobility Provider

Oleg Kanin^(✉) and Paul Drews

Institute of Information Systems, Leuphana University Lüneburg, Lüneburg, Germany
oleg.kanin@stud.leuphana.de, paul.drews@leuphana.de

Abstract. Enterprises in various sectors execute digital transformation (DT) projects with a significant impact on their business and information technology (IT) architecture. The enterprise architecture management (EAM) function is designed to support these transformation endeavors. As DT projects are increasingly customer- and partner-driven, business-oriented and based on innovative digital technologies, the requirements for the EAM function are expected to change. With our case study, we investigate the changing requirements and potential new challenges associated with adapting and advancing the EAM function. This case study was conducted at one of the largest mobility service providers in Europe. Based on a case study grounded in expert interviews, we identified 39 changed and new requirements and compared them with requirements presented in the literature. The findings present new and changed requirements for adapting the EAM function to better support DT projects and to structure them according to EAM domains.

Keywords: Enterprise architecture (EA) · Enterprise architecture management · EA model · Digital transformation · Change requirements

1 Introduction

Technological progress and fast-paced innovation have spurred the adoption of information and communication technology (ICT) on a broad scale across a range of business areas. The adoption of ICT is significantly changing the contours of business processes and the IT landscape. This digital transformation is a critical task for enterprises and requires a fundamental understanding of their core processes and of available digital technologies [13]. DT aims at complementing and enriching existing products and services and enables the creation of entirely new business models [13]. Additionally, existing IT infrastructure must be adapted to current and expected future requirements [13].

Very large enterprises with employee numbers in the tens or even hundreds of thousands have DT projects with numerous transformation activities running simultaneously across different departments. This makes managing the enterprise's overall DT a highly

complex task. To tame this complexity and supporting the management of DT projects, appropriate methods and models are needed to make the transformation activities more visible, plannable and controllable. Many enterprises have established EAM in support of large-scale IT-based transformations. There is scant research, however, on the adoption of EAM in support of DT projects in large enterprises. In order to address this research gap, we conducted a case study designed to identify changes in EAM requirements in the context of large-scale DT projects. Hence, this study seeks to answer the question: How are EAM support requirements changing in very large enterprises as a result of the increased number and impact of DT initiatives?

This paper is structured as follows. The following section summarizes the theoretical basis and related work for this paper. In the third section, we describe the research method employed to answer the research question and provides details about data gathering and analysis. The fourth section presents the results of our study. The final sections of this paper comprise a discussion and a conclusion.

2 Theoretical Foundations and Related Work

In the following sections, we introduce the theoretical foundations of EA and EAM as well as how EA supports DT projects.

2.1 Enterprise Architecture

Since the mid-1980s, enterprise architecture models and metamodels have been developed to support decisions about the use of IT in organizations [18]. The goal of EA is to describe a structured and aligned collection of plans for the integrated representation of the business and IT landscape of the enterprise, in past, current, and future states [18].

An EA model is a specifically structured abstraction of the enterprise which includes both technical and business aspects such as application systems, business processes and strategy [16, 19]. It can be developed to depict the current state (as-is) as well as desired target states (to-be) of an enterprise [16, 19].

EAM is established in many enterprises as an instrument for supporting the planning and transformation of the enterprise's architecture [16, 19]. It leverages a model of all important business and IT elements as well as their relationships in an enterprise [16]. EAM aims to provide an overall view of existing and planned states of the architectures [19]. In addition, EAM deals with the documentation, analysis, planning, and operational readiness of an EA [16]. EAM comprises the demanding processes of supporting the development of viable, economical solutions in compliance with defined architectural standards [20].

Architectural methodologies offer a basic structure which can be used as a basis to develop architectural models and to set up EAM processes [20]. Architecture frameworks aim to provide a method to design the target state of the enterprise in terms of building blocks and their interaction [20]. A number of IT tools and recommended standards can be applied in the process [20].

By using architecture frameworks such as TOGAF, EA is changed continuously to address the stakeholders' needs [20]. The following four architectural domains are

often used to structure the EA: business strategy, data architecture, application architecture, and technology architecture [20]. The business strategy determines the business architecture, organizational structures and business processes [20]. The data architecture deals with the structure of the logical and physical data and with the database [20]. The application architecture plans the interaction of individual applications and their relationships to business processes [20]. The technology architecture describes the hardware and software features supporting business processes, data management, and application services [20]. Architecture principles, standards, and EA models for EA architecture are essential to EAM [19]. Failing to apply EA in enterprise strategy, project development and service management can lead to numerous problems, such as technical debt, redundancy, inconsistent communication, and disconnected development efforts [7].

2.2 Digital Transformation

In recent years, DT has become an important phenomenon in strategic information systems (IS) and as a driving technology for business and society [21]. IS research distinguishes between DT and digitization.

While digitization is the technical process of converting analog signals into digital form [13], DT can be understood as a process in which organizations respond to changes in their environment by using digital technologies to transform their value creation processes [21].

DT is driven through the adoption of digital technologies like cloud computing or internet of things (IoT) which have a major impact on enterprises' IT infrastructure [16]. EA practitioners increasingly embrace the cloud as a natural extension of enterprise architecture [18]. The areas where new process-automated and intelligent systems are being deployed are having a major impact on processes, products, services and business models. For instance, the internet of things [17], big data, mobile computing and cloud computing are connecting machines, things and people and enabling new models of work, collaboration and automation [13]. Cloud services provide low-cost access to very powerful IT infrastructures, and developments like cloud computing make it easier to outsource elements of the IT value chain [13].

These fast-paced developments are placing digital technologies center stage in creating and reinforcing transformation at the society and industry levels [21]. This transformation, in turn, triggers strategic responses of organizations [9]. To remain competitive, organizations are employing digital technologies to transform the way they create value [21]. This requires them to make structural changes and overcome barriers that hinder their transformation efforts [2]. These changes yield positive effects for enterprises and, in some cases, also for individuals and society, but can also have undesirable consequences [21].

2.3 Enterprise Architecture Support for DT Projects

The digital revolution has gradually led national and regional governments to defining digitization as a strategic priority and establish large-scale initiatives for the digital transformation of business and society [13]. Digital opportunities offer great potential for innovation and the development of digital business models, products and services [13].

Indeed, digital technologies are becoming more user-friendly [13], which has enabled enterprises to operate more successfully and sustainably in new digital sales markets.

The enterprise architecture models and metamodels have already proven their value in large enterprises as a useful approach to support their DT [3]. The large scale and high complexity of DT and the increasing number of DT projects at large enterprises have created a concomitant rise in requirements for EAM. The technology and application domains are inherently complex, and this complexity is further compounded by other factors, such as large numbers of project partners and customers, non-transparent collaboration, and the heterogeneous IT infrastructures of customers and partners [3]. Hence, it should come as no surprise when we read recent research showing that many DT projects fail partially or entirely [3]. In order to avoid such failures, EAM should provide better support for DT projects through improved EA models, metamodels, and processes.

EAM is already leveraged to support DT projects and is designed to address new challenges resulting from DT [16]. Enterprises use EAM to optimize the connection between IT and business units, implement the enterprise strategy in the best way possible, and generally create a more transparent and flexible IT infrastructure [19]. A flat organizational hierarchy, an effective implementation of the strategy and a competitive IT enable enterprises to better support the envisioned DT. For these and other reasons, EAM is of essential and strategic importance for DT in the enterprise. EAM can be beneficial for DT in several use cases. However, the potential benefit of EAM for DT is still largely unexplored [16]. Due to this lack of research, EAM support for DT is rarely discussed in current research.

The focus of this research lies in understanding changed and new requirements for EAM support in large enterprises in the context of DT projects. A better understanding of the changing requirements for EAM support might serve as a starting point for advancing the EAM function in large-scale enterprises. We seek to contribute to EAM research by developing a better understanding of changing and new requirements which the EAM has to face due to DT initiatives in large scale enterprises.

3 Research Approach

In the following sections, we describe the research method, the context of the case study as well as the data gathering and analysis.

3.1 Research Method

To address the research question, we chose to conduct a case study at a large European mobility provider. The case study enables an empirical investigation of the phenomenon of DT in projects by exploring the details in practice. The chosen mobility provider is a prime candidate for such a study as it carries out a high number of DT projects while also having a long-established EAM function.

The case study investigates the enterprise's current set up in terms of projects, organization and technology. The focus of the study lies on investigating changed and new requirements for EAM support of DT projects.

In the research phase, we collected and analyzed data on the current project-related, organizational and technical set-up of the EAM at the mobility provider and its changed and new requirements, as well as on new challenges and questions arising from the ongoing DT projects for EAM. More data was acquired through expert interviews [10, 15]. In addition to the interviews, we performed a document analysis according to Bowen [4] (see Sect. 3.4). The data obtained with the help of the interviews and the document analysis was evaluated by applying qualitative content analysis according to Mayring [14] in order to develop a description of the challenges of EAM in the DT.

3.2 Context of the Case Study

In order to clarify the essential role of EAM in DT projects, the following case study investigates the changed and new requirements for EAM support in large enterprises in the context of DT projects.

For this purpose, we interviewed EA experts involved in DT projects, as well as EA and EAM specialists. The obtained information formed the basis for determining which requirements had changed with regard to EAM. We used additional information from in-house documents to obtain a better overall picture. We then classified the EAM support tasks in DT projects based on the respective requirements of EAM. These requirements have been identified as critical elements representing challenges for the management support of DT projects. The objective was to understand the views of the different stakeholders of DT and the related requirements of EAM support in a specific context.

Below, we describe the proposed approach to clarifying the changed and new requirements for EAM support and demonstrate its application.

3.3 Data Gathering: Qualitative Expert Interviews

We conducted 10 expert interviews with staff in different positions, in order to identify changed requirements, new challenges and the questions regarding pertinent to the further development of EAM. The qualitative expert interviews were developed, planned, and conducted according to Kaiser's [10] data collection method. The interview guidelines were developed for experts from three DT project areas: research projects, enterprise architecture, and EAM strategy. Furthermore, we adapted them to the individual interviewees. The interview guidelines were written as complete scripts for structured interviews [15]. Further additional questions were asked during the interview.

The scripts for the interview guidelines each differ in structure, number of questions, and question settings. The duration of an interview was planned for 60 min, with 30 questions for the project-level experts and 33 questions for the EA business unit experts and EAM strategy experts. The guidelines comprised types of questions such as introductory questions, direct questions, indirect questions, and specifying questions.

Experts were selected for voluntary interviews based on their position, status, and experience. Further criteria for selecting the experts were knowledge of relevant functions, as well as the ability to provide accurate information and their availability to be interviewed [10]. The expert interviews were conducted respecting the principles of personal data protection, informed consent, anonymization, integrity and objectivity. The interviews were carried out continuously once or twice a month from the first quarter

to the second quarter of 2021. The interviews were completely transcribed saved in text form and added to the MaxQDA analysis tool database. We anonymized directly identifying characteristics such as names of the interviewees for further analysis or replaced them with function and position designations. For pseudonymization, the interviewees have been assigned consecutive alphanumerical codes.

In order to obtain the relevant information and access the functional knowledge of the experts in a targeted manner, each expert was assigned to one of three working areas according to their expertise, DT projects, enterprise architecture and EAM strategy. Some 40% of the experts work with EA in the area of DT projects, a further 40% are concerned with EA in their respective business units, and the remaining 20% of respondents are managers of EAM strategy.

The experts interviewed have different positions and functions with regard to enterprise architecture. The experience of the interviewed experts concerned with architecture management equates to several years of professional activity in their enterprise. Table 1 lists the expert positions by alphanumeric code.

Table 1. Positions of the experts in the enterprise.

Expert	Position	Working area	Functional knowledge
E1	Technical architect	DT project	EA technical and functional
E2	Architect IT service management	EA	EA service
E3	Product owner	DT project	Data & Artificial intelligence (AI)
E4	Technical architect	DT project	EA technical
E5	Head of data intelligence center	EA	Strategy & Organization
E6	Chief digitalist	DT project	DT research
E7	Head of IT architecture management	EA	EA mobility
E8	EA consultant	EA	EA consulting
E9	Head of enterprise architecture management	EAM	Strategy & Synergy
E10	Member of the chief technology office	EAM	Excellence & Technology

The experts working in the areas of project-related functional architecture, data exchange, IT architecture management for mobility, and research projects have shown particular interest in the results of this case study related to new EA metamodels, new data models, automation of data, new practical methods and processes for EAM, and clarity of relevant information.

3.4 Data Analysis: Document Analysis and Qualitative Content Analysis of the Expert Interviews

The general and basic information about EAM, the EA model and the central EA tool were retrieved from the in-house documents such as architecture and service principles, the central EAM tool and corporate intranet sites according to the document analysis [4]. We derived relevant information from the expert interviews according to qualitative content analysis [14]. From this set of relevant information, we derived findings and results in the form of key statements and classified them according to their significance for enterprise architecture. The classification was based on transcripts and rated as positive, neutral or critical. The key statements were compiled by area and ranked according to absolute frequencies in which they occurred in the data.

Summaries of the selected key statements are categorized under EA layers and EAM activities. Different characteristics were observed in the key statements. Through these observations, we were able to identify key themes that came up over and over again. These key themes made it possible to summarize the most important statements. Interestingly, we did not encounter major contradictions in the statements and the concerns were mainly pointing into the same direction. We employed a systematic procedure to analyze the key statements according to the information from the interviews pertaining to source, field, description, reason, possible solution and assessment. In addition, the selected key statements were verified through the EAM expert. Verification of correct recording was done through meetings with a member of the company's chief technology office. These key statements represent changed and new requirements, new challenges or new questions relevant to EAM. In this paper, we focus on the requirements which have changed or newly emerged in order to better adapt EAM to DT projects. In the result section, these changed and new requirements are structured by EA layers and EAM activities.

4 Results

We present the key findings from our case study in the form of 39 changed and new requirements which enable EAM to better support DT projects and the further development of EAM. Table 2 provides an overview of these requirements.

The case study makes it clear that EAM has encountered an increasing number of challenges as DT has progressed in the focal organization. Additionally, there is evidence that more enhanced EAM support is required as the number of projects grows. Several of the interviewed experts mentioned similar requirements for EAM and DT which like automation and standardization in all cross-functional fields. They pointed to the need for investment in automating the EA documentation, as well as the problems related to the data inventory, IT information flows, obsolete IT landscapes and the EA model. Several experts called for more use of standardization of IT application solutions and software. Standard solutions, e.g., from cloud providers, can offer better opportunities to integration solutions and services. In the following, we focus on changing and new requirements in terms of EAM support for DT projects.

We structured our findings according to the following categories of EA layers and activities for EAM: business, process, data, IT application, IT infrastructure and cloud

services, OT infrastructure, relation to DT projects, modelling and IT tool support, data automation, IT know-how and human resources, security and strategic application of AI/API. These categories were clustered based on our data analysis and show the composite requirements for each work area from the variety of tasks for EAM support. Each of these categories contain several changed and new requirements, which we describe in more detail below. The most significant changes are related to the areas of IT infrastructure and cloud services, IT application, data automation and DT projects.

In the category (1) **business** layer, one expert pointed out that views of services developed by competing DT projects do not show what services they offer due to reusability. For these reasons, there was a need to ensure the view of the services provided by the other DT projects through the up-to date EA model.

In the (2) **data** layer category, the adaptation of the data architecture to future IT processes and applications was requested by experts. They expect massive changes in the data processing due to increased data volumes (big data) and the use of AI.

The category (3) **IT application** layer includes six changed and new requirements. The increased number of distributed systems as a result of DT projects was mentioned by the experts as a new challenge for EA models. In addition, they expressed the need for an approach to the trouble that occurs in a highly distributed system when inventory components fail. The interviewees want to estimate the impact on the business process and the consequences of the additional dependencies for such a failure. The increased complexity requires new concepts for EA modelling. In addition, the experts stated that they seek to support and simplify complex dependencies of distributed systems as well as the dependencies in the IT architecture. New digital technologies/IT and new OT require an adaptation of the existing application architecture which should be monitored by EAM for better supporting DT projects. In line with new requirements, the speed of renovating existing applications should be increased with the help of cloud-native technologies. Heterogeneity in the implementation of DT solutions in business units should be reduced and should not lead to increasing time and costs in DT.

The category (4) **IT infrastructure and cloud services** layer contains ten changed as well as new requirements. It is of high importance for providing good support to DT projects and has to be adopted to meet requirements of future developments.

According to the experts, controlling the bandwidth capacity required for connecting the technical equipment can profit from EAM models and AI. For meeting future demands, the bandwidth should be adapted to ensure sufficient capacities to connect workshops and vehicles to the internet. With more sensors being installed on vehicles, sufficient bandwidth for directing data from the train to the operations control center will become even more important. As more sensors will be available in the future, the technical infrastructure architecture will be transformed into sensor-based architecture. This will demand more expertise in systems engineering from enterprise architects.

The DT of technical equipment and vehicles is currently hampered by bandwidth constraints in connecting workshops or vehicles to the internet. This can be explained by the increasing DT activities and projects related to workshops and vehicles and the growing number of sensors and technologies on vehicles. The current developments of DT in vehicle IT such as wireless communication of vehicles for customer services and

updates can influence the connection of vehicles to the internet even more. The shift of communication to the internet will largely continue at railroads.

Similar to IT application, new digital technologies/IT and new OT require adaptation of the technical architecture and monitoring by EAM to better support DT projects. In addition, the standardization of proprietary IT architectures and IT elements is required by experts in DT projects.

The areas of IT application and IT architecture are strongly influenced by the connection of IT with OT. The functional role between IT and OT has shifted. IT has now become more significant. As a result, EAM is increasingly active regarding the use of OT and OT-related tasks. These tasks can only be managed if the connection between the fields of electrical engineering and IT is considered. The new challenges in system engineering arose for technical architects during their participation in DT projects. This has lent increasing importance to the role of technical infrastructure architects. The requirements for IT expertise and electrical engineering are increasing with advancing DT. Additionally, a need to improve the implementation of cloud-native technology for versioning and fast deployment of software releases into production was identified.

At the cloud services layer, the interviewees mentioned requirements for data protection-compliant transfer of security-relevant applications to the cloud as well as a reduction of constraints in the network architecture for data transfer to the cloud. For data protection reasons, not all IT applications could be migrated to the cloud. Therefore, data-protection-compliant solutions for the transfer of IT applications have to be applied. In addition, they identified a need to optimize interoperability of the internal use of services in the cloud for the development and use of competitive enterprise's own applications and services. The case study shows that cloud products from various cloud solution providers (CSPs) are heavily used by the enterprise. However, the use of cloud services among deployed CSPs is inconsistent. The resulting dependencies on CSPs should be taken into account in order to securely build and operate specific applications and services.

At the same time, the experts saw a need to reuse services in many places for implementation in DT projects. In a large DT project, the EA specified a need for commitments to EAM to comply with certain technical IT architectures and avoid possible reprioritization and budget constraints in the future implementation of DT.

Complementary, a reduction in the heterogeneity of technological decisions in the IT landscape is required and should be controlled by a good EA model. This requirement can be justified with the better connectivity for DT projects. More investment is needed to modernize obsolete IT landscapes and make DT more efficient. Fulfilling these requirements increases the transparency of the networked system landscape.

In the category (5) **OT infrastructure** layer three requirements are mentioned by the interviewees: provide asset management service for OT through EAM with compatible IT and compliant security, accelerate IT processes over EAM to better support of OT and consider OT in EA models. Providing asset management services for OT through EAM with compatible IT and compliant security is becoming increasingly important as DT progresses. Heterogeneous and slow IT processes are stimulated to constantly improve in their homogeneity and their duration as a result of DT. Faster IT processes are needed to better support DT projects and the implementation of DT solutions. Innovative and

faster IT processes enable the EAM to better manage OT and the data architecture, as well as to reduce the increasing complexity of the IT landscape. The distinction between internal and cross-enterprise processes [19] allows process flows to be presented in a more transparent and controllable layout. This increases the visibility of both internal and external drivers of IT processes for efficiency and quality. In addition, OT should be increasingly considered by EAM in EA models due to DT progress.

In the category (6) **relation to DT projects**, the interviewees highlighted the need of improving the collaboration between EAM and DT projects. They demanded an improvement of functional communication with DT projects and an earlier EAM support for DT projects. Furthermore, DT requires a good collaboration between projects to ensure a better synchronization. This requirement was particularly raised by the EAM for communicating the availability of interfaces, for introducing a new technology, for informing of the deactivation of an application, and for monitoring dependencies. The interviewees also highlighted the need to ensure up-to-date information on the use of new digital technologies in concurrent and planned DT projects. In addition, it is necessary to improve the transparency of IT information flows and the documentation of services developed in DT projects.

In the category (7) **modelling and EA tool support**, then interviewees mentioned requirements such as establishing a metadata-oriented method of working, higher demand for well-structured EA to-be plans. Due to the increasing size of the EA database, which includes documents, databases or files, metadata-oriented working methods will be needed in the future. Through a standardized and tool-supported EA model, the higher demand for well-structured EA to-be plans can be met, and EAM can deal with the OT in a more targeted way. However, a continuous objective of integrating high-level EA models and data models for EA model stakeholders is not recommended [5]. In addition, due to better knowledge of the impact and context of the internal enterprise architecture, DT project leads primarily prefer internal rather than external EA consultants to assist in project work. In terms of EA tools support for DT projects, interviewees mentioned that appropriate EA tools should be offered for DT projects as simple or complex versions depending on the experience level of the user and the frequency of use. DT project work requires greater flexibility in the use of simple access to EA models as well as cross-team collaboration which takes the interests of the project participants into account. Regarding improved use of EA tools, experts mentioned characteristics such as simple and self-explanatory operation, good quality of data and user-friendly user experience (UX). These are especially important for non-expert users.

The category (8) **data automation** includes four requirements, which are responsible for using of an automated data processing system for filling and capturing EA documentation, increase the size of the EA database due to automatic data processing, ensuring of access to automatically generated IT information through EAM and support the development of the EA model towards to-be architectures through automated recommendations. An automated central data processing system should create, store and capture the EA documentation. Automated capture and assurance of accessibility of IT information through the central data processing system should be provided and ensured by the EAM. The requirement to support the development of the EAM model with data will become even more important in the future due to the increasing importance of data

for organizations [6]. Implementing these requirements based on targeted automated data processing can reduce significant risks and laborious, inaccurate manual data entry activities. In addition, it is also advisable to automate the interfaces for cloud services, for assets and for configuration item management.

In the category (9) **IT know-how and human resources**, the experts mentioned the requirements of transferring IT know-how from EAM to business departments engaged in DT projects and preparing for a higher demand for employees with IT & EA expertise for DT. To fulfill the changing requirements, IT knowledge needs to be transferred to specialist business departments, and more IT staff has to be employed in the departments. The development of competitive IT products and qualitative IT services as well as the implementation of DT in the enterprise could only be achieved through successful interaction in transformed teams with a high level of IT expertise and sufficient personnel. The challenge of EAM is to provide each department with IT competencies needed for DT. In addition, more investments are needed in personnel deployment and development.

In the category (10) **security**, the experts reported that the DT of the safety-critical IT systems in vehicles and infrastructure is associated with high costs, risks and amounts of time. The experts thought that this should be avoided through a well-designed to-be EA. In addition, the interviewees saw a requirement that EAM must ensure timely approval of safety-critical processes and systems in the technical infrastructure. The relevance of such processes and systems is of the highest priority for the maintenance of rail operations and is important for the successful implementation of DT at both the technical and organizational levels. Such systems should be considered a component of any technological development and innovation by DT. With the increasing DT of vehicles and further innovations in the technical infrastructure, safety-critical systems will be correspondingly more involved. EAM is required to respond to an event within a certain time. The dependence on sensor technology will require the design of reasonable and lean system architectures while meeting the specified safety requirements.

In the category (11) **strategic application of AI and APIs**, the experts mentioned the following requirements: strategic application of the use of AI and intelligent systems for supporting EA as well as of EA services and API services as a technical platform. The focus on the increased use of AI aims at making systems smarter, so that people can make better decisions due to advanced analyses or that these decisions can even be made automatically by the system. Due to technical innovations of digital technologies and increasing demands for moving trains with the same availability of the infrastructure, the requirements for the EAM strategy have changed. As a result, EAM should strategically consider intelligent solutions for a better utilization of the rail networks. The strategic application of AI and intelligent systems could be used in DT to solve complex problems in the future.

Table 2. Changed and new requirements for EAM support to DT projects

Categories EA layer	Requirements	
	State	Case study
(1) Business	Changed	Ensure up-to-date EA model regarding services developed by concurrent DT projects (e. g. for reusability)
(2) Data	New	Adapt the data architecture to future IT processes and applications
(3) IT applications	New	Consider the increasing use of distributed systems resulting from DT projects in EA models with the trouble of failure of existing components
	New	Support and simplify the complex dependencies of an EA increasingly based on distributed systems
	New	Adapt the existing application architecture to new IT and OT
	New	Renovate existing applications faster with cloud-native technology
	Changed	Reduce the IT landscape complexity
	Changed	Reduce heterogeneity in the implementation of digital technologies in business units
(4) IT infrastructure and cloud services	New	Control the bandwidth capacity with AI due to increased number of connected technical assets (workshops, vehicles)
	New	Adjust the bandwidth for connecting further workshops or vehicles to the internet in the future
	New	Adapt the technical architecture to new IT and OT
	Changed	Standardize proprietary IT architectures and IT elements
	Changed	Data protection-compliant transfer of security-relevant applications to the cloud
	New	Reduce constraints in the network architecture for meeting increased demand for data transfer to the cloud

(continued)

Table 2. (continued)

Categories EA layer	Requirements	
	State	Case study
	Changed	Optimize interoperability of services in the cloud for the development and use of the enterprise's own applications and services
	New	Ensure on-time delivery of components required for DT projects through binding commitments and contracts
	Changed	Reduce the heterogeneity of technology decisions in the IT landscape
	Changed	Increase investments for modernizing obsolete IT landscapes
(5) OT infrastructure	New	Provide asset management service for OT through EAM with compatible IT and compliant security
	New	Accelerate IT processes over EAM for better support of OT
	New	Consider OT in EA models
EAM activity		
(6) Relation to DT projects	New	Improve functional EAM communication with DT projects
	New	Earlier EAM support for DT projects
	Changed	More synchronization in collaboration with DT projects
	New	Ensure up-to-date information about the use of new digital technologies
	New	Improve transparency of IT information flows and documentation of services developed in DT projects
(7) Modelling and EA tool support	Changed	Establish metadata-oriented method of working

(continued)

Table 2. (continued)

Categories EA layer	Requirements	
	State	Case study
	Changed	Higher demand for well-structured EA to-be plans
	New	Offer appropriate EA tool support for EA experts and beginners to more flexibility and self-explanatory operation
(8) Data automation	Changed	Create an automated central data processing system for filling and capturing EA documentation
	New	Increase the size of the EA database due to automatic data processing
	New	Ensure access to automatedly generated EA information through EAM
	New	Support the development of EA models towards to-be architectures through automated recommendations
(9) IT know-how and human resources	New	Transfer of IT & EA know-how from EAM to the business departments engaged in DT projects
	Changed	Prepare for a higher demand of employees with IT & EA know-how for DT
(10) Security	Changed	Avoid higher costs and risks of safety-critical systems in vehicle IT for future DT projects through a well-designed to-be EA
	New	Timely approval of safety-critical processes and systems for technical infrastructure from EAM
(11) Strategic application of AI and APIs	New	Strategic application of AI and intelligent systems for supporting EA
	New	Strategic application of EA services and API services as a technical platform

5 Discussion

The goal of this case study was to identify the changing requirements for EAM in support of DT projects. By comparing the literature and the case study, we identified and described the differences in requirements for EAM in large enterprises between the literature and the case study. The differences between the literature and the case study can be summarized as follows: Only 6 requirements from the 39 requirements of EAM could be found in the literature. The results of previous research describe the requirements for EAM in government and the private sector. These requirements cannot be applied to EAM in large enterprises unverified, however, because they lack the scale and specifics of DT in large enterprises.

The literature already mentions requirements for EAM such as integration of different architectures [16, Table 1], management challenges [8, Table 3], and cooperation challenges in collaboration [8, Table 3]. This study case shows that the increased use of distributed systems should be controlled by the EAM through integration in an overall architecture. Literature suggests that Lean IT could be useful with bottom-up implementation for employees in a group and top-down leadership support for the changed and new requirements of EAM [12]. To design a transparent overall view of currently opaque IT information flows, EA communication needs to be supported via the central EA tool, which can provide a wide range of EA artifacts in a web interface. Such initiatives have already been tested [1].

The requirement for methods adaptation is already known in the literature [16]. In the case study, the focus of the requirement for modernization of an obsolete IT landscape is to increase the speed of renewal, reduce time to market, and strategically use cloud services in the IT landscape. The requirement for the standardization of interfaces and the application landscape has already been described in the literature [16, Table 1]. However, this requirement has only been addressed in the context of government and not to the specific standardization of IT architecture in large enterprises. To the contrary, the requirement for standardization of the many specific IT architectures with specific IT elements has been derived mainly to reduce the growing complexity in large enterprises. In the security area, the literature has already described compliance with confidentiality, integrity and availability for systems and infrastructure for compliant application [11]. In this context, the case study requires the timely regulatory approval of safety critical processes and systems in the technical architecture of large enterprises. Although this requirement from the literature relates to compliance with confidentiality, integrity and availability, it does not address the proactive and timely management of EAM required in a large enterprise.

EAM plays an essential role in organizations for connecting IT with business units and for the best possible implementation of corporate strategy. This has been evident in many enterprises for several years [19]. Based on its role, EAM is obliged to also support DT projects with these tasks. The changed and new requirements make clear that DT should be supported earlier by EAM and that more synchronization should take place in the collaboration between DT projects.

In summary, we learned that the areas of IT infrastructure, data automation, better IT application and the use of adjusted CSP cloud services contain the most requirements for EAM support. The automatic filling of the EAM model with data as well as data

processing in the cloud play an essential role in the further development of the EAM. Furthermore, attention to data protection is a critical factor in data distribution. Further changing and new requirements are placed on EA management for distributed systems and the handling of large amounts of data. Cross-functional systematic EAM collaboration and its long-term strategic deployment are critical for effective implementation of the identified changed and new requirements.

6 Conclusion

This study contributes to research by methodically identifying changing EAM requirements to further develop EAM support for DT projects in large enterprises. The findings improve the understanding of the changing and new requirements for EAM. The findings were empirically investigated and are based on a single case study at a large European mobility provider and are grounded on data stemming from qualitative expert interviews and documents. This paper provides results in the form of changed and new requirements, and also outlines strategies for responding to these changing and new requirements. The main tasks for better EAM support of DT projects are IT infrastructure, data automation, accelerated IT processes and adjusting how cloud services are used. In addition, other areas of EAM such as the EA model, IT landscape, OT infrastructure, IT application, IT know-how, IT tools, security and strategy are required for better EAM support of DT projects.

While our findings are based on a single case study, we assume that similar requirements might be found in other large enterprises [16]. However, due to its methodological limitations, our study does not claim to be applicable to all large enterprises. Subsequent case studies may reveal that other companies face similar or different changing and new requirements for EAM in its support for DT projects.

References

1. Aier, S., Fischer, C., Winter, R.: Construction and evaluation of a meta-model for enterprise architecture design principles. In: *Wirtschaftsinformatik Proceedings*, vol. 51, p. 643 (2011)
2. Burmeister, F., Drews, P., Schirmer, I.: An ecosystem architecture meta-model for supporting ultra-large scale digital transformations. In: *Twenty-Fifth Americas Conference on Information Systems*, Cancun, pp. 1–4, 4–8 (2019)
3. Burmeister, F., Drews, P., Schirmer, I.: Leveraging architectural thinking for large-scale E-government projects. In: *Fortieth International Conference on Information Systems*, Munich, pp. 1–14 (2019)
4. Bowen, G.A.: Document analysis as a qualitative research method. *Qual. Res. J.* **9**(2), 27–40 (2009)
5. Cammin, P., Heilig, L., Voß, S.: Assessing requirements for agile enterprise architecture management: a multiple-case study. In: *54th Hawaii International Conference on System Sciences*, Hawaii, p. 6012 (2021)
6. Chen, H., Chiang, R.H.L., Storey, V.C.: Business intelligence and analytics: from big data to big impact. *MIS Q.* **36**(4), 1165–1188 (2012). <https://doi.org/10.2307/41703503>
7. Gill, A.Q.: Adaptive enterprise architecture driven agile development. In: *24th International Conference on Information Systems Development, ISD 2015, Harbin*, pp. 1–5 (2015)

8. Hafselde, K.H.J., Hussein, B., Rauzy, A.B.: An attempt to understand complexity in a government digital transformation project. *Int. J. Inf. Syst. Proj. Manag.* **9**(3), 71, 74, 81–82 (2021). Article 5. <https://aisel.aisnet.org/ijispm/vol9/iss3/5>
9. Kaidalova, J., Kurt, S., Ulf, S.: How digital transformation affects enterprise architecture management – a case study. *Int. J. Inf. Syst. Proj. Manag.* **6**(3), 6–7 (2018). Article 2. <https://aisel.aisnet.org/ijispm/vol6/iss3/2>
10. Kaiser, R.: *Qualitative Experteninterviews - Konzeptionelle Grundlagen und praktische Durchführung*. Springer, Wiesbaden (2014). <https://doi.org/10.1007/978-3-658-02479-6>
11. Kaisler, S.H., Armour, F., Valivullah, M.: Enterprise architecting: Critical problems. In: 38th Hawaii International Conference on System Sciences, Big Island, Hawaii, p. 9. IEEE Computer Society (2005)
12. Kobus, J., Westner, M., Strahinger, S.: Change management lessons learned for Lean IT implementations. *Int. J. Inf. Syst. Proj. Manag.* **5**(1), 48 (2017). Article 4. <https://aisel.aisnet.org/ijispm/vol5/iss1/4>
13. Legner, C., et al.: Digitalization: Opportunity and challenge for the business and information systems engineering community. *Bus. Inf. Syst. Eng.* **59**(4), 301–308 (2017). <https://doi.org/10.1007/s12599-017-0484-2>
14. Mayring, P.: Qualitative content analysis. Theoretical foundation, basic procedures and software solution. In: Social Science Open Access Repository SSOAR, pp. 39–43 (2014). <http://nbn-resolving.de/urn:nbn:de:0168-ssoar-395173>
15. Myers, M.D., Newman, M.: The qualitative interview in IS research: Examining the craft. *Inf. Organ.* **17**, 2–26 (2007)
16. Obermeier, M., Wolf, P., Krcmar, H.: Anforderungen an ein EAM-Konzept für die öffentliche Verwaltung in Deutschland – Eine Fallstudie. In: *Wirtschaftsinformatik Proceedings*, Leipzig, Germany, vol. 57, pp. 895, 896, 898, 901–903 (2013). <http://aisel.aisnet.org/wi2013/57>
17. Sandkuhl, K., Wißotzki, M., Schmidt, R., Zimmermann, A.: On the effect of digitalization of products and services on enterprise architectures. In: 28th International Conference on Information Systems Development, Toulon, pp. 1–2 (2019)
18. Simon, D., Fischbach, K., Schoder, D.: An exploration of enterprise architecture research. *Commun. Assoc. Inf. Syst.* **32**, 2, 23 (2013). Article 1. <http://aisel.aisnet.org/cais/vol32/iss1/1>
19. Schwarzer, B.: *Einführung in das Enterprise Architecture Management*, pp.16–19, 20–33. Books on Demand, Norderstedt (2009). <https://books.google.de/books>
20. The Open Group: *Open Group Standard TOGAF Version 9.1*, pp. 7–10. Published in the U.S. by The Open Group (2011)
21. Vial, G.: Understanding digital transformation: a review and a research agenda. *J. Strat. Inf. Syst.* **28**(2), 118–144, 3–4 (2019)



Interoperability of Digital Government Services: A Brazilian Reference Architecture Model to Promote Communication, Management, and Reuse of Solutions

Adriana Xisto¹, Felipe Sommer², Marcus Vinicius Costa³,
José Lutiano Costa da Silva⁴, Claudia Cappelli⁵, and Vanessa Nunes⁶(✉)

- ¹ PRODEPA - Information and Communication Technology Company of the State of Pará,
Belém, PA, Brazil
- ² CIASC - Center for Informatics and Automation of the State of Santa Catarina, Florianópolis,
SC, Brazil
felipeas@ciasc.sc.gov.br
- ³ SERPRO - Federal Data Processing Service, Brasília, DF, Brazil
marcus-vinicius.costa@serpro.gov.br
- ⁴ Cesar School, Recife, PE, Brazil
jlcs2@cesar.school
- ⁵ UERJ - State University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil
claudia.cappelli@gmail.com
- ⁶ UFF - Federal Fluminense University, Niterói, RJ, Brazil
vanunes@gmail.com

Abstract. Digital Transformation has changed strategies and presented challenges on how to plan and implement electronic services and solutions in Governments. In the Brazilian scenario, we face the existence of 27 Federative Units along with more than 5000 municipalities that need to comply with Federal Statements and want to establish mechanisms to collaborate with each other. Inside the Digital Transformation Group of the States and the Federal District (GTD.GOV) one concern is how to promote a common environment to organize the management and development of digital services that focus on organizing interoperability requirements and standards to serve as a common focal point for addressing and collaborating on this demand. Therefore, this paper presents the development steps and the proposal for a reference model that supports the implementation of an architecture for the development and management of digital government services that promotes interoperability between services. This is an aspect of great relevance within the Brazilian federative entities to unify discussion and a view of the service development efforts and reuse of solutions. A practical pilot test was carried out in the State of Pará that highlights the potential of the reference model.

Keywords: Integration · Interoperability · Enterprise architecture · Digital government

1 Introduction

Since 2001, the United Nations Department of Economic and Social Affairs (UN DESA) has published the United Nations E-Government Survey [1]. It measures e-government effectiveness in the delivery of public services and appoints where it can be further exploited and evolved. Data from the 2020 Survey has shown many more countries and municipalities are pursuing digital government strategies, even when considering developing countries. It has been considered a tool to improve service performance, enhance customer experience, streamline operations, and create new business models [2]. When we take Sustainable Development Goals (SDGs) 2030 as a global and common focus of action it is about bringing services and engagement opportunities directly to people, including in remote or underprivileged communities. It may play an important role in strengthening digital literacy (Goal 4), digital inclusion (Goals 5, 8, and 10), digital connectivity (Goal 9), and digital identity (Goal 16) [1].

Digital Transformation (DT) emerges to bring the planning of Information and Communication Technologies (ICT) solutions to a strategic level, by directing the definition of changes and evolutions in society using digital technologies [3]. Strategies to implement DT in the public sector face many challenges as to how to implement it consistently [2–5] considering government planning, execution, and decision-making in the short, medium, and long terms.

From the Brazilian perspective, for a long time, most initiatives to introduce technology in services provisioning were the responsibility of Ministries and entities of indirect administration. However, given the new demands from society, it seemed necessary to create councils or groups that would converge and align the Brazilian federative units in terms of a discussion about a new federal model of service provision. The Digital Transformation Group of the States and the Federal District (GTD.GOV), was created to discuss the DT challenges in its different contexts considering a country of continental dimensions and to support, align and accelerate the efforts of the 27 Brazilian federative units (26 States and the Federal District) and their municipalities, in the development of DT strategies both for the performance of the public administration and for better delivery of public services. The creation of the GTD.GOV and its perceived initial impacts have been studied [6] and it was possible to observe some initial positive impacts.

Within its mission to concatenate DT efforts through the 27 Brazilian federative units, one of the strategic pillars discusses the use of enterprise architecture (EA) as an instrument to propose conceptual and technical models to organize the technological and service architecture in Governments, meeting the business demands, with a focus on the digital transformation of services. The subgroup created to work with the EA pillar is responsible for the diagnosis, systematization, and the unified proposition of guidelines and models for the structuring and strengthening of EA under specific domains with a focus on digital transformation in States and District Governments, as well as monitoring and accompanying the development of capabilities of this Architecture. This is closely related to the concept of Inter-enterprise architecture (IEA) which discusses the use and management of models to optimize and align business processes, relationships, and IT across organizational borders [7].

The EA subgroup has been focusing its efforts on the proposition of a reference architecture to provide a digital platform where citizens can execute services. This goal

has a lot of challenges. One of them is how to conduct a discussion considering such several federative units (27), and numerous municipalities (more than 5,000), with different economic, political, cultural, and social realities. Another one, arising from this first, is that with this number of elements (states and municipalities) the number of systems and infrastructures involved grows exponentially.

Naturally, this work, as members of the EA subgroup, should begin with a survey of possible approaches for the implementation of an Architectural Model throughout Brazil, followed by an evaluation of related studies. Although, because of this scenario, the first movement, following Nonaka and Takeuchi's knowledge spiral model [8] has been towards promoting knowledge transfer, through the socialization, externalization, combination, and internalization of what each Government knows, uses, and does.

The first domain of discussion inside the EA subgroup is related to interoperability, one of the priorities in GTD.GOV. Promoting the integration and interoperability of public services is not a trivial activity, considering the wide variety of computerized systems that support the processes and services carried out by the different public entities, even considering rules and definitions at the Federal level. We understand interoperability as a characteristic that refers to the ability of different systems and organizations to work together (interoperate) to ensure that people, organizations, and computer systems interact to exchange information effectively and efficiently.

The interoperability of systems in Brazil is a great challenge. In fact, bringing together an entire system encompassing federal, state, and municipal entities requires some essential elements such as the will of public managers, legal support, and information management infrastructure. Efforts have been made, and today it is understood that the biggest issue is not the proper use of technologies in digitization, but the readiness of actors so that they understand what digitization involves, which is much greater than computerizing in-person routines. This growing demand is in line with the premise of computerizing all public services, which is one of the pillars of the DT movements in Governments around the world [2], including in Brazil.

Consequently, due to the increasingly intense use of digital services by society, the federative units realized that the form of implementation is not adequate for their needs, which made more evident the demand for integration and interoperation. Within this reality, Brazilian federative units realized there is not a unified model that establishes an infrastructure composed of processes and ICT to support the management and development of electronic services considering interoperability and integration aspects.

The aim of this paper is to present the development process and the preliminary results of the definition of the first version of a reference architecture model for interoperability between entities in the provision of electronic government services. This reference model was developed using enterprise architecture systematic methods and design framework and focused on the point of view of two federative units: the state of Santa Catarina and the state of Pará. Our goal is that, as the reference model starts being used and its use expands to other states, they can build a repository of information that favors interoperability in the exchange of experiences and solutions. The model allows interoperability in the state (between municipalities), between states, and between them and the federal government.

A demonstration of its use was carried out within the context of the State of Pará. It was considered positive for organizing and integrating isolated initiatives that had

already been undertaken and others that were in a planning phase with the purpose of developing interoperability capabilities for digital services in the State.

The paper is organized as follows: Sect. 2 presents the theoretical foundation for the development of this work; Sect. 2 also presents the proposed method that guided the development; Sect. 3 describes the reference model and the demonstration in the State of Pará. Section 4 concludes this paper and presents next steps.

2 Materials and Methods

This work was conducted under the Design Science Research (DSR) paradigm. DSR is a qualitative research approach in which the conduction of the design process, simultaneously generates knowledge about the method used to design an artifact and the design or the artifact itself [9].

Figure 1 presents the framework proposed by Hevner et al. [10] used as the basis to conduct this work as follows: (2.1) The Environment under which we applied our work and to whom it wants to provide results and benefits; (2.2) The Knowledge Base that relates to the definition and discussion of the fundamentals and methodologies adopted in the research; and (2.3) the IS Research Cycle, that is related to the process of build and evaluate an IT artifact. In the following subsection, we detail each of them.

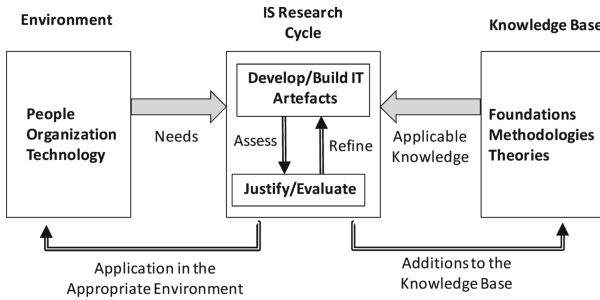


Fig. 1. Design science research framework for information systems based on [10]

2.1 Environment: DT and the Brazilian Government Perspective

According to the latest assessment carried out in 2020 on the development of e-government in 139 countries [1], the United Nations describes that, in general, governments around the world are using digital technologies to innovate and transform the way they operate, share information, make decisions, and deliver services, as well as engage and make partnerships to face challenges of public interest.

The concept of Digital Transformation arises in this context to link the adoption of digital technologies as an agent of transformation of the organizational strategic paradigm. Given this, managers of government agencies have been rethinking their strategies and procedures, since conception. The Brazilian Federal Government published Law No. 14,129 on March 29, 2021 (Digital Government Law) which deals with

the principles, rules, and instruments for Digital Government. The Digital Government Law aims to transform government into a digital service platform. Its content sets out some rules to improve the delivery of public services to society and it is valid at the three levels: federal, state and municipalities. The idea is that citizens can request services, documents, certificates, and other types of requirements directly by cell phone or computer, 24 h a day, seven days a week.

In this context, the Digital Transformation Group of the States and the Federal District, (GTD.GOV) emerges. It is a national network that brings together specialists in digital transformation from State and District Governments across the country. Considering its sponsors, CONSAD (National Council of State Secretaries of Administration) and ABEP-TIC (Brazilian Association of State Entities for Information and Communication Technology), GTD.GOV proposes actions with an emphasis on the mission to support and accelerate digital transformation in state governments.

Within this mission, the group is divided into three strategic pillars in the form of subgroups. The enterprise architecture subgroup works on the development of conceptual and technological frameworks for public service delivery platforms, intra-government integration, and the generation of information and strategic indicators for government decision-making. The authors of this paper are part of the Integration and Interoperability for Digital Transformation Workgroup and developed this project.

This subgroup provides the models and technical resources to enable digital transformation through the implementation of the concept of government as a platform, more “pluggable”, connected with other platforms and services from the government itself, from the industry, academia, the third sector and society in general. A government that is not only a service provider but also a provider of inputs, such as public data and public knowledge to stimulate open innovation in society in favor of a more open government and innovative public services co-created with citizens.

2.2 Knowledge Base

This work is based on the current two demands for Digital Transformation of Governments: The challenges of interoperability within this context and how the practice of Enterprise Architecture can be an instrument to organize and share solutions. We talk about these issues to understand the rationale and the proposal itself.

Interoperability. Interoperability is a much-discussed subject concerning the interface between data and systems. In the context of digital government, it is also a challenge, due to intra- and interstate public organizations still having a very independent and non-standardized operation in the management and development of digital services to the citizen. This is also one of the goals of the Digital Government Law.

In Brazil, over the last 15 years, several electronic government (e-Gov) initiatives have been implemented. One of the best knowns is the one focused on data and process integration, the Global Data Model – MGD, whose standard was incorporated into the standard called ePING Interoperability Architecture in 2011 proposed by Federal public organizations. Other initiatives followed to work on interoperability between information

and/or specific systems, but no initiative towards discussing a platform for the management and development of services considering interoperability in a systematic way has emerged since then.

Another important aspect when discussing challenges in the interoperability between information and systems is the understanding of the business that defines the meaning of things and, for that, between different organizations it is first necessary to work on the semantic interoperability that determines a language of relationship between concepts in a way to solve, for example, how to identify the same concepts, but with different names, and how to perceive different concepts with the same names, considering all the possible nuances between one and the other. The fact is that progress in providing solutions increasingly depends on the acquisition, processing and analysis of large volumes of data. The validity of the results and the security of the applications depend on an adequate understanding of the real-world semantics of this data and that are related to its interpretation and the context in which it is acquired and processed. This is challenging as interpretations vary, context is infinite and can change over time [12]. Therefore, understanding the semantics of things and the relationship between them in the real world becomes essential to design systems that will act according to the understanding and processing of the world.

Enterprise Architecture and FACIN. We can understand Enterprise Architecture as a well-defined practice that guides organizations to execute their strategy through analysis, planning, implementation, and changes in their processes, information, and technology, always using a holistic approach [13].

Enterprise architecture aims to represent organizations and their relations. It helps define and optimize processes and practices to help with business analysis, project design, and implementation. Its purpose is to optimize the processes, information, information systems, technologies, and even physical infrastructure of an organization, providing support for decision-making and alignment with the business strategy.

This theme arises in the context of governments around the world, due to the need to adopt strategies for the development of electronic government in an efficient and effective way. In this sense, the practice of enterprise architecture serves as an instrument to support the process of Digital Transformation that must occur in an organized and thought-out way within the context of the strategy of public entities in all their spheres and the relationship between them. The best-ranked countries in the 2020 biannual assessment carried out by the United Nations on e-government development in partner countries [1] have generally invested in the implementation of enterprise architecture practices and disseminated their standards. This is the case of Denmark, Korea, Estonia, Finland, Australia, United Kingdom, New Zealand, United States, Netherlands and Singapore.

In Brazil, discussions on the implementation of a framework aimed at the Brazilian context began in 2014 and in 2016 the proposal of the FACIN - Enterprise Architecture Framework for Interoperability in Support of Governance was launched to support the implementation of the Digital Governance Strategy as an instrument to promote collaboration between government organizations and transparency in communication and provision of electronic services [14]. FACIN has been used as a base framework for all actions within GTD.GOV's Enterprise Architecture subgroup.

Interoperability and Enterprise Architecture. Although Interoperability and EA have been taking an important role in e-Government interoperability development, and they have been discussed jointly, there are some challenges presented in [15] that we are planning to address either in the proposed reference model or considering the existing partnerships and collaboration protocols within GTD.GOV related to: Layers of interoperability; Coordination and organizational inflexibility; Lack of leadership and trust; Poor technical design; and Lack of practical guidelines.

Regarding the layers of interoperability, we defined that the reference model must address semantic interoperability to ensure unification of meaning and understanding among all federative units. Technical Interoperability is to be addressed as a second layer of the reference model to model each technical solution planned and implemented by each Federative Unit. Organizational Interoperability is both addressed by the reference model itself in relation to the use of known information architectures inside governmental and non-governmental databases and also by the method to instantiate the reference model that suggests the model and discussion about necessary transformations on business processes to adopt the new model.

Discussions about coordination and organizational inflexibility and lack of leadership and trust were also pointed out. GTD.GOV plays a special role since all Federative Units representatives established and have strengthened the commitment regarding projects, like this one, that are selected and voted to be developed.

Lastly, a possible impression of poor technical design and lack of practical guidelines was discussed from the beginning by addressing expectations and the strategy adopted. Aligned with that, the project team had to discuss how to propose a reference model broad enough to be recognized as useful by all 27 federative units but also could approach more detailed layers to provide technical examples, specific solutions, and common practices. All three challenges are concerns that will be continuously evolved after each evolution round of the reference model.

2.3 IS Research Cycle

The Architecture Development Method (ADM) described in the TOGAF 9.2 standard (The Open Group Architecture Framework) was used as a reference (<https://www.opengroup.org/togaf>) in this work. ADM is a generic method that defines a recommended sequence for the various phases involved in developing an enterprise architecture. The steps and deliverables based on the ADM are shown in Fig. 2.

Step 1 (Architecture View) aims to establish the work strategy, the execution plan and the strategic analysis of the problems and impacts of the current situation in relation to the demand. Regarding Strategy Statement, the following should be defined: the enterprise architecture team with contacts and responsibilities; the method, techniques and tools that will be adopted (considering as a basis the use of FACIN metamodel and the metamodel used to develop the reference model); and enterprise architecture principles that must be respected. With respect to the Statement of Work, the team must defined: the interested parties composed of people with a direct interest and power to take decisions or influence the project; the context that describes and justifies the demand for implementing the reference model; the scope and goals of the project (considering the

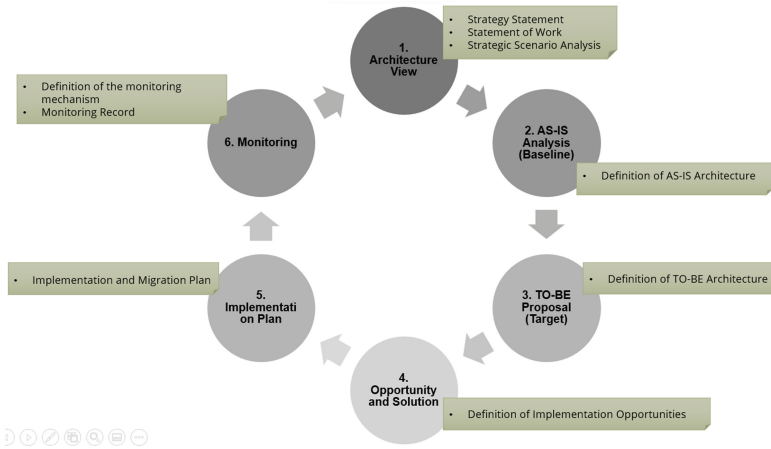


Fig. 2. The method

scope of the reference model that is planned to be implemented); a high-level description of the current architecture (and the strengths and weaknesses) and an overview of the target architecture (how stakeholders envision a new reality given that the goals have been achieved); an assessment of the readiness of development teams and stakeholders to carry out the transformation actions that will be planned; the identification of risks to the project and possible mitigation actions; the indicators that will be used to assess the quality of the project and the value it has delivered to the State; and a work execution schedule. Finally, regarding the Strategic Scenario Analysis, the following must be carried out: an assessment of the current situational context, characterizing and justifying the guiding/motivating elements for this work; the establishment of strategic goals in terms of the values that must be delivered to Society and what outcomes must be implemented to achieve these goals; and the definition of the macro-requirements that must be implemented by the proposal of a future architecture to deliver the desired values.

In step 2 (AS-IS Analysis), the current situation is modeled presenting how the organization works in relation to the demand. In step 3 (TO-BE proposal), the proposal for the future architecture is discussed and modeled, aligned with the reference model, which presents how the organization can function once the proposed transformation actions are implemented. In step 4 (Opportunity and Solution), opportunities, and solutions for implementing the future architecture are analyzed, worked out and decided. In step 5 (Implementation Plan), the implementation plan to develop the future architecture (or part of it) is defined, which is managed through a unified vision. Step 6 (Monitoring) includes monitoring the development, use and evolution of the implemented architectures of each state and the evolution of the reference model fueled by its use and evolution.

To model the diagrams, we used the ArchiMate enterprise architecture modeling language, which is an open standard. We used the free ArchiMate Tool (<https://www.archimatetool.com/>) to create the diagrams.

Before proceeding with the presentation of the reference model development process and the model itself, it is important to discuss the reason why it was worked only from the

perspective of 2 Federative Units. The reason starts with the discussion that evaluations in DSR usually check the artifact against the requirements, the development process, and the goals, and focus on evaluating individual users and their ability to learn and use the artifact. However, given the failure of many IT artifacts to fulfill their expected purposes when considering many people and even more when they are from different organizations and locations, according to [16] there is a need for improvement in the evaluation methods considering methods and theories from the social sciences. “The use of social theory in evaluation is predominantly to understand social behavior and culture, identifying issues that are not necessarily related to technical aspects of the systems, but may affect its use and acceptance” [16]. It combines with the way we choose to conduct the development and evolution of the reference model since users use existing knowledge to form their opinion of a product or service and from there, they develop new views as they acquire new knowledge, both from its use and from researching and benchmarking other similar approaches. We didn’t make use of a specific social sciences method, but we chose to work with 2 federative units instead of the existing 27 to focus and deepen the discussions about the real context and interactions. We used this reasoning since step 1, to understand the context of use prior to commencing with problem identification and from this understanding, develop an initial proposal for a reference model.

3 The Reference Model for Interoperability Between Entities in the Delivery of Electronic Government Services

To develop the reference model, we executed only steps 1, 2, and 3 to generate the target model (Sect. 3.1). Thenceforth, our goal was to perform a pilot of its usage. Therefore, for the State of Pará pilot, we executed steps 1, 2, 3, 4, and 5 (Sect. 3.2). Lastly, in step 6, we defined the implementation tracking mechanism for the continuous evolution of the model (Sect. 3.3).

3.1 The Reference Model

Steps 1, 2, and 3 are described below.

1. Architecture View. Regarding the Strategy Statement, it involved the definition of the working method itself (Sect. 2.3), the training strategy and the definition of architectural principles. With respect to the training strategy of the participants, the participant with high experience in ArchiMate modeling was responsible for the modeling and for transmitting this knowledge to the other participants.

Principles define aspects that must be respected during the construction, transformation, and operation of the organization, directing the target enterprise architecture and how it must be implemented. As a method to systematize the practice of enterprise architecture has not yet been established within the GTD, three principles were defined, which had a direct impact on the realization of this project. For each one, the name, an explanatory statement, the justification for its existence and the positive implications of its adoption were described. The principles are: Promoting Information Transparency;

Perform Integrations with Low Coupling; and Compliance with legal requirements of the public administration (federal and state).

Regarding the Statement of Work, we defined all the important items related to planning for the execution of this project like scope, goals, risks, transformation readiness and schedule. The scope of the work had as its motto the need to work on the lack of a unified systematic, inter- and intra-state, that promotes interoperability in the development and management of digital public services offered by the States.

In this sense, the goal was to build a Reference Model for interoperability between entities in the provision of government services that, as it is used and has its use expanded to other states, can represent a repository of information that favors communication and sharing of interoperability solutions for the development of government services. The model must be general enough to allow the discussion of interoperability solutions within the state, between states and between states and the federal government.

This work was initially carried out with two States to manage the complexity of working with the 27 Federative units at the same time: Santa Catarina and Pará. The analysis was carried out by two specialists who work directly with this topic in their respective States. In the current scenario, when the need for systems integration arises, professionals work together to decide complex issues such as which technology can be applied, which data dictionary is going to be selected and developed and which service limitations are, decisions that should already be standardized and that could allow professionals to focus only on the business rules and integration requirements.

A strength worth noticing is that large enterprise systems have already developed an integration layer to solve urgent problems and meet government-critical business rules. Basically, each system manager developed integration services to meet the most requested demands, but without completely planning the reuse of these services.

The challenges in the current environment include: (i) Lack of alignment and general guidance on the most suitable technologies to carry out integration and interoperability between systems; (ii) Lack of a standard semantics to facilitate interoperability within and outside the State; (iii) Unavailability of integration services, as well as their documentation; (iv) The State's technical teams often lack the knowledge and training to properly choose and define the technologies to be used in integrations and interoperations; and (v) There is no minimum level of governance regarding interoperability within the scope of the state public administration.

The expectation is that an architecture that contemplates the concept of interoperability will bring more dynamics and speed to the digital transformation movement, specifically to the management of the development and offering of public services, easier integration with other public or private entities and more cost savings of the implementation of integrations between systems.

Therefore, in the desired future environment, each state will be able to have its own interoperability architecture based on the proposed reference architecture so that all are recognized and communicated, providing a high degree of interoperability and collaboration and, with that, contributing to: Standardization of concepts, formats and data delivery; Optimization of communication between the entities involved with the actions of the State and with the delivery of services to the citizen; Improvement in the management of processes, solving bottlenecks that occur due to lack or delay in sharing information; Transfer of information between two or more systems without a

technological dependency between them; Unified access to services that share information that is scattered across different systems; and More complete information, favoring decision-making.

Eleven factors were defined to assess the degree of readiness to carry out an implementation of the reference model to be designed considering the initial architecture vision that was discussed. On average, the level of readiness was considered from acceptable to high with easy to moderate levels of difficulty to deal with challenges that may arise. This work helped to identify seven risks of the reference model implementation project that each state may face.

Two indicators were also defined to monitor the acceptance and use of the model by the States. A work schedule was also defined and carried out smoothly in 5 months.

In the last deliverable, the Strategic Scenario Analysis, we worked in greater depth on the analysis of the current scenario in terms of motivation and strategy. Diagrams were created defining the expected goals and the expected operational outcomes and their impact on the goals. Thenceforth, the requirements were defined to specify what needed to be accomplished to deliver the expected results. Figure 3 presents a partial view of this diagram. Semantically it reads: A requirement, once implemented and deployed, accomplishes the delivery of (part of) one or more outcomes.

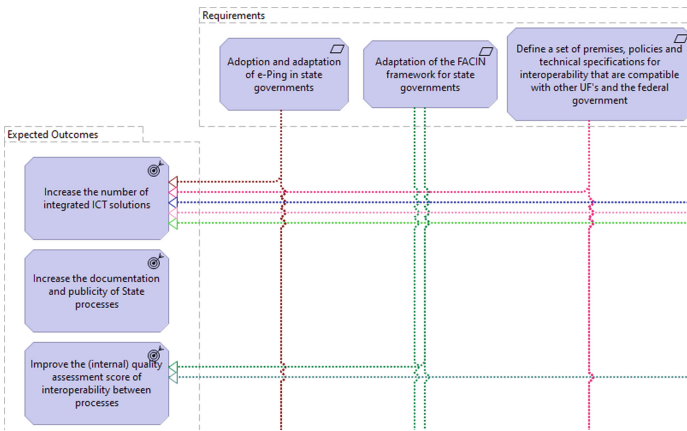


Fig. 3. Diagram of requirements x expected outcomes

2. AS-IS Analysis (Baseline). During the performance of step 2, we chose to carry out a textual description and not through the development of diagrams, to optimize time, since the current situation was fully understood and inside the domain of the specialists of each State. Thus, the deliverable “Strategic Scenario Analysis” (from step 1) contains explanatory and more detailed sections on the reality of each of the States separately.

The analysis of the two states brought up the need for a paradigm shift in urgent care to provide digital services to citizens versus the importance of thinking about mechanisms that unify and integrate the management and the development of digital services. The main problems reported relate to: the lack of an appropriate place to consult existing

integration services, as well as easy access to documentation; lack of standardized definitions to establish security in transactions; decision-making depends on a series of tools to generate dashboards; and there is no standard for integration, documentation, security and development of digital services, which generates a high effort in the planning and implementation of integrations.

3. TO-BE Proposal (Target). The Reference Model for interoperability between entities in the provision of government services is presented in Fig. 4. The elements in yellow color are related to business domain concepts such as: Business Role, Business Interface, and Contract. The elements in blue color are related to application and data domain concepts such as: Application Service, Application Interface, Application and Data Object.

The Reference Model interfaces with access channels to citizen services and with integrating entities that, through the establishment of contracts, consume and provide integration services. The Reference Model also promotes integration with the access authentication mechanism of the GOV.BR platform that is responsible for the unification of identification mechanisms for citizens in Brazil to access federal, state and municipal digital services.

The main components of the Reference Model are:

- Elements that can compose a Service Manager and that should be concerned with offering a Public Integration Portal, with free access, that organizes and presents the Services Catalog, How to access, Documentation and Terms of responsibility, and FAQ. There must be Platform Management mechanisms for various queries, publication of new services, content management and profile-based Access Management. These elements focus on managing organizational interoperability among public and private organizations. It is also important to interoperate the Semantics of related concepts, fields, and any element used to understand and interoperate digital services. For the public, it must offer mechanisms for the evaluation of services.
- The Identity Manager should be concerned with ensuring safe access to services and it is suggested States to continue the maintenance of a State identification Base but also that it is integrated into the Federal Government Digital Platform (GOV.BR). It allows the State to have a citizen database and to manage its access, by client applications, to integration services. The following requirements were established: User identification through the State's own database; Identification via digital certificate; Identification via GOV.BR login provided by the Federal Government; Fast authentication through Single Sign On (SSO) technology and OAuth2 protocol; Enable multi-factor and adaptive authentication;
- The Integration Manager recommends mechanisms to: monitor and protect access to technology infrastructure to control the amount and frequency of access to integration services, respecting the technology infrastructure capacity of the service provider agency; track the access and data manipulation (LGPD) that aims to the quick identification of integration traffic, making it possible to identify who accessed it, what data they accessed and when they accessed it; and distribute requests transparently, through the Service Bus (ESB). The ESB component receives a request and redirects

it to the appropriate path, where the request will be processed. Consumer applications only have contact with the interface that was published.

- The Integration Services component are the routines built to make data available according to a business rule. They can be called services that provide information that will be requested by APIs services, according to the service catalog rules. For the Communication Interface, it is recommended to use SOAP (Simple Object Access Protocol) and REST (Representational State Transfer). For Message Format, it was recommended to use JSON (JavaScript Object Notation) and XML (Extensible Markup Language). These technologies were based on the realities of the 2 States that participated in this project, and they will evolve in the next evolution rounds with other States.
- Gov application and Gov database are State governmental components which Integration Services communicates with.
- Outside of the State Reference Model (in grey) there are different channels that citizens use to access digital services and other Gov and Non-Gov Applications and Databases that are to be integrated with the State.

All the details of the Project can be accessed (in Portuguese) at <https://gtdgov.org.br/uploads/publications/U1hujhLBNCfGLaeY5IJXv9t9tJDT5ynOGAGiJcE8.pdf>.

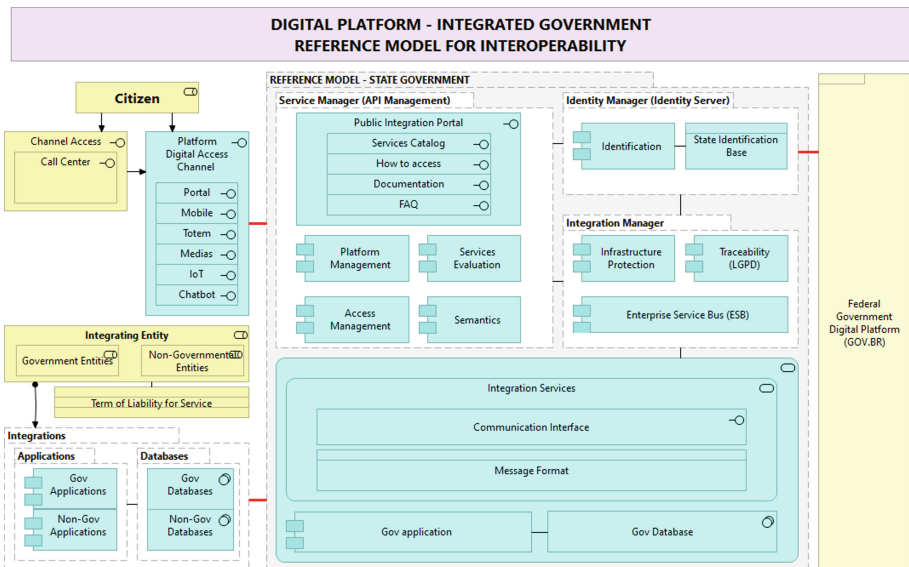


Fig. 4. Reference model for interoperability

3.2 A Proposal for the Model Implementation

The state of Pará was the scenario, through PRODEPA, which is the public ICT (Information and Communication Technologies) Company responsible for processing data in

the state. They are already involved in an initiative to implement an infrastructure for the interoperability of services in the state and they are developing some of the elements of the Reference Model.

The State specialist carried out the modeling from the perspective of the integration processes considering an existing service and considering the development of a new one. She followed steps 1, 2 and 3 of the method, instantiating the model for the State.

In the analysis of the current scenario, she modeled the process for developing integration of the existing services and for developing a new service. In the Target Architecture, considering an implementation adherent to the reference model, it was possible to elaborate a process for the development and sharing of the services. The proposal is that Managers will have access to the new platform (to be developed) to provide information about the services and configure the levels of access. In this way, a client entity has access to information via a Public Portal of Integration Services.

A strategic impact analysis and the benefits of using the reference model were carried out in relation to the official indicators and goals of the State Multiannual Plan (PPA 2020–2023) related to digital transformation, which are: Time savings and acceleration in the exchange of information given the ease of access to service documentation, available data set and communication interfaces; Debureaucratization in providing access to private services; Development of innovative solutions making use of different integration services, distributed in different systems; Automating data sharing, and reducing manual work, which requires effort from technical teams; and Greater agility in the delivery of digital services to the citizen.

In addition, value deliveries for PRODEPA's internal outcomes were perceived: Expanding the Services Portfolio; Increase the Number of Integrated ICT Solutions; and to be a motivating and facilitating agent for the digital transformation of the State.

The implementation and migration plan were designed to coordinate a series of isolated efforts that were already being carried out, and through this initiative, to unify and consolidate efforts. To this end, transition architectures were proposed that present the deliveries in four major milestones as shown in Fig. 5.

3.3 Monitoring the Use and Evolving the Reference Model

Two monitoring processes were defined: a Local level and a GTD level. The first deals with an internal monitoring where those responsible for the Federative Unit must monitor the implementation of transformation actions aiming at: providing information on the modeling, planning and decisions taken at project time and at future model evolutions; and, in the case of changes, supporting discussions and providing information when requested about the update of models as a result of deviations made during implementation.

Also, inside the GTD enterprise architecture sub-group, a specific team is responsible for monitoring the use of the Reference Model by the States to collect inputs for continuous evaluation and evolution. In this sense, all Federative Units must inform GTD about the use of the reference model and request a consultative follow-up.

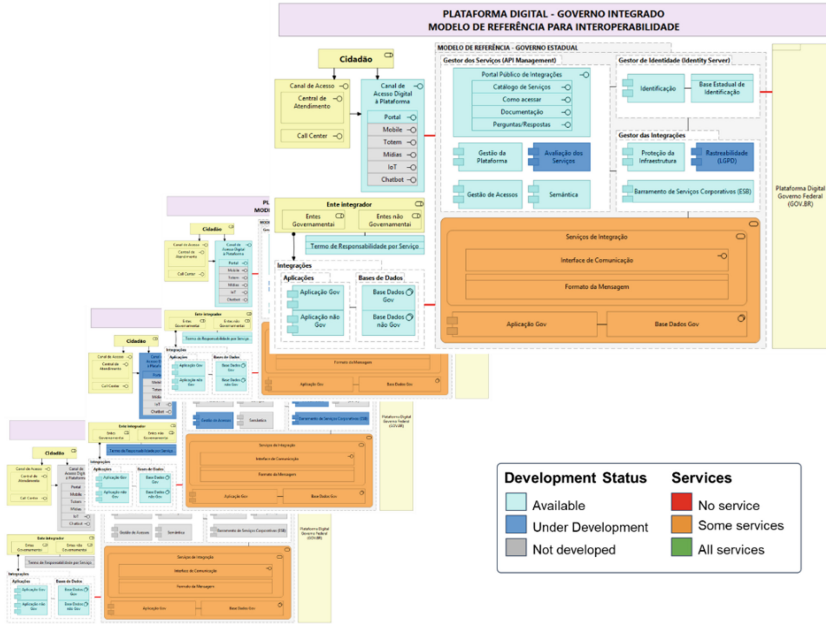


Fig. 5. Transitions architectures – state of Pará

4 Conclusions

The management and governance of ICT solutions within the government is key to the success of the design and implementation of digital government services in a unified, scalable, and quality manner. The creation of a Digital Transformation Group whose objective is to promote the meeting of Federative Units and to serve as a connecting link with the federal government in establishing paths for the adoption of federal determinations and solutions has shown to be a promising path.

In this sense, the proposal of a project to work with a high priority demand, which has already been discussed both at the federal and state levels, was the problem of the lack of an approach that deals with the management, development and offering of services with a focus on an element that is dear to states today: The lack of clear definitions to approach interoperability through all services. The use of a mechanism proposed by the Federal Government to establish a common modeling language and methodology (FACIN) for the design of solutions only strengthens the idea that the Federative Units can continue to develop their solution platforms considering their specificities, but within a common architecture that serves as an instrument to strengthen the communication bridge between States and the Federal District in the understanding and exchange of contexts, problems, experiences and solutions.

In this context, this project resulted in an immediate contribution to the two participating States. The State of Pará, in the developing a series of initiatives for this purpose, used the pilot to organize their isolated initiatives and to present the value of them in an integrated way, and as a tool to redefine strategies according to the relations

of contribution and dependence between initiatives. In the State of Santa Catarina, the initiative served as a way of making it clear to potential Sponsors the importance of an interoperability strategy and standards for the development of services.

This reference architecture was established from the experiences of the components of the EA subgroup and other collaborators consulted during the work, and finally, disseminated by CONSAD and ABEP among their affiliated institutions.

The reference architecture is not a static model and must be continually evolved according to new perceived needs. Likewise, the final document will serve as a basis for State and District Governments to build/develop/share/deploy their digital platforms, also allowing each Government to assess the evolution and degree of maturity of its platform. In addition, the reference architecture will enable EA subgroup to develop a model for monitoring and evaluating the development of Digital Government Platforms for the States and the Federal District.

One of the consequences of this monitoring is to reinforce the importance of this reference architecture as a basis for other works with a focus on Digital Government. One is to work a maturity ranking of government services, and the development of components collaboratively between governments in different spheres and powers.

Future work involves 3 fronts: Support for the use of the reference model in the State of Pará; Support for the establishment of an integrated project that supports the actions under discussion by the State of Santa Catarina; and the selection of two States that have a high level of readiness and availability to work on a second round of the implementation and evolution of the reference model.

Acknowledgements. We are grateful for the support of GTD.GOV represented by its National Coordinator Vânia de Carvalho Marçal Bareicha. We are also grateful for the financial support provided by the State University of Rio de Janeiro (UERJ).

References

1. United Nations: 2020 Digital Government in the Decade of Action for Sustainable Development. United Nations, New York (2020)
2. Gong, Y., Yang, J., Shi, X.: Towards a comprehensive understanding of digital transformation in government: analysis of flexibility and enterprise architecture. *Gov. Inf. Q.* **37**, 101487 (2020)
3. Reis, L.C.D., Bernardini, F.C., Ferreira, S.B.L., Cappelli, C.: An ICT governance analysis for the digital and smart transformation of Brazilian municipalities. In: The 22nd Annual International Conference on Digital Government Research (DG.O 2021), Omaha, USA (2021)
4. Mergel, I., Edelmann, N., Haug, N.: Defining digital transformation: results from expert interviews. *Gov. Inf. Q.* **36**(4), 101385 (2019)
5. Pittaway, J.J., Montazemi, A.R.: Know-how to lead digital transformation: the case of local governments. *Gov. Inf. Q.* **37**(4), 101474 (2020)
6. Rodrigues, G.A., Avila, T.J.T., Lanza, B.B.B.: Impacts of an articulation group for the development of the Digital Government in the Brazilian Subnational Government. In: The 22nd Annual International Conference on Digital Government Research (DG.O 2021), New York, NY, USA, pp. 339–350 (2021)

7. Ehrensperger, R., Sauerwein, C., Breu, R.: Current practices in the usage of inter-enterprise architecture models for the management of business ecosystems. In: IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC) (2020)
8. Nonaka, I., Takeuchi, H.: *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, Oxford (1995)
9. Carstensen, A.-K., Bernhard, J.: Design science research – a powerful tool for improving methods in engineering education research. *Eur. J. Eng. Educ.* **44**(1–2), 85–102 (2019)
10. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* **28**(1), 75–105 (2004)
11. Weber, S.: Comparing key characteristics of design science research as an approach and paradigm. In: *Proceedings of the PACIS 2012*, p. 180 (2012)
12. Davies, J., Welch, J., Milward, D., Harris, S.: A formal, scalable approach to semantic interoperability. *Sci. Comput. Program.* **192**, 102426 (2020)
13. FEAPO: A Common Perspective on Enterprise Architecture (2018). <https://feapo.org/wp-content/uploads/2018/10/Common-Perspectives-on-Enterprise-Architecture-Final-1-copy.pdf>
14. Nunes, V., Cappelli, C., Costa, M.: FACIN: the Brazilian government enterprise architecture framework. In: *Proceedings of the 19th International Conference on Enterprise Information Systems, ICEIS*, vol. 3, pp. 433–439 (2017)
15. Saekow, A., Boonmee, C.: Bridging the gaps in e-government interoperability implementation: towards a realistic approach. In: *Proceedings of the 3rd International Conference on Information Sciences and Interaction Sciences (ICIS)*, pp. 265–273 (2010)
16. Lawrence, C., Tuunanen, T., Myers, M.D.: Extending design science research methodology for a multicultural world. In: Pries-Heje, J., Venable, J., Bunker, D., Russo, N.L., DeGross, J.I. (eds.) *TDIT 2010. IAICT*, vol. 318, pp. 108–121. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12113-5_7

Business Process Modeling and Monitoring



Modeling, Executing and Monitoring IoT-Driven Business Rules with BPMN and DMN: Current Support and Challenges

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

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

Abstract. The involvement of the Internet of Things (IoT) in Business Process Management (BPM) solutions is continuously increasing. While BPM enables the modeling, implementation, execution, monitoring, and analysis of business processes, IoT fosters the collection and exchange of data over the Internet. By enriching BPM solutions with real-world IoT data both process automation and process monitoring can be improved. Furthermore, IoT data can be utilized during process execution to realize IoT-driven business rules that consider the state of the physical environment. The aggregation of low-level IoT data into process-relevant, high-level IoT data is a paramount step towards IoT-driven business processes and business rules respectively. In this context, Business Process Modeling and Notation (BPMN) and Decision Model and Notation (DMN) provide support to model, execute, and monitor IoT-driven business rules, but some challenges remain. This paper derives the challenges that emerge when modeling, executing, and monitoring IoT-driven business rules using BPMN 2.0 and DMN standards.

Keywords: IoT · BPM · BPMN · DMN · Business rules · Challenges

1 Introduction

As electronic components have become smaller, less expensive, and more powerful, the Internet of Things (IoT) has received an upswing [3]. Many embedded components are equipped with software, sensors, actuators, and network connectivity that enable the collection and exchange of data (sensors) as well as physical responses to events (actuators) [2]. Such physical objects can be embedded in everyday devices such as smartphones, wearable devices, washing machines, or refrigerators. They can be further found in large systems such as, smart cities, logistics or healthcare [4]. In general IoT refers to a network of physical objects populated by sensors and actuators that communicate and exchange data over the Internet [5]. While sensors are used to collect data about the real-world (e.g., temperature sensor, humidity sensor, heart rate sensor, or camera sensor),

actuators are used control the physical world (e.g., watering systems, security systems, or air conditioner) [6]. Such interconnected IoT devices enable capturing the dynamic context of the physical world into the digital world.

While IoT enables exchanging and collecting data about the physical world over the Internet, BPM enables modeling, implementing, executing, monitoring, and analyzing business processes [7]. By enhancing business processes with IoT capabilities, process execution and monitoring as well as decision making can be enhanced. Furthermore, a more comprehensive view becomes possible for such IoT-aware business processes. Besides sensing the physical world, physical tasks such as moving a robot, as well as digital tasks, such as notifying a system, can be automated based on IoT devices [1]. By integrating the physical world as a key perspective in business processes, contextual information that was previously invisibly embedded in various environments can be continuously and automatically captured by IoT devices. IoT-aware business processes understand the dynamic context of the physical world, which makes them context-aware as well [8].

IoT has the ability to continuously and automatically support IoT-aware business processes with real-world IoT sensor data in real-time. IoT-driven decisions in business processes expose a need for context aggregation, context-awareness, and up-to-date (i.e. real-time) data, which are the key data source for dynamic decision making [8,9]. To address this need, IoT sensor data collection should proceed as follows (I) sensing low-level data from the real-world (e.g., temperature, switch state, humidity, brightness), (II) combining low-level data and aggregating them into high-level information, and (III) enabling decision making based on the obtained information [10]. This means, **low-level data** are captured in the physical world and need to be aggregated and combined to process-relevant **high-level data** [10,11]. Data from traditional repositories such as databases and data warehouses are not sufficient for IoT-aware decision making [1]. Decisions in IoT-aware business processes require up-to-date data about the physical environment [10]. For example, when using IoT devices such as temperature sensor, humidity sensor, and brightness sensor, the condition of the goods in a truck can be checked. By aggregating these low-level IoT data and combining them, decisions can be made in the course of a business process. Related to the example, the temperature and humidity value can be combined. If the maximum temperature and humidity are exceeded, the decision *start cooling system* can be made. We refer to this type of conditions *business rules*.

The integration of IoT in BPM has gained significant attention in literature, in particular several BPMN extensions and notations [19,32–34] have been proposed to integrate IoT in business processes in terms of resources. Consequently, IoT data is directly used without aggregating and combining it with other contextual process data. As a result, the possibility of generating high-level information is not exploited, which impairs the potential capability. In addition, decisions are traditionally hard-coded into business processes, which affects the ability to make dynamic decisions [8]. Current approaches mostly focus only on the integration of IoT into business processes to visually represent IoT involvement. The modeling, execution, and monitoring of IoT-driven business rules is neglected. Moreover, the integration of IoT and BPM is constrained due to the

lack of a methodological framework for connecting the IoT infrastructure with the BPM system [8]. For modeling business rules, in turn, the Decision Model and Notation (DMN) [12] standard can be used in combination with BPMN. By using DMN, the decision logic can be separated from the process logic. Furthermore, DMN enables the aggregation of low-level data into high-level one. However, DMN does not provide official support for modeling IoT-driven business rules, which creates new challenges. In this paper, we derive and highlight research challenges that need to be tackled in order to properly model, execute, and monitor IoT-driven business rules.

This paper is structured as follows: Sect. 2 illustrates the support for modeling, executing, and monitoring IoT-driven business rules based on either BPMN or BPMN plus DMN. In Sect. 3 we derive challenges that need to be tackled when modeling, executing, and monitoring IoT-driven business rules. Finally, in Sect. 4 we summarize and discuss the results.

2 Current Support for IoT-Driven Business Rules in BPMN and BPMN + DMN

2.1 IoT-Driven Business Rules in BPMN

BPMN 2.0 is a standardized graphical process modeling language that provides elements for modeling business processes and workflows [13]. However, BPMN 2.0 does not provide official support for modeling IoT involvement and capabilities, but provides different possibilities that can be used for representing IoT such as (i) tasks, (ii) events, and (iii) resources [14]. For IoT-driven business rules, different gateways may be used in the current BPMN 2.0 standard, these can be divided into the following categories (I) **exclusive**, (II) **inclusive**, (III) **parallel**, (IV) **complex**, and (V) **event-based** [13]. *Example 1* describes a business process with IoT-driven business rules. Note that the IoT-driven business rules are modeled exclusively with standard BPMN 2.0 elements.

Example 1: *Consider a medical system that monitors the health status of a patient who has been diagnosed with Chronic Obstructive Pulmonary Disease (COPD). COPD is a disease in which the lungs are permanently damaged and the airways (bronchi) are restricted. At anytime, the patient may experience unpleasant complications such as shortness of breath on exertion, coughing, sounds when breathing, fast heart rate, hyperactive muscle use, increased blood pressure, and a cold skin. Several studies [15, 16] have shown that the IoT-driven monitoring of sensor-equipped patients can improve their quality of life by identifying the severity of COPD disease and responding accordingly. In order to detect COPD, all required sensors are polled (cf. Fig. 1). Based on the values provided by the IoT sensors and the defined IoT-driven business rule, either no treatment, treatment with an oxygen mask, or treatment with an inhaler is administered.*

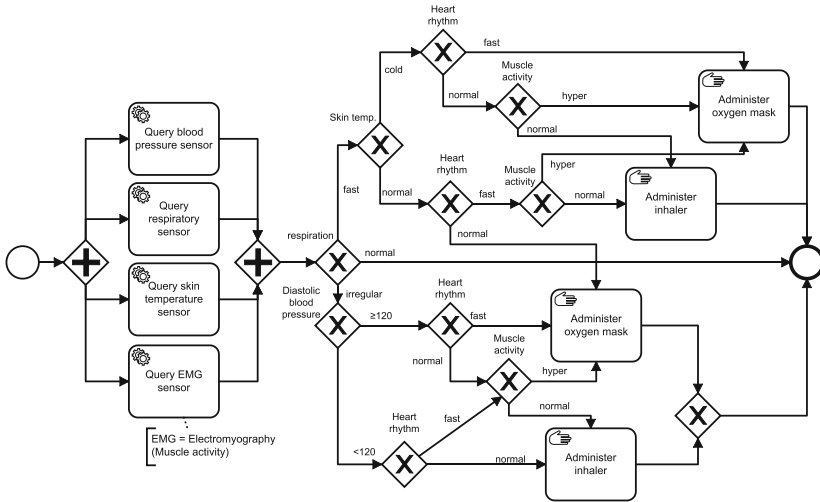


Fig. 1. Example of an IoT-driven business rules expressed in terms of BPMN 2.0

2.2 IoT-Driven Business Rules in BPMN + DMN

The combined use of BPMN [13] and DMN [12] has already been studied in [17] and [18]. The interplay between process and decision logic plays a crucial role for business processes, as business rules are evaluated during process execution and may affect process outcomes [17]. DMN is a decision modeling standard that consists of two levels: The first one represents the decision requirements, where the dependencies between the elements involved in the decision model are captured [8]. The decision requirements are represented by DRDs (Decision Requirements Diagrams) and form the dependencies between the data and sub-decisions. The input data for DRDs may be static or dynamic. The second level is the decision logic, which is usually modeled in terms of decision tables [5, 8]. To construct a DMN model, low-level data needs to be aggregated to higher-level one and enables consequently to aggregate contextual data [5, 12]. *Example 2* describes a decision-aware COPD process (cf. Fig. 2). Note that the business process is modeled in terms of BPMN 2.0 and the business rules in terms of DMN using the elements provided by the two standards.

Example 2: *To identify the severity of COPD, the patient is equipped with several sensors. The severity of COPD is determined based on the defined business rules (cf. Fig. 2 [6]), the data values provided by sensors in real-time (cf. Fig. 2 [2]), and data from a database (cf. Fig. 2 [3]) in DMN. The decision in DMN becomes evaluated when activating the Check COPD severeness business rule task in BPMN 2.0 (cf. Fig. 2 [1]). After deciding whether treatment with oxygen mask or inhaler, or no treatment becomes necessary, the heart status is checked based on real-time sensor data (cf. Fig. 2 [5]). Depending on the result, the patient is either not treated or treated with a defibrillator.*

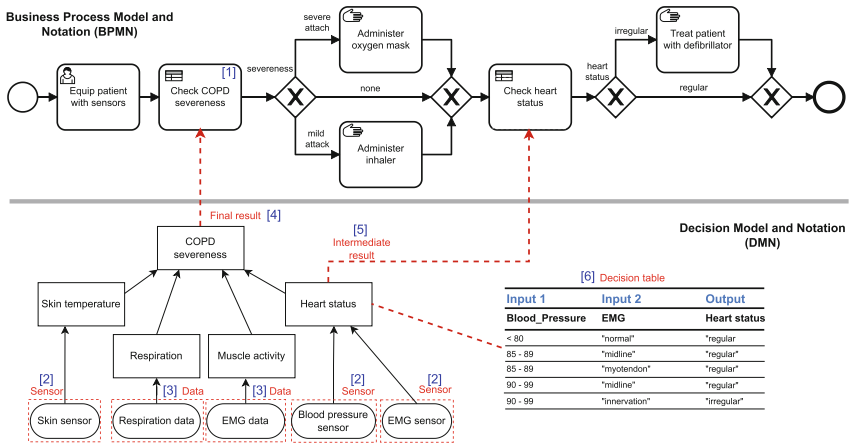


Fig. 2. Relationship between BPMN 2.0 and DMN

3 Challenges

Although the BPMN 2.0 and DMN allow expressing certain aspects of IoT-driven business rules, several challenges remain [5,14]. We studied literature and IoT-driven business rules from different domains modeled in terms of either BPMN or BPMN + DMN. Moreover, we were able to identify additional challenges in the context of the two IoT-related projects BPMN Extension for IoT (BPMNE4IoT) [2] and IoT Decision Making for BPMN (IoTDM4BPMN) [10] we are involved in. We discuss the derived challenges and divide them along the modeling, execution, and monitoring of IoT-driven business rules. The following structure is applied for each challenge; we briefly describe the challenge, provide an example, and reference relevant literature that tries to address the research gaps described in the challenge. A summary of the challenges can be found in Table 1.

Table 1. Challenges for IoT-driven business rules in BPMN and BPMN + DMN.

Modeling challenges
C1 - Modeling IoT-driven business rules in BPMN 2.0
C2 - Modeling IoT-driven business rules with BPMN 2.0 + DMN
C3 - Reducing the complexity of IoT-driven business rules
Execution challenges
C4 - Extending process log with IoT data
C5 - Treatment of IoT data outliers
C6 - Treatment of defective IoT devices
Monitoring challenges
C7 - Traceability of IoT-driven business rules
C8 - Fault monitoring in IoT-driven business rules
C9 - Real-time monitoring of IoT-driven business rules

3.1 Modeling Challenges

C1 - Modeling IoT-Driven Business Rules in BPMN 2.0.

Description: In Sect. 2.1, we discussed the current support of the BPMN 2.0 standard for incorporating IoT devices and the modeling of IoT-driven business rules. In order to express IoT involvement within business rules, it should be possible to model the involved IoT devices. Note that the returned data values of the IoT devices are used as basis for evaluating the IoT-driven business rule. Therefore, it is crucial to be able to properly capture IoT involvement.

Example: To describe the problem, Fig. 3 illustrates the treatment of COPD. When modeling the business process and the corresponding IoT-driven business rules with the standard BPMN 2.0 elements, it remains unclear which tasks are IoT-related and which are not. Furthermore, it is unclear which sensors are actually used in the context of IoT-driven business rules (cf. Fig. 3). To decide whether no treatment, a treatment with oxygen mask, or a treatment with inhaler is required, skin temperature (Task 2), respiration (Task 3), and EMG value (Task 4) are needed, whereas the ECG value (Task 5) is not needed for the treatment but for the alarm. Note that it is unclear which sensor is important for which business rule. The entire process model must be carefully read and understood in order to determine this. Furthermore, it is impossible to distinguish between sensors (Tasks 2–5) and actuators (6). In order to distinguish between sensors and actuators, the labels should reflect the involvement of IoT and the modeler needs to be familiar with IoT devices and their behavior. As another drawback, no visual difference between IoT-aware service tasks (Tasks 2–6) and BPMN service task (Tasks 1, 9, 10) exists. Moreover, the complexity and thus, the comprehensibility of the business rules increases with growing number of involved IoT devices. This aggravates reading and understanding of the process models as well as the IoT-driven business rules. With increasing number of business rules and increasing complexity of the rule logic, the flexibility, scalability, and maintainability of the resulting process model and IoT-driven business rules is impaired. The complex nesting and ambiguous involvement of IoT makes any later extensions or changes difficult. As IoT-driven business rules are hard-coded in the business process in form of gateways, aggregation and combination of IoT low-level data into high-level one is not appropriate with BPMN 2.0. Obviously, the IoT-driven business rules cannot be reused in a different context. When using BPMN, both process and decision logic are defined in one and the same process model. As a result, the modeled logic is hard-coded and constrained to a local location. Therefore, reusability is impaired [5].

Possible Solution: There exist several works [2, 19, 32, 33] that introduce extensions for representing IoT devices in the context of BPMN 2.0. These extensions enable the explicit modeling of IoT participation by introducing IoT specific elements. They propose a visual discrimination between regular BPMN elements and IoT elements in the modeling phase [14]. In [35], two approaches for modeling IoT-driven business rules are presented. The first one extends the BPMN 2.0

standard by providing specific IoT decision modeling elements. Note that BPMN 2.0 is a rather complex language and any extension constitutes a deviation from the standard [14,25]. By extending BPMN with additional IoT elements complexity might increase. In turn, this might effect model comprehensibility. The second approach proposes an IoT-specific drag&drop modeler, which separates the business rules from the process logic. As the drag&drop modeler outsources the business rules from the BPMN process model, the structure of the IoT-driven business rule cannot be viewed in BPMN. This makes it difficult to extend, maintain, and troubleshoot the IoT-driven business rules.

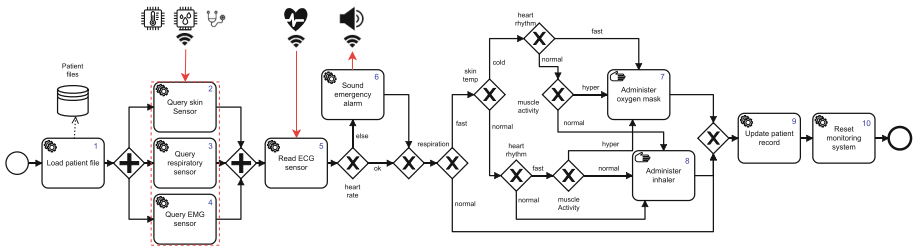


Fig. 3. IoT awareness in BPMN-based process model (adopted from [2]).

C2 - Modeling IoT-Driven Business Rules with BPMN 2.0 + DMN.

Description: Combining BPMN and DMN can solve some of the problems and gaps mentioned above. For example, DMN is suited for aggregating and combining business rules as it uses appropriate techniques such as decision tables. Since DMN does not provide any explicit elements for modeling IoT, the modeling of business rules based on IoT data constitutes a challenge.

Example: Consider Fig. 4. The results of the queried IoT sensors are stored in data objects which then flow into the business rule task *Check COPD severeness* (cf. Fig. 4). Representing the received IoT data as data objects increases the complexity and the number of modeling elements in the business process. If IoT data is not represented in terms of data objects, such as in the process model depicted in Fig. 3, it will be unclear which IoT data actually concern the business rules in DMN. This, in turn, affects model readability and comprehensibility. In addition, it is impossible to distinguish between IoT data objects on the one hand and BPMN data objects on the other. Note that the *Check COPD severeness* business rule task represents the decision modeled and executed in DMN. As DMN outsources the decision logic from the BPMN process model, the structure of the IoT-driven business rule cannot be directly viewed in BPMN. Typically, the business rule task only provides the final decisions. In addition, DMN does not officially support the modeling of IoT-driven business rules. Consequently, it cannot be distinguished between IoT input data (cf. Fig. 3 [2]) and, for example, input data from a database (cf. Fig. 3 [3]). When using DMN, decision logic is

captured in decision tables (cf. Fig. 3 [4]). With increasing number of IoT devices, however, the complexity of the decision table increases as well. Accordingly, the error detection becomes more difficult.

Possible Solution: Several authors have argued that DMN is capable of modeling IoT-driven business rules [5, 49]. For example, [5] shows how DMN elements can be used to model different IoT-driven business rules (e.g., smart transportation, smart ventilation, and smart healthcare). Thus, no discrimination between regular DMN elements (e.g. input data) and IoT-related DMN elements is present. One possible solution to close this gap would be to extend the DMN standard with IoT decision elements.

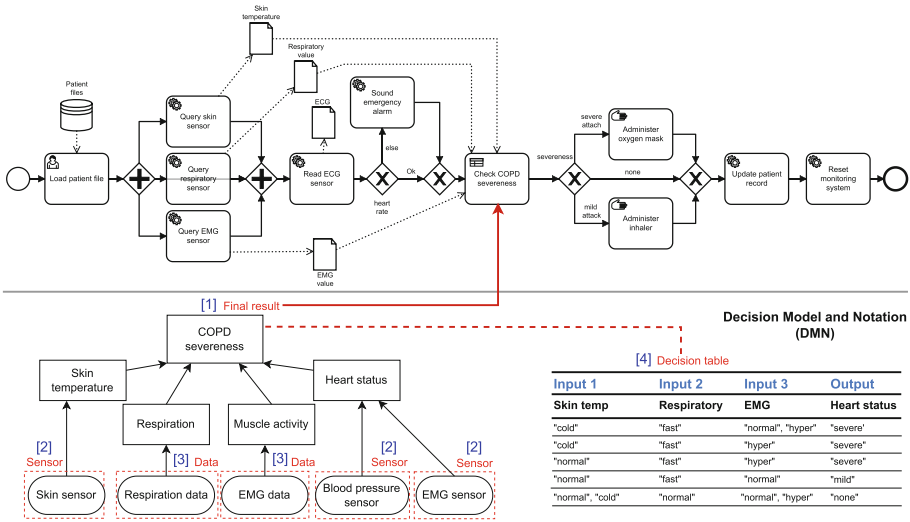


Fig. 4. Using BPMN 2.0 and DMN for modeling an IoT-driven business rule.

C3 - Reducing the Complexity of IoT-Driven Business Rules

Description: As discussed in the context of C1 and C2, the complexity of both the process and the decision model increases when involving IoT devices. Modeling IoT-awareness for a process and decision model is a complex undertaking, and the resulting model often turns out to be difficult to understand due to the potentially ambiguous use of modeling elements [2]. IoT-driven business rules become more complex when modeling them with BPMN 2.0 as the involvement of IoT is not supported by the standard. In turn, this has a negative repercussion on modeling IoT-aware processes and IoT-driven business rules.

Example: As processes running in an IoT setting are often data- and decision-intensive, the modeled process might be too extensive and, thus, too complex

to be understandable. When modeling IoT-driven business rules in BPMN (cf. Fig. 3), the number of gateways and control flow paths grows as additional business rules are introduced. This leads to a large number of branch conditions and control flow elements, resulting in a complex structure [14]. When combining BPMN and DMN we can encapsulate this complexity by defining the business rules in decision tables. This significantly reduces the number of gateways and sequence flows on one hand. On the other, the business rules are hidden in decision tables. At the BPMN level, it is impossible to see how the IoT-driven business rules are defined and how they depend on each other. In the following, we consider metrics proposed in literature to evaluate the complexity of the models described in the previous sections [14, 26]. The metrics are used for identical processes with different modeling approaches. The BPMN metrics NOA (number of activities), NOG (number of gateways), and NOF (number of flows) were defined in [26]. The metrics for DMN are taken from [27] and consist of TNR (total number of rules), NOD (number of decisions), and TNDO (total number of data objects). Table 2 shows that the complexity of modeling IoT-driven business rules in BPMN is larger compared to BPMN + DMN due to the higher number of sequence flows (NOF) and gateways (NOG). As opposed to BPMN, the complexity in BPMN + DMN is shifted to the decision tables (TNR).

Table 2. Evaluation of complexity through the application of metrics

Case	NOA	NOG	NOF	TNR	NOD	TNDO
BPMN	10	11	31	–	–	–
BPMN+DMN	11	6	23	17	5	5

Possible Solution: A possible solution for reducing complexity related to IoT-aware processes is presented in [2]. The authors introduce new modeling elements that, for example, merge individual sensor artifacts into one sensor group artifact in order to increase the abstraction level. Another possible solution is the definition of guidelines for IoT-driven business rules. For example, [37] proposes seven process modeling guidelines (7PMG). However, these do not consider the modeling of IoT-driven business rules. Another approach is the definition of patterns for modeling IoT behavior. These patterns could, for example, reduce the number of message flows between the central pool and the IoT-aware pools by using the computing capacities of the IoT devices [50].

3.2 Execution Challenges

C4 - Extending Process Log with IoT Data

Description: The sensors used in a business process record the physical world and transform it into the digital world. The data generated by IoT devices allows for the continuous monitoring and provision of opportunities for analysing and

optimizing the performed processes, e.g., through process mining or real-time monitoring [41, 42]. Furthermore, as a data source IoT can improve the verification of the conformance between the actual execution of a business process in the physical world and its execution as recorded by the Business Process Management System (BPMS) based on a secondary log of sensor data [1]. Through the use of common business process engines (e.g. Camunda [20]), which are unaware of IoT involvement, extending the process log with IoT data is difficult and complex. As standard BPMN elements are used for modeling the IoT-driven business rules, the business process engines is unaware of the involvement of IoT, whereby the event log cannot be extended by the engine with IoT data.

Example: Process mining has become an important research area in Computer Science, which aims to extract knowledge from event logs to discover, monitor, and improve business processes [43, 44]. To allow for a finer grained discovery, monitoring, and improvement of IoT-aware business rules, the event log needs to be extended with IoT-related data collected from smart objects. Extending the event log to include IoT data requires an IoT-aware process engine and a suitable architecture. Most IoT infrastructures are based on isolated IoT devices and integrated with applications that are not necessarily process-aware. Furthermore, such applications are often based on proprietary control software with non-standard interfaces [42].

Possible Solution: A possible solution to enhance a business rule log with IoT data is to use or develop an IoT-aware business rule engine as well as an embedding architecture. The IoT-aware engine should be able to detect IoT actions and record them in the log. Another way to extend the log with IoT data is to capture the IoT actions in a separate event log. Then the IoT event log may be merged with the process event log.

C5 - Treatment of IoT Data Outliers

Description: Outliers constitute irregularities or behavioral deviations of the IoT devices and the delivered IoT data. IoT sensors are responsible for capturing, collecting, and transmitting data. The data collected from the physical environment, however, might be prone to outliers [28]. The treatment of outliers is very important in relation to IoT-driven business rules to avoid erroneous decisions being made based on the faulty sensor data.

Example: The IoT is used in a wide variety of environments and scenarios, e.g. environmental monitoring, smart cities, disaster warning, and agriculture. In this context, sensors are often installed in harsh environments. As a consequence, the sensors are susceptible to malfunction, rapid wear, and tampering. This, in turn, can lead to outliers [28]. In Table 3, fault categorizations in IoT implementations are mentioned and summarized [24].

Table 3. IoT fault categorizations [24].

Fault	Definition
Short	An IoT data point deviates significantly from the expected temporal or spatial trend of the data.
Stuck-at	A series of data points has zero or almost zero variation for a period of time greater than expected.
Noise	Sensor data exhibiting an unexpectedly high amount of variation

Illustrating Example: Consider an environmental monitoring station that consists of temperature sensors, humidity sensors, and brightness sensors. If the brightness and temperature sensors are manipulated such that they are directly hit by the sun, the IoT sensor will provide distorted values. This, in turn, leads to the faulty execution of corresponding IoT-driven business rule.

Possible Solution: There are different approaches [45–47] and techniques (e.g. machine learning) for detecting and treating of outliers. Most approaches, however, do not equip their IoT-aware business rule engine or architecture with the techniques for handling outliers. One possible solution is to equip the architecture with middleware that detects the IoT outliers and handles them accordingly.

C6 - Treatment of Defective IoT Devices

Description: Handling defective IoT devices constitutes another major challenge to be tackled in the context of IoT-driven business rules. As discussed in C5, IoT-driven business rules need real-time processing. In this context, IoT sensors must provide a result within an acceptable amount of time. The challenge is to detect and handle defective or non-reachable IoT sensors.

Example: As IoT devices constitute electronic components, they might suddenly fail and then no longer function [29]. Such an IoT device failure result in the evaluation of IoT-driven business rules with missing measured values or zero values. Note that this might lead to several runtime issues in the business process relying on these rules. In literature, such failures are referred to as binary failures [30]. The use of IoT devices in harsh environments and limited computing capacity can lead to failures as well. Other reasons include limited battery life, hardware failures, and human mistakes [31]. The failure of IoT devices involved in IoT-driven business rules might have dire consequences. For example, if a queried IoT sensor is not accessible, no decision can be made, which can lead to deadlocks as the workflow engine continuously checks whether the condition is met or not. Another possible sequence would be the execution of an incorrect business rule. If an IoT sensor suffers from a binary failure, i.e., it returns a zero value by default or the occurrence of an error, it might result in an incorrect business rule executed by mistake. Assume, for example, that the business rule “*temperature_sensor < 25°C*” returns a zero value if the referred temperature sensor is defective; the condition would still be met.

Possible Solution: Several techniques [29,30] exist for detecting of defective IoT devices. However, there is no specific approach that deals with such failures in the context of IoT-driven business rules. One approach is to use a middleware that enables fault tolerance based on redundant IoT devices or the replacement of faulty IoT data by accessing historical records, depending on the duration of the outage. Another solution is to introduce a prioritization mechanism for IoT devices. The process modeler can assign priority levels for IoT devices. Depending on the priority level of the defective IoT device, the process execution may be aborted or the IoT devices be ignored for decision making.

3.3 Monitoring Challenges

C7 - Traceability of IoT-Driven Business Rules

Description: When monitoring IoT-driven business rules (during both process and business rule execution) traceability constitutes a fundamental challenge. Traceability refers to the understanding of the decision resulting from the evaluation of an IoT-driven business rule. To understand what triggered an IoT-driven business rule, it is crucial to comprehend which IoT devices were queried how and why. A monitoring approach should address this challenge and present both the modeled and the executed IoT-driven business rules in a structured and understandable way.

Example: With increasing context-intensity of the environment in which the IoT-driven business rule is performed, the number of monitoring challenges increases. Moreover, an IoT sensor may be used by multiple business rules. Furthermore, the results of one IoT-driven business rule can be utilized by other rules, resulting in a complex nesting. Therefore, the traceability of IoT-driven business rules must be monitored in an appropriate manner during execution.

Possible Solution: [10] introduced a BPMN 2.0 extension that represent business rules graphically in combination with a truth table. This approach further uses overlays and color highlighting to visualize the result of a business rule execution. Although it is possible to determine how the decision was reached, it is not possible to reconstruct the exact decision-making process of a business rule in retrospect, as detailed temporal information is missing, especially for time-critical sensor queries. One possible solution is to time-stamp the incoming data of the queried IoT devices and each evaluation of a business rule, and to visualize it accordingly in the process.

C8 - Fault Monitoring in IoT-Driven Business Rules

Description: Monitoring errors constitute another challenge. If IoT devices are used in business rules without receiving any feedback on faults, the IoT-driven business rule process might suffer. Detecting and monitoring IoT devices involved in business rules provides the visibility needed to understand exactly what went wrong and where it went wrong, to subsequently ensure that this error is avoided in the future.

Example: IoT sensors generate large amounts of data and operate automatically and continuously. In order to ensure that sensors properly work in the context of IoT-driven business rules, a precise monitoring system is needed to check the behavior and performance of the IoT sensors. As a key monitoring challenge faulty sensors need to be detected during runtime. Another challenge is to detect and monitor anomalies and outliers. Moreover, the monitoring of IoT devices supports the confidence of the data collected by the sensor and, thus, the quality of the business rules. The higher the confidence of the received IoT data is, the better and more precise the resulting IoT-driven business rule will be. For example, if four temperature sensors are used in a smart home to calculate the average temperature, but only two sensors provide a value as the other two are defective, the confidence of the data is compromised. As a consequence, the resulting output of the IoT-driven business rule will be not accurate and possibly an incorrect action be performed. Without a monitoring system detecting defective IoT devices, the discovery of such scenarios would be not possible.

Possible Solution: [10] uses color highlighting (e.g., green, orange, and red) of IoT devices (sensor artifact) involved in the business rule in case an error such as a timeout occurs while polling sensor data. In addition, the corresponding error message is displayed in the execution log. However, the detection is limited to communication errors and errors in the source code. A possible solution is to realize a component that not only detects defective IoT devices, but outliers and insufficient data quality as well. Such a component can mark and visualize the respective errors and provide additional information about the kind of error.

C9 - Real-Time Monitoring of IoT-Driven Business Rules

Description: Real-time monitoring of IoT-driven business rules enables continuously updated information streamed with low latency. In turn, the continuous streaming of up-to-date IoT data allows for the immediate detection of problems, i.e., based on the real-time monitoring of IoT-driven business rules alerts can be forwarded more quickly to systems for mitigation in the event of a failure of IoT devices.

Example: Monitoring IoT-driven business rules and relevant IoT devices in real-time constitutes another challenge. In particular, this monitoring shall provide additional information about the current processing state of the respective rule. For example, it needs to be monitored which IoT-driven business rules are currently running, in what state they are (e.g., are there faulty IoT devices?), and what will happen next. For the real-time monitoring of IoT-driven business rules, the monitoring system needs to communicate with all IoT devices involved in process and rule execution. Furthermore, the monitoring system should be extensible and scalable to be able to add IoT sensors and IoT-driven business rules on the fly.

Possible Solution: Several works [10,39,40,48] exist for monitoring IoT-driven business rules after their execution. In context, during execution it is only possible to monitor the final result of the IoT-driven business rules, but not how they are composed and which intermediate results (of sub-rule evaluation) exist. Moreover, it is not indicated in what state the rules are (e.g. ready, executing, or finish). One possible solution is to develop a suitable architecture that delivers all existing information about the IoT-driven business rules to the monitoring system in real-time. Note that this requires appropriate communication protocols and a real-time capable monitoring component in the architecture.

4 Conclusion

In this paper, modeling, execution, and monitoring IoT-driven business rules was examined by either exploring corresponding rules with BPMN 2.0 or BPMN in combination with DMN. The IoT adds value to BPM through its ability to transform the physical world to its digital twin. Integrating BPM with IoT capabilities should exploit the complete potential of IoT and cover all relevant use cases in this context. Current research related to the combination of IoT and BPM is concerned with the integration of IoT with BPM as a resource. Less attention is paid to the integration of IoT into BPM for decision making through IoT-driven business rules. As a result, several challenges exist with respect for the modeling, execution, and monitoring of IoT-driven business rules. In this paper, in addition to exploring the current support for IoT-driven business rules in BPMN and BPMN + DMN, we have derived these challenges through studying literature, real-world IoT-driven business rules, and hands-on experiences in our IoT-related projects BPMN Extension for IoT (BPMNE4IoT)[2] and IoT Decision Making for BPMN (IoTDM4BPMN) [10]. Existing solutions were described based on a literature review, including a discussion of their strengths and weaknesses. If no solution was found in literature, a possible solution approach was discussed.

The identified challenges should be addressed in future with the goal of enabling the integration of IoT and BPM for executing of IoT-driven business rules.

References

1. Janiesch, C., et al.: The internet of things meets business process management: a manifesto. In: Systems, Man, and Cybernetics Magazine (2020)
2. Kirikkayis, Y., Gallik, F., Reichert, M.: Towards a comprehensive BPMN extension for modeling IoT-aware processes in business process models. In: 16th International Conference on Research Challenges in Information Science (RCIS) (2022)
3. Ashton, K.: That ‘internet of things’ thing. *RFID J.* **22**(7), 97–114 (2009)
4. Chang, C., Srirama, S., Buyya, R.: Mobile cloud business process management system for the internet of things: a survey. *ACM Comput. Surv.* **49**(4), 1–42 (2016)



5. Hasić, F., Serral, E., Snoeck, M.: Comparing BPMN to BPMN + DMN for IoT process modelling: a case-based inquiry. In: 35th ACM/SIGAPP Symposium on Applied Computing (2020)
6. Valderas, P., Torres, V., Serral, E.: Modelling and executing IoT-enhanced business processes through BPMN and microservices. *J. Syst. Softw.* **184**, 111139 (2022)
7. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Process-aware information systems. In: *Fundamentals of Business Process Management*, pp. 341–369. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-56509-4_9
8. Song, R., Vanthienen, J., Cui, W., Wang, Y. and Huang, L.: Context-aware BPM using IoT-integrated context ontologies and IoT-enhanced decision models. In: *Conference on Commerce and Enterprise Computing* (2019)
9. Koschmider, A., Mannhardt, F. and Heuser, T.: On the contextualization of event-activity mappings. In: *Business Process Management Workshops* (2018)
10. Kirikkayis, Y., Gallik, F. and Reichert, M.: IoTDM4BPMN: a IoT decision making framework for business processes in BPMN. In: *International Conference on Service Science* (2022)
11. Krishnamurthi, et al.: An overview of IoT sensor data processing, fusion, and analysis techniques. *Sensors* **20**(21), 6076 (2020)
12. OMG: Decision Model and Notation (DMN) 1.2 (2018)
13. OMG: Business Process Model and Notation (BPMN) 2.0, (2010)
14. Hasić, F. and Serral, E.: Executing IoT processes in BPMN 2.0: current support and remaining challenges. In: *13th International Conference on Service Science* (2019)
15. Zhou et al.: An internet of things based COPD managing system: its development, challenges and first experiences. In: *Clinical eHealth* (2019)
16. Xiang, G., et al.: Clinical guidelines on the application of Internet of Things (IoT) medical technology in the rehabilitation of chronic obstructive pulmonary disease. *J. Thorac. Dis.* **13**(8), 4629 (2021)
17. Bazhenova, E., et al.: From BPMN process models to DMN decision models. *Inf. Syst.* **83**, 69–88 (2019)
18. Combi, C., et al.: Seamless design of decision-intensive care pathways. In: *International Conference on Healthcare Informatics (ICHI)* (2016)
19. Yousfi, A., et al.: uBPMN: a BPMN extension for modeling ubiquitous business processes. *Inf. Softw. Technol.* **74**, 55–68 (2016)
20. Camunda: Process Engine. <https://docs.camunda.org/manual/7.8/user-guide/process-engine/>. Accessed 20 Apr 2022
21. SAP Signavio: SAP Signavio Process Governance. <https://documentation.signavio.com/suite/en-us/Content/workflow-accelerator/userguide/intro.htm>. Accessed 20 Apr 2022
22. Gruhn, V., et al.: BRIBOT: towards a service-based methodology for bridging business processes and IoT big data. In: Hacid, H., Kao, O., Mecella, M., Moha, N., Paik, H. (eds.) *ICSOC 2021*. LNCS, vol. 13121, pp. 597–611. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91431-8_37
23. Torres, V., et al.: Modeling of IoT devices in business processes: a systematic mapping study. In: *Conference on Commerce and Enterprise Computing (CEC)* (2020)
24. Chakraborty, T., et al.: Fall-curve: a novel primitive for IoT fault detection and isolation. In: *Embedded Networked Sensor Systems* (2018)
25. Indulska, M., Muehlen, M., Recker, J.: Measuring method complexity: the case of the business process modeling notation (2009)

26. De Oca, IMM., Snoeck, M.: Pragmatic guidelines for business process modeling, SSRN 2592983 (2014)
27. Hasic, F., Vanthienen, J.: Complexity metrics for DMN decision models. *Comput. Stan. Interfaces* **5**, 15–37 (2019)
28. Gaddam, A., et al.: Detecting sensor faults, anomalies and outliers in the internet of things: a survey on the challenges and solutions. *Electronics* **9**(3), 511 (2020)
29. Pachauri, G., Sharma, S.: Anomaly detection in medical wireless sensor networks using machine learning algorithms. *Procedia Comput. Sci.* **70**, 325–333 (2015)
30. Ye, J., Stevenson, G., Dobson, S.: Detecting abnormal events on binary sensors in smart home environments. *Pervasive Mob. Comput.* **33**, 32–49 (2016)
31. Choi, J., et al.: Detecting and identifying faulty IoT devices in smart home with context extraction. In: *Conference on Dependable Systems and Networks* (2018)
32. Sungur, C.T., et al.: Extending BPMN for wireless sensor networks. In: *Conference on Business Informatics* (2013)
33. 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
34. Cheng, Y., et al.: Modeling and deploying IoT-aware business process applications in sensor networks. *Sensors* **19**(1), 111 (2019)
35. Kirikkayis, Y., Gallik, F., Reichert, M.: Visual decision modeling in IoT-aware processes. In: *Central European Workshop on Services and their Composition* (2020)
36. Polak, P.: BPMN impact on process modeling (2013)
37. Mending, J., Reijers, H., van der Aalst, W.: Seven process modeling guidelines (7PMG). *Inf. Softw. Technol.* **52**(2), 127–136 (2020)
38. Gallik, F., Kirikkayis, Y., Reichert, M.: Modeling, executing and monitoring IoT-aware processes with BPM technology. In: *International Conference on Service Science* (2022)
39. Song, R., Vanthienen, J., Cui, W., Wang, Y., Huang, L.: Context-aware BPM Using IoT-integrated context ontologies and IoT-enhanced decision models. In: *Conference on Commerce and Enterprise Computing* (2019)
40. Oliveira, R., et al.: An intelligent model for logistics management based on geofencing algorithms and RFID technology. *Expert Syst. Appl.* **42**(15–16), 6082–6097 (2015)
41. Pegoraro, M., van der Aalst, W.M.: Mining uncertain event data in process mining. In *2019 International Conference on Process Mining (ICPM)* (2019)
42. Seiger, R., et al.: Towards IoT-driven process event log generation for conformance checking in smart factories. In: *International Enterprise Distributed Object Computing Workshop* (2020)
43. van der Aalst, W.: *Process Mining: Data Science in Action*. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
44. 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
45. Samara, M.A., et al.: A survey of outlier detection techniques in IoT: review and classification. *J. Sens. Actuator Netw.* **1**(1), 4 (2021)
46. Jiang, J., et al.: Outlier detection approaches based on machine learning in the internet-of-things. *Wireless Commun.* **7**(3), 53–59 (2020)
47. Bhatti, M.A., et al.: Outlier detection in indoor localization and Internet of Things (IoT) using machine learning. *J. Commun. Netw.* **22**(3), 236–243 (2020)
48. Celestrini, J.R., et al.: An architecture and its tools for integrating IoT and BPMN in agriculture scenarios. In: *Symposium on applied computing* (2019)

49. Song, R., Vanthienen, J., Cui, W., Wang, Y., Huang, L.: A DMN-based method for context-aware business process modeling towards process variability. In: Abramowicz, W., Corchuelo, R. (eds.) BIS 2019. LNBIP, vol. 353, pp. 176–188. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20485-3_14
50. Domingos, D., et al.: Automatic decomposition of IoT aware business processes - a pattern approach. In: International Conference on ENTERprise Information Systems (CENTERIS) (2019)



Enhanced Transformation of BPMN Models with Cancellation Features

Giorgi Lomidze¹, Daniel Schuster^{1,2} , Chiao-Yun Li^{1,2},
and Sebastiaan J. van Zelst^{1,2} 

¹ RWTH Aachen University, Aachen, Germany
sebastiaan.van.zelst@fit.fraunhofer.de

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

Abstract. Canceling ongoing process instances is a natural phenomenon in practice. As such, modeling cancellation behavior is supported in the Business Process Model and Notation (BPMN) via exception events. Event-data-driven analysis techniques using such process models, e.g., conformance checking, require converting the BPMN model into a formal process modeling representation, i.e., Petri nets. However, the existing transformation of BPMN models with exception events renders a classical Petri net, with various additional modeling constructs to mimic the exception behavior. Using such a model in a subsequent analysis renders an infeasible computational complexity. Hence, this paper presents a novel conversion of BPMN models with exception events into reset nets, significantly reducing the number of required invisible transitions in the corresponding transformation. Our results show that the enhanced conversion reduces the computational effort of using the converted models for conformance checking.

Keywords: Business process management · Business process modeling · BPMN · Petri nets

1 Introduction

Several business process modeling languages exist that allow modeling the behavior of the processes in an organization [19]. The Business Process Model and Notation (BPMN) [20] language is a commonly used standard for business process modeling. BPMN is a graph-based language and includes over 50 distinct modeling elements. Among these elements are *exception events*, which allow one to specify exceptional behavior w.r.t. the default process behavior, e.g., a sensor value exceeding a certain threshold may require a special recovery procedure.

Whereas exceptions refer to *non-standard behavior*, adequate handling of an exception that occurred is vital. To verify whether the appropriate exception handling procedure is followed, *conformance checking techniques* [8] can be applied. Such techniques compare historical process executions stored in the

information system of a company, i.e., referred to as *event data*, with a reference process model describing the main control flow of a process (and corresponding exception handling procedures).

Most conformance checking techniques are not designed to use BPMN as an input, i.e., they use *Petri nets* [23]. When applying conformance checking techniques on a process modeled in BPMN, the model is first transformed into a Petri net, after which the conformance checking algorithm is invoked. However, in the case of BPMN models with exception events, such a translation renders a Petri net with many additional modeling constructs to mimic the exception behavior. Applying conformance checking on such a transformed Petri net model is often infeasible in the context of computational complexity.

Transforming BPMN models with cancellation features to Petri nets is partly addressed in the literature (see Table 1 for an overview) and applying algorithms on the resulting Petri nets lacks efficiency. Hence, we propose an enhanced transformation that solves these limitations. Our transformation uses the concept of *reset nets* (R-nets) [14], i.e., Petri nets enhanced with means to “reset behavior”. The main advantage of our proposed transformation is that the number of elements required to model the same behavior as the corresponding BPMN model is significantly reduced w.r.t. the existing transformation procedures. Such a smaller model size, in turn, is beneficial for the computational complexity of model-driven analysis techniques, e.g., conformance checking.

We conducted experiments with BPMN models, including exception events, comparing the existing transformation approach to our newly proposed transformation. Our experiments explicitly focus on conformance checking analysis of the transformed models. Our experiments confirm that the resulting R-nets, i.e., obtained by applying our newly proposed transformation rules, yield significantly reduced computational complexity in the context of conformance checking.

The remainder of this paper is structured as follows. In Sect. 2, we discuss related work. In Sect. 3, we present background concepts, i.e., BPMN models and R-nets. Section 4 presents our method. In Sect. 5, we present the evaluation of our proposed transformation. Section 6 concludes this paper.

2 Related Work

A general overview of the BPMN language is presented in [20]. Here, we discuss mappings from BPMN to other modeling languages. In Table 1, we present an overview of these approaches, including our approach. The table shows the supported constructs in the conversion, i.e., sub-process support, internal/external exceptions, terminations, timeouts, and OR-join constructs. Note that timeouts and OR-Joins are easily supported in our proposed transformation. However, for simplicity and brevity, they are not described in this work.

Arbab et al. [4] propose to convert BPMN models to the *Reo language* [3] and focus on model verification and compliance analysis. Kheldoun et al. [16] suggest mapping BPMN models with cancellation to *RECAT nets*, i.e., a tree-based structure that utilizes Petri nets. Kherbouche et al. [17] propose model

Table 1. Overview of BPMN transformations to other modeling languages. Per construct, a ✓-symbol indicates full support; a (✓)-symbol indicates restrictions or missing information on the exact mapping of the corresponding construct.

Author	Year	Target	Supported BPMN Constructs					
			Sub-process	Int. Exception	Ext. Exception	Termination	Timeout	OR-Join
van der Aalst et al. [1]	2002	Petri net	(✓)					
Dijkman et al. [12]	2007	Petri net	✓	(✓)	✓	(✓)	(✓)	✓
Raedts et al. [22]	2007	Inhibitor net	✓				✓	
Arbab et al. [4]	2008	Reo	✓	✓	✓	(✓)	✓	(✓)
Decker et al. [10]	2008	YAWL	✓	✓	(✓)	(✓)	✓	✓
Ou-Yang and Lin [21]	2008	Colored Petri net	✓					(✓)
Ye et al. [27]	2008	YAWL	✓	(✓)			✓	✓
Ye and Song [26]	2010	YAWL	(✓)	(✓)	✓		✓	✓
Decker et al. [11]	2010	YAWL	✓	✓	(✓)	(✓)	✓	✓
Kherbouche et al. [17]	2013	Kripke structure	✓					✓
Kheldoun et al. [16]	2015	RECAT net	✓	✓	✓		✓	
Dechsupa et al. [9]	2019	Colored Petri net	✓					
<i>This work</i>	2022	reset net	✓	✓	✓	✓	(✓)	(✓)

checking for BPMN by exploiting a transformation to *Kripke structures*. Various transformations of BPMN to *Yet Another Workflow Language (YAWL)* exist [10, 11, 26, 27]. The transformations support sub-processes, internal exceptions, timeouts, and OR-joins. The YAWL model is projected to a reset net, where backward reasoning is required to handle OR-joins. The concept of cancellation is directly translated to reset arcs. Additional transformations supporting cancellation behavior exist for YAWL models, e.g., to Inhibitor nets [24] and Generalized Stochastic Petri nets with inhibitor arcs [6].

Several authors focus on transforming BPMN models to Petri nets (or subclasses thereof). A general overview of transformations from arbitrary process modeling languages (including BPMN) to Petri nets is presented in [18]. Most existing mappings have in common that the essential BPMN workflow components, i.e., start/intermediate/end events, sequence flows, tasks, and AND-/XOR-/OR-gateways, are mapped using the same approach. In [1] van der Aalst et al. propose a complete set of transformation rules for *Workflow graphs* (which can be seen as a generalization of simple BPMN models). In [22], a mapping from BPMN to Petri nets with inhibitor arcs is presented for model checking. The inhibitor arcs are used to model timeout events. Dechsupa et al. [9] map a BPMN model to a Colored Petri Net (CPN). An effective way of transforming block-structured OR-joins and OR-splits is provided. In both works mentioned above, general cancellation is not supported. The transformation approach presented in [12] is most complete w.r.t. cancellation behavior. The presented algorithm first transforms the basic workflow patterns to Petri net elements with equivalent behavior. Subsequently, sub-process features are mapped. External and internal sub-process cancellation events and nested sub-processes are transformed. Since Petri nets cannot reset certain areas by just firing one transition, the tokens are bypassed to the end of the cancellation area once the exception event has been fired.

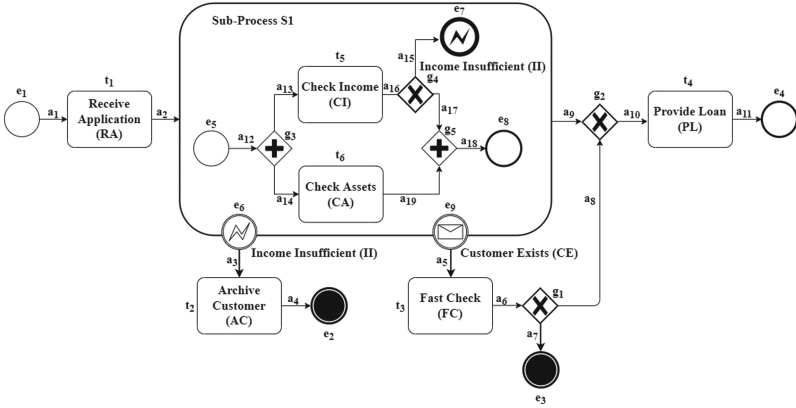


Fig. 1. Example BPMN model containing the core modeling concepts assumed in this paper. We use the model as a running example.

In summary, support for the conversion of BPMN models that include cancellation behavior to Petri net-oriented models is limited. To obtain a Petri net from a BPMN model with advanced cancellation constructs, a mapping via YAWL is required. To the best of our knowledge, the transformation proposed in this paper is the first direct mapping to R-nets that covers all relevant BPMN cancellation features.

3 Background

In this section, we present the concepts used in this paper. We specify the class of supported BPMN models in Sect. 3.1. In Sect. 3.2, we present reset nets.

3.1 Business Process Model and Notation

The *Business Process Model and Notation* (BPMN) [20] language is commonly used for business process modeling. We use a strict subset of modeling constructs, i.e., start/end events, tasks (allowed to represent sub-models), gateways, exception events, and some additional modeling assumptions. Consider Fig. 1, in which we depict an example BPMN model describing a simplified loan-application process, including all modeling elements considered in this paper.

The model in Fig. 1 describes that first the *Receive Application* (RA) activity is executed. After this, sub-process *S1* is invoked in which an *income check* (CI) and *asset check* (AC) are performed. The client’s income can be insufficient, triggering an *Income Insufficient* (II) exception. After such an exception occurs, the *customer record is archived*, and the process is terminated. During sub-process *S1*, at any point in time, a background check can reveal that the customer already exists in the system. In this case, the sub-process is canceled, and a *fast check* is performed. After the fast check, the global process either continues

or is canceled. For the two exceptions described, different *resolution strategies* are modeled, i.e., exception *Income Insufficient* is always followed by task t_2 (*Archive Customer*) and subsequently the entire process is terminated whereas external exception *Customer Exists* is always followed by task t_3 (*Fast Check*), after which we either terminate the process, or, we continue to process at gate g_2 . Observe that event *Customer Exists* can occur at any point in time, whereas the end event *Income Insufficient* can only happen after the task *Check Income*. We formally define BPMN models, i.e., as considered in this paper, as follows.

Definition 1 (BPMN Model). *Let Σ denote the universe of activity labels. A BPMN Model is a tuple $M=(N, T, G^\times, G^+, e^s, e^t, E^{ie}, E^{ii}, F, \ell, \xi)$, where¹:*

- N is a set of nodes,
- $T \subseteq N$ is a set of tasks,
- $G^\times \subseteq N$ is a set of exclusive gateways,
- $G^+ \subseteq N$ is a set of parallel gateways,
- $e^s \in N$ is a regular start event,
- $e^t \in N$ is a regular end event,
- $E^{ie} \subseteq N$ is a set of irregular end events,
- $E^{ii} \subseteq N$ is a set of interrupting intermediate events,
- $F \subseteq N \times N$ is the sequence flow relation,
- $\ell: T \rightarrow \Sigma$ is a partial labeling function,
- $\xi: E^{ie} \rightarrow T$ is a function assigning an intermediate event to a task, signaling an exception and thus interrupting the execution of the task.

Compared to the standard BPMN specification, we assume exactly one “regular end event” is present per (sub-)model (e.g., e_4 and e_8 in Fig. 1), i.e., representing *normative termination*. We assume that the set of tasks can be further divided in a set of labeled activities (i.e., those tasks $t \in T$ with $\ell(t) \in \Sigma$) and a set of sub-models (those tasks $t \in T$ with $\ell(t) = \perp$). If an activity refers to a sub-model, we assume that the underlying model recursively adheres to Definition 1, e.g., consider task $S1$ in Fig. 1. However, for simplicity, we do not formally define this. We assume that some hierarchy H is present that captures whether a model is contained in another model. Further, we assume that any node that is part of a sub-model is not explicitly contained in the set of nodes N of its parent model. The root model is assumed to be at hierarchy level H_0 , and any model that is contained in a task of the root model is at level H_1 . In general, a model that is contained in a task of a model at level H_i is at level H_{i+1} , e.g., in Fig. 1 all elements in sub-model $S1$ are at level H_1 , all other models are at level H_0 . Let $M_i=(N_i, T_i, G_i^\times, G_i^+, e_i^s, e_i^t, E_i^{ie}, E_i^{ii}, F_i, \ell_i, \xi_i)$ be a (sub)-BPMN model at some level $i \geq 0$ and let $M_{i+1}=(N_{i+1}, T_{i+1}, G_{i+1}^\times, G_{i+1}^+, e_{i+1}^s, e_{i+1}^t, E_{i+1}^{ie}, E_{i+1}^{ii}, F_{i+1}, \ell_{i+1}, \xi_{i+1})$ be any model at hierarchy level H_{i+1} , i.e., it is contained in one of the nodes in T_i . We assume that there exists a partial function $\delta: E_{i+1}^{ie} \rightarrow E_i^{ii}$ that associates certain *irregular end events* at hierarchy level H_{i+1} to a corresponding *intermediate event* in E_i^{ii} , e.g., consider end event e_7 in $S1$ which is linked to intermediate event e_6 . Note that, in case M_{i+1} is represented by some $t \in T_i$, then for every $e \in E_{i+1}^{ie}$ s.t. $\delta(e) \neq \perp$ we have $\xi_i(\delta(e)) = t$.

¹ We omit OR gateways, yet, the framework is easily extended with OR-support.

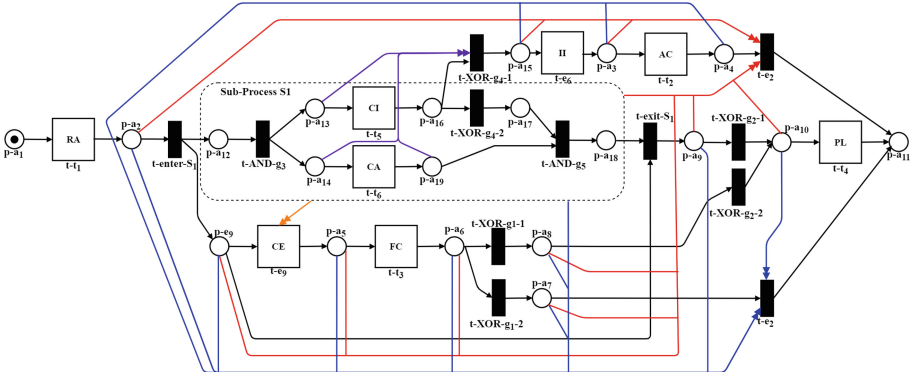


Fig. 2. Example reset-net (R-net) modeling the same behavior as the BPMN model in Fig. 1. Regular arcs are depicted as black single-headed arcs. Reset arcs are shown using double-headed arcs in different colors. Each color represents another type of cancellation behavior. A reset arc directly connecting to sub-process *S1* is assumed to empty all places in the sub-process. (Color figure online)

3.2 Reset Nets

Petri nets [23] are a mathematical model, often used for verification and automation purposes. Their mathematical definition yields Petri nets unambiguous in terms of the language they describe, i.e., as opposed to most business-oriented graphical modeling languages. A Petri net, see Fig. 2 as an example, describes a bipartite graph with two types of vertices, i.e., *places* (visually represented as circles) and *transitions* (visually represented as boxes). Places only connect to transitions and vice versa. The places of a net are used to represent the *state* of the net (referred to as the net’s *marking*), i.e., using so-called “tokens”, the transitions are used to manipulate the state. For example, the place $p-a_1$ in Fig. 2 contains *one token*. A transition in a Petri net is *enabled* iff all its “pre-places” connecting to it (places with an outgoing arc from the place to the transition) contain a token. For example, in Fig. 2, transition $t-t_1$ is enabled. An enabled transition can *fire*. When we fire an enabled transition, it consumes a token from each “pre-place” and creates a new token for each of its “post-places”. For example, if we fire transition $t-t_1$ in Fig. 2, we obtain the new marking $[p-a_2]$. Reset nets (R-nets) extend Petri nets with an additional arc type. A *reset arc*, i.e., the double-headed arcs in Fig. 2, consumes all tokens from the source place it connects to, upon firing the target transition it connects to. For example, if at least one token resides in $p-a_2$ in Fig. 2 and we fire $t-e_2$ or $t-e_3$, all tokens in $p-a_2$ are removed. Unlike regular arcs, the places associated with a reset arc can be empty to fire a transition, i.e., no tokens may exist in the places.

Definition 2 (Reset Net). Let P denote a set of places, let T denote a set of transitions ($P \cap T = \emptyset$), let $F \subseteq (P \times T) \cup (T \times P)$ denote the flow relation,

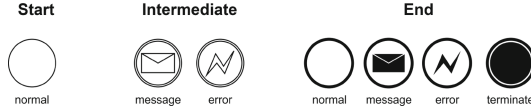


Fig. 3. The start, end, and intermediate events considered in this paper.

let $R: T \rightarrow \mathcal{P}(P)$ ² denote the reset arc relation, let Σ denote the universe of transition labels, let $\tau \notin \Sigma$ and let $\ell: T \rightarrow \Sigma \cup \{\tau\}$ denote a labeling function. A reset net N is a tuple (P, T, F, R, ℓ) .

Let $N = (P, T, F, R, \ell)$ denote a Reset net which will be called R-net for convenience. A *marking* m of N is a multiset of places, i.e., $m \in \mathcal{M}(P)$, tuple (N, m) is referred to as a *marked net*. Given marked net (N, m) , a transition $t \in T$ is enabled iff $\forall p \in P (m(p) \geq F(p, t))$. An enabled transition t in marking m can *fire*, yielding a new marking m' , where $m'(p) = 0, \forall p \in R(t)$ and $m'(p) = m(p) - F(p, t) + F(t, p), \forall p \in P \setminus R(t)$.

4 Enhanced Transformation of BPMN Models

This section presents our proposed transformation of BPMN models to R-nets. We first present additional assumptions on the input BPMN model. Subsequently, we show how we iteratively build a corresponding R-net.

4.1 Modeling Assumptions

This section presents the core assumptions of our transformation algorithm w.r.t. the input BPMN model. We discuss the different types of events and the assumptions on the modeling of exception handling mechanisms.

Events Considered. Contrary to general BPMN, we assume that a given process model has one unique *None End Event* per (sub)-process. For example, the BPMN model depicted in Fig. 1 has one unique end event (e_4) and so does the sub-process $S1$ (e_8). As such, every (sub)-process is assumed to have a clear unique *start* and *end event*, which we refer to as the *regular start/end events*. Furthermore, we assume that the start and end events have one unique outgoing, respectively incoming, arc. All other end events that are part of the BPMN model are referred to as *irregular end events*. For example, the model in Fig. 1 contains three irregular end events, i.e., e_2, e_3 and e_7 . Consider Fig. 3, in which we depict an overview of the start, end, and intermediate events considered in this paper. Observe that the events depicted in Fig. 3 are exemplary, i.e., other events (e.g., the intermediate interrupting timer) also apply to the proposed cancellation behavior. However, as such events have a merely semantic meaning (i.e., a time-based interrupt rather than a message-based interrupt), we do not consider these in detail.

² $\mathcal{P}(X)$ denotes the *power set* of set X .

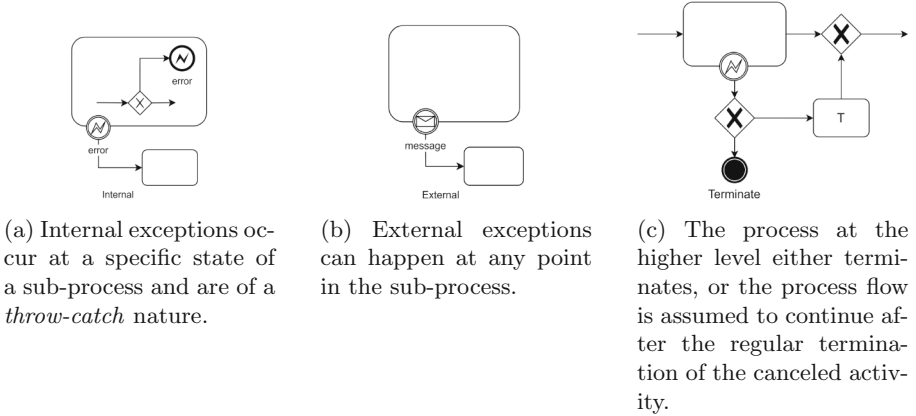


Fig. 4. The two proposed cancellation mechanisms (Figs. 4a and 4b) and subsequent resolution handling (Fig. 4c) adopted in this paper.

Exception Modeling. We assume two primary mechanisms for the modeling of exceptions and cancellation behavior, i.e., visualized in Figs. 4a and 4b. The intermediate events and the message, error, and terminate end events (cf. Fig. 3) are used for modeling exceptions and cancellation behavior. The message and error end events are used to signal an exception and are connected to an intermediate event at the boundary of the process in which they are contained. Consider Fig. 4a, which schematically represents the use of an end event and an associated “catching” intermediate event. If the end event occurs at hierarchy level H_i for $i \geq 1$, the corresponding intermediate event is located at hierarchy H_{i-1} . An unmapped intermediate event associated with some activity (possibly a sub-process) indicates that the activity it connects to can be canceled at any point in time, i.e., as visualized in Fig. 4b.

In both cases, after activating an intermediate event, a *resolution strategy* applies, i.e., depicted schematically in Fig. 4c. After observing the intermediate event related to cancellation, the process at the higher level either terminates, or the process flow is assumed to continue after the regular termination of the canceled activity. Observe that various other tasks may be modeled between the exception event and corresponding termination (as exemplified in Fig. 1). We assume that these tasks *cannot be canceled* (e.g., t_2 and t_3 in Fig. 1).

4.2 Building Reset Nets

This section presents our proposed transformation. Our approach consists of three steps, i.e., *arc mapping*, *node mapping*, and *cancellation mapping*. The first two steps are relatively straight-forward and adopted from existing BPMN transformations [9, 13, 15, 22]. The *Cancellation Mapping* step can be seen as the main contribution of this work. In the remainder, we explain each step. The final cancellation step is only applied to a model when all sub-models have been completely (recursively) transformed into an R-net.

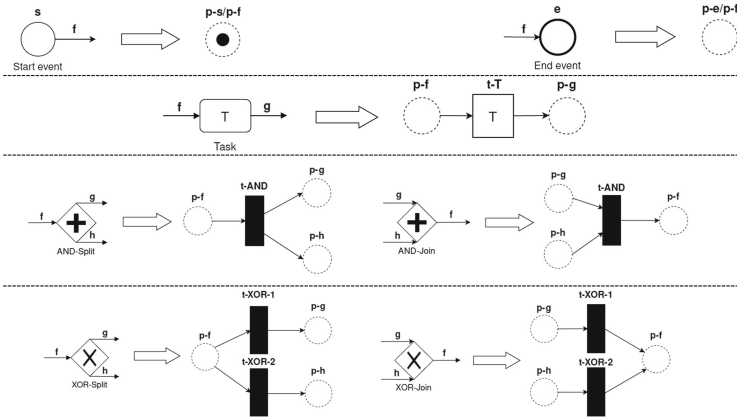


Fig. 5. Overview of node mappings of regular start/end events, tasks, AND-constructs, and XOR-constructs.

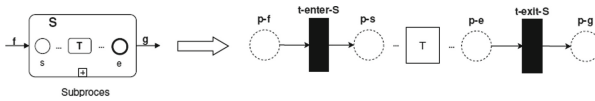


Fig. 6. Mapping procedure for sub-process tasks.

Arc Mapping. In the first step of the transformation, each arc in the BPMN model, i.e., the F component of Definition 1, is converted into a place in the resulting R-net.

Node Mapping. In the second step of the approach, we convert the nodes of the BPMN model into corresponding R-net elements. In Fig. 5, we show a schematic overview of node mappings of regular start/end events, tasks, AND-constructs, and XOR-constructs. Observe that, for the start event s , we add a token to the corresponding place $p-f$ of its unique outgoing arc. The end event is ignored. Tasks are converted to a transition for which the incoming and outgoing places are equal to the places corresponding to the incoming/outgoing arcs of the tasks in the BPMN model. AND-splits and AND-joins are converted into transitions that generate/consume a token in/from each place corresponding to an outgoing/incoming arc of the split/join. For an XOR join, a transition is created per outgoing/incoming arc of the split/join that consumes/produces a token in the place representing the outgoing/incoming arc of the split/join. The mapping of tasks that describe a sub-process is straightforward, i.e., see Fig. 6. Place $p-f$ is connected to a new invisible transition ($t\text{-enter-S}$), which is in turn connected to $p-s$. A symmetrical procedure is adopted for $p-e$ and $p-g$. Consider Fig. 7, in which we depict the transformation of the running example BPMN model after applying the first two steps (i.e., arc and node mapping).

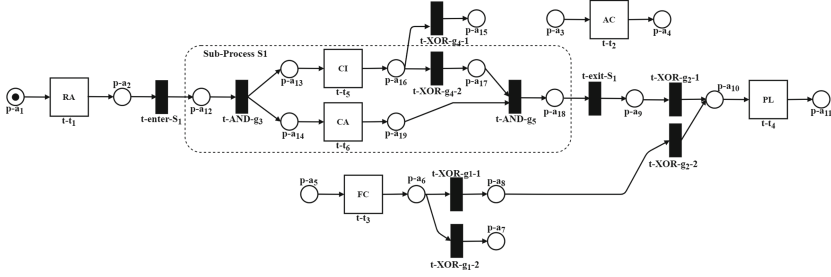


Fig. 7. Transformation of the running example BPMN model (Fig. 1) after the arc/node mapping step.

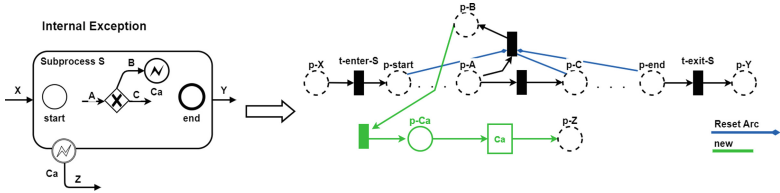


Fig. 8. Schematic visualization of internal exception handling.

Cancellation Mapping. This section presents the mappings for exception handling. Consider Fig. 8, in which we schematically depict the transformation rule for *internal exceptions*. The choice construct connected to arcs A, B, and C is already part of the partially complete net, i.e., represented by the choice construct formed by places p-A, p-B, and p-C. Firstly, we create a new net fragment, representing the intermediate error event Ca (represented by the place p-Ca and transition Ca), yielding a token in p-Z. Secondly, we attach reset nets from all places to the transition marking p-B. Finally, we connect p-B by means of a fresh invisible transition to place p-Ca. If multiple end events with label Ca exist, the previously described second and third steps can be replicated for the error event. Observe that, if p-B is marked, all places related to sub-process S are unmarked. The mapping procedure for external exception handling is depicted in Fig. 9. A place p-Ex is created that has an incoming arc from the t-*enter-S* transition and is connected to the exception transition Ex. Place p-Ex is connected by means of an outgoing arc to t-*exit-S*. If the transition labelled Ex is connected to very place in S by means of a reset arc.

Observe that the R-net in Fig. 2 is the result of applying the transformation algorithm on the running example BPMN model.

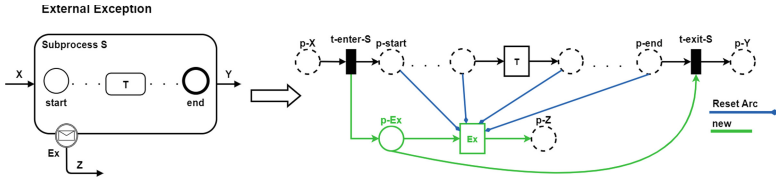


Fig. 9. Schematic visualization of external exception handling.

Table 2. Overview of the characteristics of the models used in the experiments.

Model no.	Tasks	Events	XOR gates	AND gates	Sub-processes	Exceptions
I	14	10	2	6	1	2
II	11	10	4	4	1	2
III	14	30	8	6	8	5
IV	9	20	10	4	5	3

5 Evaluation

In this section, we present the evaluation of our approach. We compare the performance of *conformance checking*, using different transformations of BPMN models with cancellation behavior. In Sect. 5.1, we present the experimental setup.³ Sect. 5.2 presents the results.

5.1 Experimental Setup

In this section, we present the experimental setup of our experiments. We describe the process adopted to generate input data, which is divided into two steps, i.e., *model construction* and corresponding *event data generation*. Additionally, we describe an alternative *transformation* of the BPMN models created, which we use to compare our approach against. Finally, we briefly provide details on the conformance checking techniques studied.

Model Construction. None of the existing work on generating process models, e.g., Burattin et al. [7], propose a BPMN generator that includes support for the (generic) addition of cancellation behavior. Hence, we design four BPMN models of varying sizes with different control-flow behavior and cancellation behavior. An overview of the characteristics of these models is presented in Table 2.

³ All models (i.e., both designed and obtained by means of transformation), event data generated and computational results, are available via https://drive.google.com/drive/folders/10Q11FfRu_Lf9kA1moR2gikQc9HAwnsNv?usp=sharing. The code used in the experiments is available via <https://github.com/require-gio/pm4py-resetnet>.

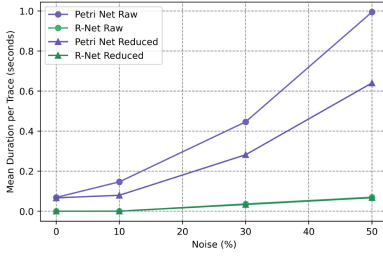
Event Data Generation. To generate data, as a first step, a noise-free language is generated exhaustively for each BPMN model. For models with looping behavior, the language is trimmed to have a maximum trace length of 25. Within the language generation, the assumed behavior of (potentially ambiguous) cancellation features is as follows. (i) An external exception event may fire even after the last enabled task has been executed inside a sub-process, and (ii) Tasks parallel to error or terminate end events may fire whenever enabled, even after any preceding task of the end event was executed. We add artificial noise to the generated languages. For each language, we randomly select 200 distinct *sub-logs* containing 15 traces, yielding a total of 3000 traces. Given a *noise probability* (0%, 10%, 30%, 50%), we iterate over each event of each trace and apply a *noise operation* with the corresponding probability. The noise operation is either an addition, substitution, or removal operation. The concrete choice of which operation to apply is made with $\frac{1}{3}$ for each operation type.

Transformations Considered. Aside from the proposed transformation in this paper, another transformation, i.e., based on Dijkman et al. [12] (referred to as *Dijkman transformation*), is considered. The *Dijkman transformation* maps BPMN models with cancellation behavior to regular Petri nets.⁴ However, the approach requires slight changes to ensure that it can transform the BPMN models in a language-equivalent manner. *Dijkman* considers intermediate events inside sub-processes as triggers for attached internal exception events. Our mapping, however, assumes that error end events are responsible for triggering internal exceptions. The input models are, therefore, manually adjusted for the *Dijkman transformation* such that error end events are exchanged with intermediate events. Terminate end events behave similarly to internal exceptions with the difference that, instead of triggering an exception event, they provide a direct path to the final state of the sub-process they are inside. Hence, the terminate events are manually replaced by intermediate events in the input models and then mapped the same way as internal exceptions, however, without an exception flow mapping.

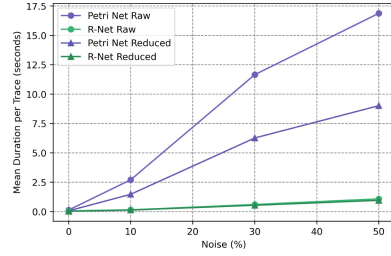
Conformance Checking. In our experiments, we consider how well the transformed models perform when applying *conformance checking* on the models. To this end, we propose to compute *optimal alignments* as conformance checking artifacts. Computing alignments requires a Petri/R-net and an event log. We assess the A^* (marking equation) implementation of alignment computation [2].⁵ During the experiments, several performance statistics are recorded: mean computation times, queued states, visited states, traversed arcs, and the number of linear programs solved. The machine used for the experiments comprises an

⁴ As there is no executable implementation available of the *Dijkman transformation*, we re-implemented the approach.

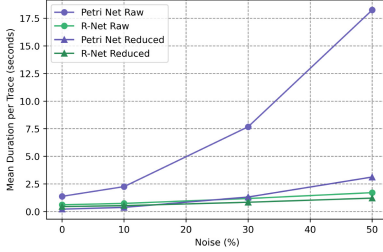
⁵ Note that the A^* variant for reset/inhibitor nets has been implemented in `python`, extending the `pm4py` framework [5].



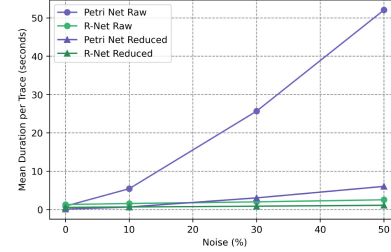
(a) Model I



(b) Model II



(c) Model III



(d) Model IV

Fig. 10. Average alignment computation time per trace of our transformation (“R-Net”) significantly outperforms the *Dijkman transformation* (“Petri net”).

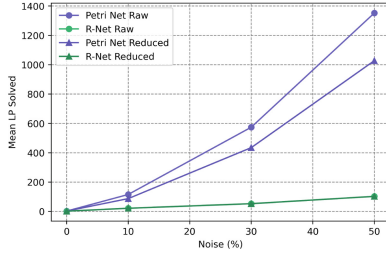
AMD Ryzen 7 2700X 8-core 4.00 GHz processor, 32 GB RAM and a Windows 10 operating system.

5.2 Results

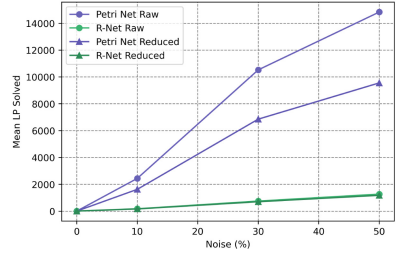
This section presents the results of our experiments, i.e., as described in Sect. 5.1. We discuss general time efficiency and search efficiency of computing alignments using the R-nets obtained by using our proposed approach and compare it with the *Dijkman transformation*. As both algorithms render many invisible transitions, we apply reduction rules on the obtained R/Petri-nets to reduce the number of invisible transitions. We use the reduction rules described in [25], yet, we only apply the non-exponential rules.

The mean computation times (in seconds) for the alignment calculation are visualized for each noise value and applied algorithm in Fig. 10. We observe that our transformation (R-Net in the figure) significantly outperforms the *Dijkman transformation* (Petri net in the figure). The model reduction has little influence on our transformation; however, it does have a significant positive effect on the time performance of the *Dijkman transformation*.

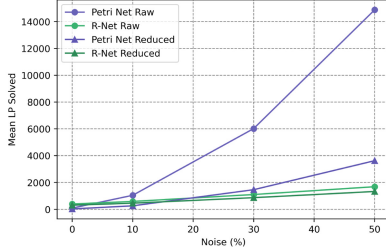
Since the *Dijkman transformation* generates more *invisible transitions* in the resulting Petri net (compared to the R-nets obtained by our transformation), we expect the underlying search efficiency to be better for the models based on



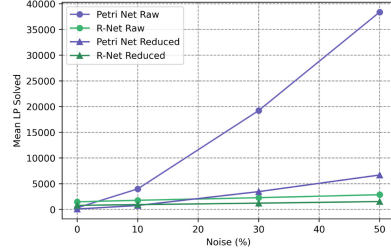
(a) Model I



(b) Model II



(c) Model III



(d) Model IV

Fig. 11. Average number of LPs solved per trace of our transformation (“R-Net”) significantly outperforms the *Dijkman transformation* (“Petri net”).

our transformation. Hence, to further investigate the underlying effect of the observed results, we consider different experimental statistics related to search efficiency. Since the A^* approach uses Linear Programming (LP) internally (i.e., as a heuristic for the search), we depict the average number of LPs solved for each model/noise combination in Fig. 11. All four plots follow the same shape as the computation time results presented in Fig. 10. This similarity indeed hints strongly at a more efficient state-space traversal of the A^* algorithm. Other metrics investigated, i.e., the number of queued states visited states, and traversed arcs per search, yield the same insights. As such, we conclude that the models obtained by our proposed transformation allow for significant efficiency in the state-space traversal of A^* -based alignment calculation.

5.3 Threats to Validity

We acknowledge that the number of models assessed in the evaluation is limited. At the same time, for all four models, we observe a similar pattern for increasing levels of noise. As such, we assume that the observed trends are also to be observed when conducting the experiments on a larger scale. We acknowledge that the transformation performed on the models used with the *Dijkman transformation* may slightly impact the performance. However, since the core problem of the *Dijkman transformation* is the excessive amount of invisible transitions in the model, we assume this effect to be negligible.

6 Conclusion

The possibility of canceling a process's main behavior is a natural phenomenon. Whereas BPMN provides rich support for cancellation features, there has been a lack of transformation algorithms that can convert said models to an alternative formal representation for further analysis. In this work, a structured mapping from BPMN to reset nets is proposed, which serves as the underlying execution model for applying advanced analytical algorithms. The mapping of cancellation features has been defined in detail and is the first to include *terminate end events* and *internal exceptions triggered by end events*. The evaluation shows that the proposed approach produces models that allow more efficient computation of conformance checking artifacts.

Future Work. We plan to extend the work proposed as follows. We aim to extend the approach to support a more extensive set of BPMN objects, i.e., there are many additional event types as well as gateways whose control flow behavior does not significantly differ from the ones considered in the presented work, e.g., interrupting escalation events, event gateways, timeouts, etc. We further aim to formally characterize classes of BPMN models that are supported by the proposed approach. Additionally, we aim to perform experiments on a larger scale. As a corresponding prerequisite, an additional interesting avenue for future work is the automated generation of BPMN models with cancellation features.

References

1. van der Aalst, W.M.P., Hirsenschall, A., Verbeek, H.M.W.: An alternative way to analyze workflow graphs. In: Pidduck, A.B., Ozsu, M.T., Mylopoulos, J., Woo, C.C. (eds.) CAiSE 2002. LNCS, vol. 2348, pp. 535–552. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47961-9_37
2. Adriansyah, A., van Dongen, B.F., van der Aalst, W.M.: Memory-efficient alignment of observed and modeled behavior. *BPM Cent. Rep.* **3**, 1–44 (2013)
3. Arbab, F.: Reo: a channel-based coordination model for component composition. *Math. Struct. Comput. Sci.* **14**(3), 329–366 (2004)
4. Arbab, F., Kokash, N., Meng, S.: Towards using Reo for compliance-aware business process modeling. In: Margaria, T., Steffen, B. (eds.) ISoLA 2008. CCIS, vol. 17, pp. 108–123. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88479-8_9
5. Berti, A., van Zelst, S.J., van der Aalst, W.M.P.: Process mining for python (PM4Py): bridging the gap between process-and data science. In: ICPM Demo Track 2019, Aachen, Germany, 24–26 June 2019 (2019)
6. Boonyawat, S., Vatanawood, W.: Transforming YAWL workflows with time constraints to generalized stochastic Petri nets. In: 3rd International Conference on Software and e-Business (2019)
7. Burattin, A.: PLG2: multiperspective process randomization with online and offline simulations. In: BPM Demo Track 2016, Rio de Janeiro, Brazil, 21 September 2016. CEUR Workshop Proceedings, vol. 1789. CEUR-WS.org (2016)

8. Carmona, J., van Dongen, B.F., Solti, A., Weidlich, M.: Conformance checking - relating processes and models. Springer (2018). <https://doi.org/10.1007/978-3-319-99414-7>
9. Dechsupa, C., Vatanawood, W., Thongtak, A.: Hierarchical verification for the BPMN design model using state space analysis. *IEEE Access* **7**, 16795–16815 (2019)
10. Decker, G., Dijkman, R., Dumas, M., García-Bañuelos, L.: Transforming BPMN diagrams into YAWL nets. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) *BPM 2008*. LNCS, vol. 5240, pp. 386–389. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85758-7_30
11. Decker, G., Dijkman, R.M., Dumas, M., García-Bañuelos, L.: The business process modeling notation. In: *Modern Business Process Automation - YAWL and its Support Environment*. Springer (2010)
12. Dijkman, R.M., Dumas, M., Ouyang, C.: Formal semantics and analysis of BPMN process models using Petri nets. Queensland University of Technology, Technical report (2007)
13. Dijkman, R., Van Gorp, P.: BPMN 2.0 execution semantics formalized as graph rewrite rules. In: Mendling, J., Weidlich, M., Weske, M. (eds.) *BPMN 2010*. LNBP, vol. 67, pp. 16–30. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16298-5_4
14. Dufourd, C., Finkel, A., Schnoebelen, P.: Reset nets between decidability and undecidability. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) *ICALP 1998*. LNCS, vol. 1443, pp. 103–115. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0055044>
15. Kalenkova, A.A., van der Aalst, W.M.P., Lomazova, I.A., Rubin, V.A.: Process mining using BPMN: relating event logs and process models. *Softw. Syst. Model.* **16**(4), 1019–1048 (2015). <https://doi.org/10.1007/s10270-015-0502-0>
16. Kheldoun, A., Barkaoui, K., Ioualalen, M.: Specification and verification of complex business processes - a high-level petri net-based approach. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) *BPM 2015*. LNCS, vol. 9253, pp. 55–71. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_4
17. Kherbouche, O.M., Ahmad, A., Basson, H.: Using model checking to control the structural errors in BPMN models. In: *RCIS 2013*, Paris, France, 29–31 May 2013. *IEEE* (2013)
18. Lohmann, N., Verbeek, E., Dijkman, R.: Petri net transformations for business processes – a survey. In: Jensen, K., van der Aalst, W.M.P. (eds.) *Transactions on Petri Nets and Other Models of Concurrency II*. LNCS, vol. 5460, pp. 46–63. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00899-3_3
19. Mili, H., Tremblay, G., Jaoude, G.B., Lefebvre, E., Elabed, L., El-Boussaidi, G.: Business process modeling languages: sorting through the alphabet soup. *ACM Comput. Surv.* **43**(1), 4:1–4:56 (2010). <https://doi.org/10.1145/1824795.1824799>
20. OMG (2021) <https://www.omg.org/spec/BPMN/2.0/About-BPMN/>. Accessed 30 Nov 2021
21. Ou-Yang, C., Lin, Y.: BPMN-based business process model feasibility analysis: a petri net approach. *Int. J. Prod. Res.* **46**(14), 3763–3781 (2008)
22. Raedts, I., Petkovic, M., Usenko, Y.S., van der Werf, J.M.E.M., Groote, J.F., Somers, L.J.: Transformation of BPMN models for behaviour analysis. In: *MSVVEIS-2007*, Funchal, Madeira, Portugal, June 2007. INSTICC PRESS (2007)
23. Reisig, W.: *Petri Nets: An Introduction*, EATCS Monographs on Theoretical Computer Science, vol. 4. Springer, Heidelberg (1985). <https://doi.org/10.1007/978-3-642-69968-9>

24. Terayawan, S., Vatanawood, W.: Transforming control-flow patterns of YAWL to Petri nets. In: International Communication Engineering and Cloud Computing Conference (2019)
25. Verbeek, H.M.W., Wynn, M.T., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Reduction rules for reset/inhibitor nets. *J. Comput. Syst. Sci.* **76**(2), 125–143 (2010)
26. Ye, J., Song, W.: Transformation of BPMN diagrams to YAWL nets. *J. Softw.* **5**(4) (2010)
27. Ye, J., Sun, S., Song, W., Wen, L.: Formal semantics of BPMN process models using YAWL. In: 2008 Second International Symposium on Intelligent Information Technology Application, vol. 2. IEEE (2008)



Next-Activity Prediction for Non-stationary Processes with Unseen Data Variability

Amolkirat Singh Mangat¹ and Stefanie Rinderle-Ma²

¹ Research Group Workflow Systems and Technology, Faculty of Computer Science,
University of Vienna, Vienna, Austria

`amolkirat.singh.mangat@univie.ac.at`

² Chair of Information Systems and Business Process Management,
Department of Informatics, Technical University of Munich, Garching, Germany

`stefanie.rinderle-ma@tum.de`

Abstract. Predictive Process Monitoring (PPM) enables organizations to predict future states of ongoing process instances such as the remaining time, the outcome, or the next activity. A process in this context represents a coordinated set of activities that are enacted by a process engine in a specific order. The underlying source of data for PPM are event logs (ex post) or event streams (runtime) emitted for each activity. Although plenty of methods have been proposed to leverage event logs/streams to build prediction models, most works focus on stationary processes, i.e., the methods assume the range of data variability encountered in the event log/stream to remain the same over time. Unfortunately, this is not always the case as deviations from the expected process behaviour might occur quite frequently and updating prediction models becomes inevitable eventually. In this paper we investigate non-stationary processes, i.e., the impact of unseen data variability in event streams on prediction models from a structural and behavioural point of view. Strategies and methods are proposed to incorporate unknown data variability and to update recurrent neural network based models continuously in order to accommodate changing process behaviour. The approach is prototypically implemented and evaluated based on real-world data sets.

Keywords: Event streams · Predictive process monitoring · Data variability · Non-stationary prediction

1 Introduction

Predicting future states of processes is a highly desirable ability as it empowers organizations to transition from reactive to proactive measures [4]. Predictive Process Monitoring (PPM) exploits traces of event data emitted during the executions of processes to enable evidence based forecasting of the remaining time, the outcome, or the next activity of ongoing processes [7].

However, most of the proposed PPM methods have one critical aspect in common: they are heavily based on the assumption that processes are stationary [9, 11]. Stationary processes refer to processes that remain unchanged over time. This enables one to make safe assumptions about the underlying data variability and permits to ignore potential issues when unseen data variability is encountered.

In practice, processes evolve over time, especially when humans are involved. Just as organizations undergo changes and require an overhaul of their processes, so can human behavior change the way to carry out tasks [10]. In both cases, this can lead to inevitable changes in the data variability. These changes can be observed in the event traces which can be encountered in two forms, i.e., as event logs (ex-post) or as event streams emitted and stored during runtime (online). Each event is expected to carry information such as the identity of a particular process instance, an activity label, and potentially context data collected and/or generated by an activity. Changes that could be observed in the event traces include the emergence of new activities, rearranged orders of activities, and new values of context data.

Moreover, in an online setting, we do not have the privilege of foresight; we cannot know all possible forms of data variability upfront that our model may encounter in future. This holds particularly true when we assume non-stationary processes; change is expected, but *when* a change will occur, *what* changes may occur, and *which* impact the changes may have remains unknown. This applies at least until the point in time unseen data variability is encountered. Reliable predictions require complete data. Incomplete data may involuntarily introduce faulty behavior to a prediction model leading to imprecise prediction results. Under these premises and the restriction that event data acquired over time is the only available source of evidence for the process behavior, predicting the next-activity is a challenging task. The goal is to enable robust, evidence-based next-activity predictions when we can solely rely on (incomplete) event data.

This paper addresses these challenges through the overarching question of

How to predict the next activity for an observed event assuming non-stationary processes in an online setting, i.e., during runtime?

We tackle this question based on the following contributions:

- We discuss potential challenges and pitfalls when dealing with unseen future data that will be utilized as input to predict the next activity.
- We discuss the requirements to build evidence-based robust prediction models under an online setting with non-stationary processes
- We propose and analyze greedy strategies for updating the prediction model and handling unseen event attribute values that serve as input for the prediction model.
- The strategies are prototypically implemented and applied to several real-world data sets. In addition to the feasibility of the strategies, the evaluation results show that considering more historic data does not necessarily lead to

significantly better prediction results. This insight is useful for finding the sweet spot between effort and output quality.

Section 2 discusses related work. Section 3 sets out challenges and requirements and Sect. 4 presents the approach for next activity prediction in an online setting. Section 5 provides an experimental evaluation of the approach. Section 6 discusses the approach and concludes the paper.

2 Related Work

The majority of predictive process monitoring approaches operates in an offline manner, i.e., on process execution logs [12]. For online settings, predictive process monitoring literature distinguishes between dynamic and static prediction approaches. A *static approach* is proposed by [9]. In detail, an annotated transition system processes the observed process sequence and selected data attributes to derive a prediction. Activities that are unknown at runtime are dropped from the input before passing them to the prediction model. *Dynamic approaches* are presented in [6, 8, 12]. [6, 12] investigate various prediction model update strategies based on decision trees in order to handle unseen process behavior. [8] investigate incremental learning strategies for neural networks by reusing existing prediction models and adapting them with newly observed training data. Our approach differs from [6, 8, 12] in several ways: i) we include more if not all available attributes in addition to the activity label and the timestamp; ii) we consider the full (partial) traces as input for our prediction model in contrast to limiting the input to a fixed number of the most recent events; iii) we use recurrent neural networks; iv) we do not assume any knowledge about future data not seen by the prediction model at training time, and v) we utilize different model update strategies. Moreover [12] investigate update strategies over selected time periods and [8] investigate strategies on a monthly basis. By contrast, the approach at hand, investigates update strategies on a daily basis.

3 Challenges and Requirements

Our focus is on process-oriented data. Let P be a process model that comprises a set of activities a_i ($i = 1, \dots, n$) which are executed in a particularly coordinated fashion. Each activity can be augmented by any number of attributes which carry information specific to this activity. At runtime, process instances are created and executed based on P . The progress of an ongoing process instance σ_j is reported by emitted events e_i that carry the *case identifier* of a process to which the event belongs, a *timestamp*, and activity specific data that includes the label of an activity and optionally activity attributes together with their values. With the help of the case identifier of incoming events we can distinguish between different process instances and thus determine an event's affiliation to one particular process instance. The timestamps enable us to reconstruct the order in which activities have been carried out. Let c represent a case identifier.

Then an ordered sequence of events e_i^c sharing the same case identifier is referred to as *trace* $\pi^c = \langle e_1^c, e_2^c, \dots, e_n^c \rangle$. Since an event's purpose is characterized by its activity label, we use the terms *event* and *activity* interchangeably from here on. For the next-activity prediction task, events and traces are the fundamental resources for evidence-based predictions.

3.1 Data Variability

In the following, we discuss at which levels data variability might occur in the context of processes, i.e., through changes on the *meta* and *value* level.

- **Meta level** changes include changes to the structure of an event. The three possible scenarios include (*M1*) the removal of existing attributes, (*M2*) the inclusion of new attributes, and (*M3*) the alteration of the reference name of an existing attribute.
- **Value level:** We distinguish between two data types for attributes, i.e., categorical and continuous values. Categorical values represent values limited to a finite set of numeric or non-numeric values (e.g., the name or identifier of an activity or resource). Continuous values represent numeric values that can potentially take on any value from an infinite set of numeric values (e.g., a temperature measurement obtained from a sensor). Value level changes refer to deviations of attribute values from values observed in the past.

Deviations may include the absence or change in the frequencies of known values, and the appearance of previously unseen values. In literature such deviations are also referred to as *concept drifts* [5], including recurring or seasonal drifts, incremental drifts reflecting small changes, sudden drifts reflecting completely new process behavior, and gradual drifts reflecting gradually deviating process behavior.

The implications of meta/structural and value/attribute level deviations are manifold. Structural ad-hoc changes e.g. by the exclusion of an expected attribute can lead to a system malfunctioning if the system is not designed to dynamically adapt to change. Value level changes may similarly impair a system's behavior when no mechanisms are in place to handle unseen data.

In this paper, we focus on value level data variability and assume no meta level changes occur. We assume that we only have access to event logs or event streams to derive the information required to build a prediction model.

3.2 Challenges of Evidence Based Predictions with Unseen Activities and Sequences

An event in the context of process execution is primarily characterized by its type, i.e., *which* activity has occurred, and *when* it has occurred. Not knowing all types of events and the order in which the events could potentially occur removes the ability to prepare a model beforehand that can handle future unseen data.

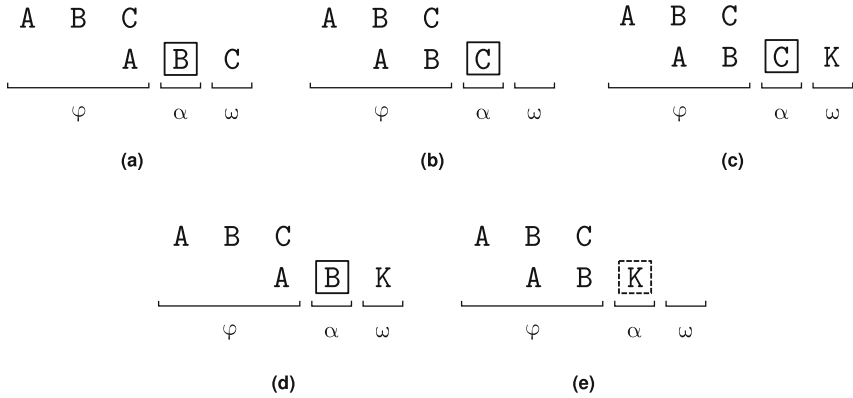


Fig. 1. Challenges when predicting next activities with unseen events. φ represents the set of past known events, α the set of currently observed events for which we want to predict the next activity and ω the set of future unknown events. The box with the solid border around the events under α indicates for which event we can attempt to predict whereas the dashed box indicates we cannot make a prediction as the event has been observed for the first time.

Figure 1 illustrates scenarios where not all event types that could occur and the order in which those event types could occur are known upfront.

For scenarios (a) – (e), we assume that the sequence of activities $A > B > C$ has been observed, where $>$ denotes the order in which the activities have been observed. In the above case activity B follows A and activity C follows B . The time frame which includes all observed events is marked by φ . We assume that all events in φ have been completed. The time frame α represents the currently observed events. For them the aim is to predict the next activity, while also taking into consideration predecessors of the events. Furthermore the prediction process considers all observed events in φ . Lastly the time frame ω represents the future and thus events that will follow the events observed in the present, but which are unknown during α .

Next we will address each of the scenarios (a) – (e) and their potential pitfalls. In scenario (a), we can predict the next activity for the observed sub sequence $A > B$. Based on the past observations, C would be the only logical candidate and correct choice with regards to ω . In scenario (b), the logical candidate would be to predict the termination of the process, which would coincide with the past as no follow-up event after C was observed. Scenario (c) depicts a similar starting setup as in Scenario (b) except that in ω event K will follow. Unaware of the future and not having observed this sequence pattern in the past, the logical candidate for the next activity prediction would be the termination of the sequence, which is incorrect. Scenario (d) depicts a similar initial setup as in scenario (c), where again based on the past observation the most likely candidate to follow B be is C. However, this again would be an incorrect prediction, since a pattern that includes K was not observed in φ . Finally Scenario (e) illustrates

the case where an event with an unseen activity is encountered. Having no prior knowledge about the unseen activity and thus consequently about the observed partial trace, making an prediction with incomplete adds a certain risk to the validity of the prediction result.

3.3 Uncertainty of the Final Activity of a Trace

Another challenging task is to predict the end of a sequence. Besides predicting the next activity, predicting the end of a process is vital and necessary in order to establish a proper termination point for a sequence of activities. We can only predict the end of an ongoing process if we know what the possible termination states are. For our scenario, where only event logs and streams are available, we differentiate between the following three cases, ordered from high to low confidence with regards to the knowledge of the possible termination states. Let O represent a set of observed activities and T a set of activities observed at the end of sequences to which we refer as *termination activities*.

- C1 There exists a subset of termination activities that only appears at the end of sequences and is never preceded by any other activity including the termination activities.

$$a_1 \rightarrow \dots \rightarrow a_n \rightarrow t \quad a_i \in O, t \in T, O \neq T \quad (1)$$

- C2 There exists a subset of termination activities that can be preceded and superseded by any activity including termination activities. The termination of a sequence however only occurs after a termination candidate.

$$a_1 \rightarrow \dots \rightarrow a_n \rightarrow t \quad a_i \in O, t \in T, T \subset O \quad (2)$$

- C3 Every activity can occur at any place in the sequence and can be a candidate for the sequence termination.

$$a_1 \rightarrow \dots \rightarrow a_n \rightarrow t \quad a_i \in O, t \in T, T \subseteq O \quad (3)$$

As mentioned above each event carries a case identifier and a timestamp. With this information we can determine to which trace an event belongs and in which order the events have been observed and thus presumably in which order activities of a trace have been carried out. Thus the event with the most recent timestamp of a trace is considered as the final event that occurred before the trace terminated.

Under this premise, *C1* is the simplest case as we are ensured that certain activities only appear at the end of a sequence. This should lead to prediction models with the most accurate prediction results as there is a clearly distinguishable set of activities which solely appear at the end of sequences. *C2* faces a more complicated challenge: after we have reached a termination candidate we need to answer the question: *Do we predict the end or do we predict another follow-up activity?* While Case *C2* has the confidence that predicting the end is

a valid option, $C3$ cannot provide any assurances in this regard as any activity can be a candidate for the last activity of a sequence.

Case $C3$ corresponds to total randomness where over time constantly new activities are added or even dropped and new sequences appear such that no patterns can be established. This makes it difficult or impossible to make any reliable predictions. $C2$ is a more likely scenario where occasionally changes might emerge in addition to established patterns which gradually may change over time. In this paper we assume Case $C2$.

4 Approach

This section presents data encoding strategies for unseen data variability and an approach for updating the prediction model for next activity prediction with unseen data variability.

4.1 Data Encoding with Unseen Data Variability

Data encoding is the bridge that enables prediction models to utilize raw event logs and stream data. In other words, the occurring event data must be transformed into the structure required by the model. Typically a prediction method has certain requirements for the data encoding whereby certain limitations can be imposed on how we can represent input data for models.

As mentioned in Sect. 3.1, we differentiate between categorical and continuous values. The traditional and widely used approach is to encode categorical features using dummy variables, where each variable for a binary feature can either take the value 0 or 1 to encode the absence or presence of a category respectively. For features with $K > 2$ categories this technique is extended to *1 of K* or *one-hot* feature encoding [2]. Here a feature is represented by a binary vector with K variables, where all the variables take the value 0, while the variable that represents the category we intend to encode takes the value 1. Assuming a set of three categories $\{Q, R, S\}$ we then can uniquely represent each category by the vectors $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ respectively.

However, one-hot encoded features can lead to sparse representations, especially if the number of categories increases. A more compact variation for binary feature encoding is achieved by applying an *ordinal encoding* scheme [2]. In this case, each category of a feature is mapped to a cardinal number. Then the mapped number of the category is represented in the base-2 numeral system to encode a binary vector. Assuming that the categories $\{Q, R, S\}$ are indexed from 1 to 3 in ascending order, we then can represent the categories with a vector of size two as $(0, 1)$, $(1, 0)$ and $(1, 1)$ respectively.

Continuous values can technically be encoded as is. However this typically depends on the prediction technique applied. The prediction technique might be subject to, e.g., only permitting integers or requiring normalization in order to achieve an overall homogeneous representation with respect to other attributes for the input data. Alternatively, one can encode a continuous value similarly to

the approach for categorical values by organizing the values into bins. The time elapse since the start of a process case, for example, can be modeled into bins of a day, a week or a month and so on.

For unseen data the challenge is that we do not have a mapping based on previously observed data that translates it into a representation a prediction model understands. To tackle this challenge, we propose two strategies, namely using a *void category* and introducing additional *reserve capacity* in the prediction model, that can be utilized with existing encoding techniques to handle unseen data variability and to enable prediction models to operate without immediate interruption and to facilitate faster updates for prediction models.

Void Category. The purpose of the *void category* strategy is to serve as a placeholder for arbitrary input unknown to us. Let V^a represent a collection of known data values for an attribute $a \in A$ and M the set of transformed and encoded values m and $m^* \notin M$ representing the void category. Then $v_i^a \rightarrow m^*$ represents the void mapping function with $v_i^a \notin V^a$. This allows to keep the structure of a prediction model as is and enable the model to keep operating with unseen data.

We define the void category as the zero vector and in addition to the known set of categories $\{Q, R, S\}$ we observe $\{T, U\}$ not known to our prediction model. Then T and U are mapped to $(0, 0, 0)$ when using the one-hot encoding or to $(0, 0)$ when using the ordinal encoding scheme.

Reserve Capacity. Another strategy is to integrate additional *reserve capacity* Ψ for every attribute of the input. Additional slots are added to the binary vector for future unseen data categories. Since the additional reserve capacity is embedded into the input structure, the prediction model must not undergo major structural changes. This enables to incrementally update existing models. For minimal alteration it is necessary to keep the order of the encoded attributes and their categories in the vector representation.

The challenge here is how to pick the additional capacity size and to decide when to further increase the capacity at runtime. Assuming the same set of known categories $\{Q, R, S\}$ for an attribute and the reserve capacity to be $\Psi = 2$ the resulting vector length for the one-hot encoding scheme is 5 (3 known categories + 2 reserve slots) and in the case of the ordinal encoding scheme the vector length is 3 (as 3 binary digits can encode 8 numbers from 0-7 which covers the overall required capacity of 5 binary numbers.) The corresponding one-hot encoded representation for Q, R, S would be $(1, 0, 0, 0, 0), (0, 1, 0, 0, 0)$ and $(0, 0, 1, 0, 0)$. A new category T could then be encoded at the next unoccupied slot $(0, 0, 0, 1, 0)$. For the ordinal encoding scheme this results in the vectors $(0, 0, 1), (0, 1, 0)$ and $(0, 1, 1)$ for Q, R, S respectively, whereas T would be represented by $(1, 0, 0)$ assuming T is indexed by 4.

For the void category strategy, which has its practical use in an online setting, it is necessary that a prediction model can continue to operate despite unseen data values. By contrast, the reserve capacity strategy aims to enable faster

prediction model update by preserving parts of the model learned from the past. Through the introduction of the additional capacity for newly observed value types, this should result in additive changes to the model and thus to faster convergence during the model training. In this paper, we focus on the void category strategy, leaving the reserve capacity strategy for future work.

4.2 Determining the Termination of a Sequence

To determine the final event of an observed trace that leads to termination, we introduce an additional attribute for each event that keeps track of the lifecycle of the trace. Existing approaches such as [1, 14], by contrast, require an additional event/activity that indicates the end of a case. The trace lifecycle attribute keeps track of the current progress of an instantiated process and is set by the underlying process execution engine. The first activity's event will carry the value 'START' whereas the final activity's event will hold the value 'END' for the trace life cycle attribute.

4.3 Online Prediction Process

The online next activity prediction approach proposed in this work is depicted in Fig. 2 and comprises three repeating sub processes. The first sub process includes activities *Read Event Stream* followed by *Aggregate/Map Events into/to Traces* π^c which processes incoming events and organizes the events into traces with the help of an event's case identifier and timestamp and persists them into a database denoted as Π . The second sub process consists of activities *Detect Data Variability* and *Trigger Prediction Model Update* which actively monitors deviations in Π and initiates the process to update a prediction model \mathcal{M}_s , both based on a model update strategy \mathcal{S} . A new model \mathcal{M}_{s+1} is built using the update function $\mathbf{h}(\Pi, \theta)$ where θ prescribes a set of rules regarding which traces of Π to consider. Examples for θ include applying a sliding window or expanding window approach. In the sliding window approach one only includes traces that are within a time frame (e.g., traces observed within the last month from today). An expanding window approach includes all future traces since a particular point in time in the past. The third sub process is composed of the activities *Query Running Traces* and *Predict Next Activities with \mathcal{M}_{s+1}* . It predicts a subsequent activity with regards to the last activity registered in a trace for all incomplete (running) traces with the most recent prediction model \mathcal{M}_{s+t} available.

4.4 Prediction Model Update Strategies

In this paper, we consider and analyze three greedy prediction model update strategies \mathcal{S}_1 – \mathcal{S}_3 illustrated in Algorithms 1–3 respectively. These strategies enable us to address changes in activities and sequences and facilitate investigating the impact of unseen activities and sequences on the prediction results.

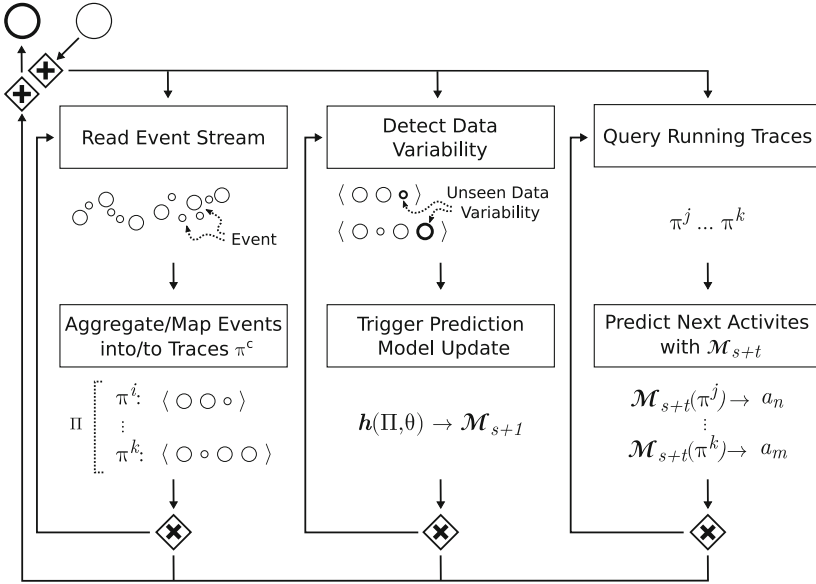


Fig. 2. Online next activity prediction approach.

The first strategy \mathcal{S}_1 described by Algorithm 1 employs a periodic update strategy where the time span between updates is controlled by δ . This strategy does not consider deviations in the observed collection of traces Π . For all strategies θ determines which part of the historical data in Π is considered for the prediction model update. \mathcal{S}_1 is suitable for stationary processes where change in the process behavior and data is not expected.

Algorithm 1. \mathcal{S}_1 : Periodic Updates

```

1: Input:  $\Pi, \theta, \delta \leftarrow$  time until next update
2:  $t \leftarrow \text{now}()$ 
3:  $\mathcal{M} \leftarrow \mathbf{h}(\Pi, \theta)$ 
4: while true do
5:   if  $t + \delta \leq \text{now}()$  then
6:      $\mathcal{M} \leftarrow \mathbf{h}(\Pi, \theta)$ 
7:      $t \leftarrow \text{now}()$ 
8:   end if
9: end while

```

In contrast to \mathcal{S}_1 , strategies \mathcal{S}_2 and \mathcal{S}_3 base their decision to trigger a model update when unseen variability in Π is detected. Both \mathcal{S}_2 and \mathcal{S}_3 ensure unseen activities and sequences respectively are incorporated into the prediction model which exposes the model to new patterns and improves the ability to make evidence based predictions. Thus these strategies are suited for non-stationary processes. \mathcal{S}_2 actively monitors for activity labels in Π that are unknown to the prediction model \mathcal{M} while \mathcal{S}_3 actively monitors for new sequences, i.e. new

Algorithm 2. \mathcal{S}_2 : Update on new Activity

```

1: Input:  $\Pi, \theta, \Omega$ 
2:  $\mathcal{M} \leftarrow \mathbf{h}(\Pi, \theta)$ 
3:  $t \leftarrow \infty$ 
4: while true do
5:   if  $\Pi$  has unseen activities then
6:      $t \leftarrow$  schedule next update with policy  $\Omega$ 
7:   end if
8:   if  $t \leq$  now() then
9:      $\mathcal{M} \leftarrow \mathbf{h}(\Pi, \theta)$ 
10:     $t \leftarrow \infty$ 
11:   end if
12: end while

```

Algorithm 3. \mathcal{S}_3 : Update on new Sequences

```

1: Input:  $\Pi, \theta, \Omega$ 
2:  $\mathcal{M} \leftarrow \mathbf{h}(\Pi, \theta)$ 
3:  $t \leftarrow \infty$ 
4: while true do
5:   if  $\Pi$  has unseen sequences then
6:      $t \leftarrow$  schedule next update with policy  $\Omega$ 
7:   end if
8:   if  $t \leq$  now() then
9:      $\mathcal{M} \leftarrow \mathbf{h}(\Pi, \theta)$ 
10:     $t \leftarrow \infty$ 
11:   end if
12: end while

```

arrangement of activity sequences. \mathcal{S}_2 and \mathcal{S}_3 require an update policy Ω which determines *when* to update a model if new activities or activity sequences are encountered. Such a policy could, for example, dictate to update a model immediately, after at least K number of new occurrences have been observed since the last update, or, at the end of a day.

5 Experiments

We evaluate model update strategies \mathcal{S}_1 – \mathcal{S}_3 (cf. Section 4.4) on several data sets using an expanding window approach in combination with the proposed void category encoding for unseen data variability. We predict the next activities on a daily basis in order to simulate an online setting. The underlying assumption is that strategies leading to more frequent updates are expected to perform better as more data is available during the training of the prediction model. However, how much improvement can be expected and at what cost the improvement is acquired is unknown and thus the subject of investigation. In the following, we describe the experimental setup in more detail and present the results of the evaluation. The source code for the experiments is available on GitHub¹.

5.1 Online Prediction Simulation and Model Update

For the periodic update strategy \mathcal{S}_1 we use a delay δ of 24 hours and infinity. The latter case equates to training a prediction model once and is utilized as our baseline. We predict the next activity for all events observed on the next day outside the past window. The past window considers all observed events prior to the prediction day. Once all the next-activity predictions have been made for the events in the prediction window, the past window is expanded by including the events in the prediction window. These steps are repeated until all events of the data set have been processed. For the update strategies \mathcal{S}_2 and \mathcal{S}_3 we use a similar approach, except the past window is expanded only after a criterion (i.e., unseen activities or activity sequences) is met and after all events in the

¹ <https://github.com/auroeur/kronos>.

prediction day window have been processed. If the activity of a newly observed event is unknown, no prediction for the next activity is made. This applies to all update strategies. A prediction model update is only done after the window has been expanded.

5.2 Datasets

Helpdesk Dataset. The Helpdesk² data set comprises of event logs of a ticketing management process of an Italian software firm. Overall the data set includes 20777 events that are part of 4580 distinct process instances. The data was collected between 2010-01-13 and 2014-01-03. The ticketing management process comprises of 14 distinct activities. Each activity includes up to 12 attributes as additional context data such as the resource, customer and severity of the issue.

BPI 2012. The BPI2012³ data set has been released as part of the Business Processing Intelligence Challenge (BPIC) 2012. This data set is a collection of event logs from a Dutch financial institute concerning processes for loan applications. The log consists of 262220 events distributed among 13087 cases. Each process instance comprises of three sub-processes that include automatic and manual tasks carried out by humans. The data set includes process cases from 2011-10-01 till 2012-03-14 with 24 unique activities and two attributes including the resources handling the applications and the applied loan amount.

Sepsis. The Sepsis⁴ data set comprises of anonymized real-life event logs collected over a time span of almost two years that tracks the pathway in a hospital of patients with sepsis. The log consists of 15214 events distributed among 1050 cases. Overall the dataset consists of 16 distinct activities which can include up to 30 attributes.

5.3 Data Pre-processing

For all data sets, we omit the life cycle information by discarding events that track the start and intermediate states of an activity. We only consider events marked as complete. As addressed in Sect. 4.2, we in addition augment events with information that signals if an event marks the first, an intermediate or the final activity of a trace. Furthermore, we consider all non-redundant trace and event level attributes found in the data sets in addition to the activity label and timestamp attributes as features for the prediction models.

For every data set, we split the event logs by taking the first 10% of the events to build the initial prediction model and we treat the remaining events as the source for future incoming events. We include all events that occur on the same day as the last event of the initial split.

² <https://doi.org/10.4121/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb>.

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

⁴ <https://doi.org/10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460>.

For the prediction model training, the already observed events are aggregated into traces with regards to matching case identity numbers. From these (in-)complete traces we incrementally generate $n - 1$ partial traces/sequences of events where n represents the number of observed events belonging to the same trace. The smallest trace has size 1. The order of the events in a trace is determined by the timestamp of the event. As the prediction label we use the event activity that would naturally follow the incomplete partial trace.

5.4 Event Data Encoding

We encode categorical and continuous values of a trace as a binary vector using the base-2 numeral system. Unknown attribute values are encoded using the *void category* strategy for which we use the zero vector encoding.

In addition we include an attribute ‘delay’ for every event as a categorical feature which describes the time passed since the first event of the joint trace. The attribute ‘delay’ is subdivided into the following categories: an hour, four hours, eight hours, a week, 2 weeks, 3 weeks, a month, 2 months, 3 months and beyond 3 months.

5.5 LSTM Neural Networks and Model Training

As the prediction method we use a variation of Long-Short-Term Memory (LSTM) [3] recurrent neural network proposed in [13]. LSTM in general have shown to be well-suited for sequential data such as process events as demonstrated in [1, 14].

For the LSTM we use 16 hidden cells with a single layer. The output nodes of the LSTM are first fed into a batch normalization layer and then are passed onto a fully connected layer. The input and output size of the network is dynamically set during the prediction model training and depends on the data observed. As the loss function we use cross-entropy since we are dealing with a classification problem where we want to predict the label of the next activity based on the partial trace observed out of K classes. As the learning algorithm we use Adam with a step size of $1e^{-3}$ and epsilon set to $1e^{-8}$. The prediction models are implemented with PyTorch and trained on the CPU. All experiments were conducted on a Fedora 32 system with an AMD Ryzen 5 3600 6-Core CPU and 32 GB memory.

5.6 Model Selection

We split the events of the past window by first grouping them into traces with the help of the case identifier and then taking the first 80% for training and the remaining 20% for validating prediction models. The training set is used to train the prediction models and the validation set is used to determine when to stop the training procedure in order to avoid the models over-fitting the underlying data and to ensure the model is capable of generalizing for unseen

data variability. Overall we train a model at most for 16 epochs with a batch size of 4. We stop the training early if the prediction model accuracy does not improve by 1% point over the last 5 consecutive epochs. We adopt the most recent prediction model that surpassed the improvement threshold.

5.7 Results

Table 1 summarizes the results for the experiments. We report the accuracy (ACC), Matthews correlation coefficient (MCC), F1 as the weighted average of the per activity label F1 scores w.r.t to a label’s support (i.e., the number of occurrences of an activity label), precision (PR) and recall (RC) scores. These scores include the results of next-activity predictions post the initial past window. The column $\#P$ represents the number of predictions made, whereas $\#NP$ the number of times a prediction has not been made due to an unseen activity label. The column $\#Model$ depicts the number of prediction models trained after an update is triggered and $\#Time$ the cumulative time spent training models for a strategy over the entire time span of a data set.

The suffix of a data set represents the update strategy applied: $\mathcal{S}_1^{\delta=\infty}$ does not update models, $\mathcal{S}_1^{\delta=24h}$ updates models every 24h, \mathcal{S}_2 updates models when encountering unseen activities, whereas \mathcal{S}_3 triggers model updates when unseen activity sequences are detected. The bold scores represent the best scores obtained for the respective metric.

Table 1. Next-activity prediction results for the model update strategies $\mathcal{S}_1 - \mathcal{S}_3$.

Data Set- \mathcal{S}_i	ACC	MCC	F1	PR	RC	$\#NP$	$\#P$	$\#Models$	Time
Helpdesk- $\mathcal{S}_1^{\delta=\infty}$	0.837	0.794	0.818	0.845	0.837	346	17992	1	0:00:11
Helpdesk- \mathcal{S}_2	0.914	0.89	0.908	0.907	0.914	14	18324	6	0:03:14
Helpdesk- \mathcal{S}_3	0.931	0.912	0.926	0.923	0.931	14	18324	334	4:34:59
Helpdesk- $\mathcal{S}_1^{\delta=24h}$	0.928	0.908	0.923	0.919	0.928	13	18325	1253	15:14:58
BPI2012- $\mathcal{S}_1^{\delta=\infty}$	0.736	0.722	0.693	0.716	0.736	286	113691	1	0:01:02
BPI2012- \mathcal{S}_2	0.742	0.729	0.699	0.724	0.742	286	113691	1	0:01:21
BPI2012- \mathcal{S}_3	0.778	0.765	0.756	0.782	0.778	286	113691	128	14:17:45
BPI2012- $\mathcal{S}_1^{\delta=24h}$	0.777	0.764	0.754	0.781	0.777	286	113691	144	16:07:17
Sepsis- $\mathcal{S}_1^{\delta=\infty}$	0.588	0.528	0.562	0.574	0.588	8	12734	1	0:00:19
Sepsis- \mathcal{S}_2	0.615	0.559	0.593	0.607	0.615	2	12740	2	0:00:41
Sepsis- \mathcal{S}_3	0.654	0.602	0.634	0.636	0.654	2	12740	425	15:05:40
Sepsis- $\mathcal{S}_1^{\delta=24h}$	0.656	0.604	0.636	0.636	0.656	2	12740	503	19:57:03

All update strategies significantly outperform our baseline strategy $\mathcal{S}_1^{\delta=\infty}$ that does not update the prediction model. This seems to indicate that the presence of unseen data variability is not reflected in the initial starting training data. In terms of the prediction quality the results clearly suggest that more

frequent model updates lead to better prediction scores. The only exception in this case applies to the most greedy update strategy $\mathcal{S}_1^{\delta=24h}$ with periodic model updates on a daily basis, which despite significantly higher cumulative computation time at most performs as well as or in some cases even worse than the update strategy \mathcal{S}_3 that is triggered when unseen activity sequences are observed. A possible explanation for this observation is that too frequent updates impede a model from generalizing due to the model overfitting the data, especially considering that the use of an expanding window includes all observed events for training, leading to potentially complex models.

In light of the higher computational cost incurred by strategies with more frequent updates the question arises, how much additional computational effort is justifiable for acquiring more accurate prediction results. With regards to the obtained results, \mathcal{S}_3 provides the optimal trade-off; it achieves on par prediction results and requires less computational effort in contrast to the aggressive daily update strategy $\mathcal{S}_1^{\delta=24h}$.

Table 2. Next-activity prediction accuracy of LSTM models [8] against \mathcal{S}_1 - \mathcal{S}_3 .

Data set	Periodic update		Data driven update		
	[8] (Monthly)	$\mathcal{S}_1^{\delta=24h}$	[8] (Drift)	\mathcal{S}_2	\mathcal{S}_3
Helpdesk	0.78	0.928	0.81	0.914	0.931
BPI2012	0.79	0.777	0.79	0.742	0.778

In Table 2, we compare the prediction quality of our LSTM models with the ACC scores reported by [8] for the common data sets. In [8], the authors use a LSTM architecture presented in [14] and operate on batches of events collected on a monthly basis with an expanding window, whereas we operate on a daily basis (cf. Sect. 2). For Helpdesk, our daily periodic update strategy outperforms the monthly approach of [8] by approx. 14% and our \mathcal{S}_2 and \mathcal{S}_3 outperform their drift detection based update strategy by 10-12%. For BPIC2012, our $\mathcal{S}_1^{\delta=24h}$ and \mathcal{S}_3 achieve similar results in comparison to their periodic and drift based update strategy, where our approach falls short of approx. 1.3% in ACC.

6 Conclusion

We have proposed encoding and update strategies for predicting next process activities in an online setting, in particular dealing with unseen data such as process activities that have not been observed in the process event stream so far. The encodings enable the consideration of such unseen activities by holding “empty spots”. The update strategies vary in the frequency the prediction model is updated. This enables us to compare and analyze whether continuous updates result in the best prediction or if even less frequent updates are more beneficiary. Based on a prototypical implementation and three real-world data sets it could

be shown that an “update on demand” strategy yields the best results in terms of balancing prediction quality and performance. This work has some limitations. In our experiments we did not consider potential data imbalances, such as infrequent activities. For instance it is possible that if we split the historic data into a training and validation set, that all infrequent activities might land in the validation set and thus are unknown to the prediction model and potentially ignored because they are underrepresented. Future work includes the testing of greedy update strategies based on additional attributes, using a sliding window in order to limit training data to the recent past, conducting additional experiments with alternative encoding techniques for the data, experimenting with above proposed reserve capacity encoding approach, experimenting with alternative prediction model training approaches such as randomly dropping attribute values during training to further increase a prediction model’s ability to further generalize and be able to handle future unseen data.

Acknowledgments. This work has been supported by Deutsche Forschungsgemeinschaft (DFG), GRK 2201 and by the Austrian Research Promotion Agency (FFG) via the Austrian Competence Center for Digital Production (CDP) under the contract number 881843.

References






1. Evermann, J., Rehse, J.-R., Fettke, P.: A deep learning approach for predicting process behaviour at runtime. In: Dumas, M., Fantinato, M. (eds.) BPM 2016. LNBP, vol. 281, pp. 327–338. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58457-7_24
2. Hancock, J.T., Khoshgoftaar, T.M.: Survey on categorical data for neural networks. *J. Big Data* **7**(1), 1–41 (2020). <https://doi.org/10.1186/s40537-020-00305-w>
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
4. Ly, L.T., Maggi, F.M., Montali, M., Rinderle-Ma, S., van der Aalst, W.M.P.: Compliance monitoring in business processes: functionalities, application, and tool-support. *Inf. Syst.* **54**, 209–234 (2015)
5. Maisenbacher, M., Weidlich, M.: Handling concept drift in predictive process monitoring. In: *Services Computing*, pp. 1–8 (2017)
6. Márquez-Chamorro, A.E., Nepomuceno-Chamorro, I.A., Resinas, M., Ruiz-Cortés, A.: Updating prediction models for predictive process monitoring. In: Franch, X., Poels, G., Gailly, F., Snoeck, M. (eds.) *Advanced Information Systems Engineering. Lecture Notes in Computer Science*, vol. 13295, pp. 304–318. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-07472-1_18
7. Márquez-Chamorro, A.E., Resinas, M., Ruiz-Cortés, A.: Predictive monitoring of business processes: a survey. *IEEE Trans. Serv. Comput.* **11**(6), 962–977 (2018)
8. Pauwels, S., Calders, T.: Incremental predictive process monitoring: the next activity case. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 123–140. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_10
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. Rinderle-Ma, S., Mangler, J.: Process automation and process mining in manufacturing. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) BPM 2021. LNCS, vol. 12875, pp. 3–14. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_1
11. Rinderle-Ma, S., Winter, K.: Predictive compliance monitoring in process-aware information systems: state of the art, functionalities, research directions. Technical report [arXiv:2205.05446](https://arxiv.org/abs/2205.05446) (2022). <https://doi.org/10.48550/ARXIV.2205.05446>
12. Rizzi, W., Di Francescomarino, C., Ghidini, C., Maggi, F.M.: How do i update my model? on the resilience of predictive process monitoring models to change. *Knowl. Inf. Syst.* **64**, 1385–1416 (2022). <https://doi.org/10.1007/s10115-022-01666-9>
13. Sak, H., Senior, A., Beaufays, F.: Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv preprint [arXiv:1402.1128](https://arxiv.org/abs/1402.1128) (2014)
14. 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

Business Process Mining and Discovery



On the Origin of Questions in Process Mining Projects

Francesca Zerbato¹ , Jelmer J. Koorn² , Iris Beerepoot² ,
Barbara Weber¹ , and Hajo A. Reijers² 

¹ University of St. Gallen, St. Gallen, Switzerland
{francesca.zerbato,barbara.weber}@unisg.ch
² Utrecht University, Utrecht, The Netherlands
{j.j.koorn,i.m.beerepoot,h.a.reijers}@uu.nl

Abstract. In line with the growing popularity of process mining, several methodologies have been proposed to guide the conduct of process mining projects. Such methodologies reason that process mining projects start with a concrete question. However, in practice we observe projects with a different starting point, often aimed at exploring the data. Existing methodologies provide limited aid in such situations, and as a result, we wonder: how are questions developed *within* process mining projects? In this paper, we present the results of an interview study that sheds light on question development in process mining. We provide insights from expert interviewees, resulting in six recommendations that enhance existing methodologies. In doing so, we present concrete examples of how process mining analyses can support question formulation and refinement.

Keywords: Process mining · Question development · Interview study · Process mining methodology

1 Introduction

Process mining brings together a variety of methods and techniques for the analysis of process execution data recorded in event logs [1]. Enterprises conduct process mining analyses as part of projects aiming at improving, standardizing, and automating their business processes. Several methodologies have been proposed to guide the planning and execution of process mining projects [8]. These include, among others, the L* lifecycle model [1], the Process Mining Project Methodology (PM²) [6] and the question-driven methodology [14].

Most existing methodologies recommend starting process mining projects by defining “concrete research questions” [6], which, in turn, can be used to guide the extraction and analysis of event logs to find answers [1]. Some methodologies also recommend iteratively refining questions through exploratory analyses [6], suggesting that process mining analyses and their findings can themselves contribute to the development of questions. However, a good research question is not

always available at the start of a project [6]. Sometimes, projects are not triggered by concrete questions but are driven by the availability of data [2]. Moreover, analysts may need to familiarize themselves with the data before being able to formulate concrete questions that they can answer with process mining [16]. As a result, starting a process mining project with a concrete question is not as straightforward as it may sound.

While it is clear that questions play a crucial role in process mining projects [8], research has provided little insight into how questions are developed. Descriptions of case studies focus on answering a specific question, providing a limited picture of question formulation and refinement. In this paper, we aim at closing this gap by looking into the development of process mining questions through the eyes of experts in the field. In detail, we focus on the following research question: *how are questions developed within process mining projects?* We are particularly interested in finding out whether analysts typically start a project with a question, and if not, how they formulate such a question.

To investigate this research question, we followed an empirical approach and engaged in an interview study with experts to learn how they develop questions in their work practices. This empirical approach is fitting as experts can provide insights into how they deal with this issue in practice. Such insights are usually difficult to obtain as it would require significant effort and resources to, for example, organize direct observations. In addition, the literature describing process mining case studies, such as those surveyed by [8], suffers from a reporting bias, meaning it often only reports on the most relevant questions and the related findings. Thus, they provide limited insights into question development.

With this study, we contribute to an improved understanding of question development in two ways. First, we describe how questions are developed within a process mining project as reported by our interviewees. Here, we describe how specific process mining analyses can support question formulation and refinement. Second, we draw on the results of the study and propose a set of recommendations that enhance process mining methodologies, by demonstrating how questions are developed throughout process mining projects. Our findings contribute to existing research by providing concrete ways of supporting the development of questions in process mining projects. We propose a set of recommendations to enhance existing methodologies with question development steps for practitioners who are in charge of overseeing process mining projects.

This paper is structured as follows. In Sect. 2, we introduce background concepts. Section 3 describes our research method that led to the findings presented in Sect. 4. Then, in Sect. 5, we present the recommendations. We close with Sect. 6, where we discuss the limitations of this work and future research plans.

2 Background

In this section, we look at questions in the context of process mining methodologies and discuss the terminology used in this paper.

To support the use of process mining in research and industry, various methodologies have been proposed to guide the execution of process mining

projects. These methodologies include, amongst others, the Process Diagnostics Method [5], the L* lifecycle model [1], and PM² [6]. As summarized in [8], process mining methodologies generally adopt the following structure: (1) definition of questions and goals, (2) data collection, (3) data pre-processing, (4) mining & analysis of results, (5) stakeholder evaluation, and (6) implementation. In this paper, we are mainly concerned with the first step: defining questions and goals, in which objectives and research questions are specified by process analysts in collaboration with organizational stakeholders and domain experts.

The first step has been described differently throughout the literature, since the term *question* has been associated with different levels of abstraction and specificity. For example, in PM² the authors define a “research question” as “a question related to the selected process that can be answered using event data” [6], which can be “abstract” or “concrete” based on the setting. In [13], the authors define *types* of frequently-posed questions (FPQs) in healthcare, differentiating between “generic” and “specific” ones. Generic questions concern general process mining problems, e.g., “What happened?”, while specific ones address specific healthcare needs, e.g., “Do we comply with internal guidelines?”.

Looking at the start of a project, there are a number of aspects that are described: (i) goals/objectives, (ii) questions, and (iii) scope. The methodology by Erdogan and Tarhan [10] addresses all three elements. They are defined as follows: “The *scope* of a project indicates what the process is, when it starts and where it ends, and for which processes and which patients. The *goals* of a project may be related to improving KPIs (e.g., time, cost, risk, and quality). A set of concrete performance-driven *questions* are used to determine the way to assess or achieve these goals” [10, p.5]. Here, examples of questions are “How does the process look like?” or “Where are the bottlenecks in the process?” [10, p.11] Other methodologies focus more on questions, such as the question-driven methodology for analyzing emergency room processes [14], in which process mining projects are prescribed to start from a list of FPQs designed by domain experts. An example of FPQ is “What is the process for treating patients with different diagnoses?” [14, p.4]. Yet others emphasize a subset of elements questions and goals [1,6], or omit all three elements entirely [5]. From the literature, it is clear that questions play a central role at the start of a process mining project, and they largely influence how the project develops. However, none of the considered methodologies describes how a question should be formulated and refined.

In this paper, we adopt a number of definitions of questions. First, we relate to the definition of question given in PM² [6] reported above. In addition, we borrow the concept of “exploratory” questions from literature [4]. With *exploratory* questions, we refer to questions that do not necessarily correspond to a specified goal but focus on understanding the data, discovering patterns, and generating hypotheses. An example is “What happens in this process?”. Exploratory questions are opposite to *confirmatory* questions, which aim at testing a specific hypothesis, such as “Is the invoice process delayed on weekends?”. We consider this distinction relevant as it allows us to understand better how the degree to which a question is exploratory influences how it is formulated and refined.

3 Research Method

To understand how process mining questions are developed, we interviewed experts who have participated in process mining projects. In this section, we describe our research method. First, we describe the data collection. Then, we elaborate on the data analysis based on qualitative coding.

3.1 Data Collection

In this section, we cover the study *design*, where we describe the set-up of the study, the *setting*, where we elaborate on the execution of the study, and the *participants*, where we report on the participants selection and demographics.

Study Design. The interview study presented in this paper is part of a broader study in which we collected data using three methods: (1) a questionnaire, (2) think-aloud, and (3) interviews. The questionnaire was designed to capture the demographics of participants on three primary matters: area of occupation, level of experience, and project experience. It consisted of 18 closed questions:

- Six questions captured basic demographics, including the sector and position in which the participant was employed at that time.
- Seven questions focused on the experience of the participant in terms of: process mining, business intelligence, and data science/engineering.
- Five questions focused on the practical experience with using process mining tools and conducting process mining projects and event log analyses.

Participants were then invited for a virtual session which consisted of two parts: think-aloud and interview. In the first part, they were asked to engage in a realistic process mining task using think-aloud [9]. The task concerned an analysis of the road traffic fine management event log [7] guided by a high-level question asking to investigate circumstances and reasons for not paying a fine.

In this paper, we focus on the information collected in the interviews that were conducted in the second part of the session. Here, participants were interviewed using a semi-structured interview guide. The interview guide consisted of four parts: (1) activities and artifacts, (2) goals, (3) strategies, and (4) challenges. The first part, *activities and artifacts* focused on the steps that participants perform and the information they gather when engaging in process mining analyses. In the second part, *goals*, participants were invited to provide details on their analysis objectives and the amount of exploratory work they typically engage in. This flowed into the third part, *strategies*, where participants were asked about specific plans of actions they follow to achieve their goals. Finally, in the *challenges* part, participants could reflect on the obstacles they run into during the analysis and what kind of support might aid in overcoming them. In each of the four parts, participants were asked to reflect on the interview questions in two contexts: the recently performed process mining task and the broader context of their work practices and experiences. This constant comparison allowed us to better understand how experts work in process mining projects.

Setting. The data was collected between May 1st and July 28th, 2021. Participants were invited for virtual one-on-one sessions with the first author to ensure that the task and the interview were conducted in the same way for all participants. In this session, the participant was granted access to a remote desktop environment with the materials, i.e., data, protocol, and tools. The think-aloud part of the session took roughly 40 min and was recorded through screen capture and voice recording. After this, the participant was asked to report their answers to the guiding question in a post-task questionnaire. Then, the interview was conducted. This lasted roughly 30 min per participant, resulting in a total of 1046 min of audio recording. The audio records were transcribed verbatim for coding purposes by the first author. We note that the example statements from the participants reported in Sect. 4 have been edited to exclude pauses, fillers, and repetitive words. Participants were informed that they could ask questions during the session. Also, they were encouraged but not required to finish the process mining task. In addition, the successful or unsuccessful completion of the process mining task was not relevant for the interview; the task served as a basis to discuss how the participants performed a process mining analysis, but the task itself has little intrinsic value in the context of this study.

Participants. Participants were approached via email through the professional network of the authors, encouraging the recipients to forward the email to anyone else interested. For this paper, we target process mining experts. In detail, we consider the following inclusion criteria. Participants must (1) have analyzed at least two real-life event logs¹ in the two years prior to the study, (2) perceive themselves as knowledgeable with at least one of the process mining tools available for the process mining task and (3) have participated in at least two process mining projects having the goal to analyze process data for a customer. Such criteria allowed us to exclude participants without practical experience in customer projects. The final sample selected for this study consists of 33 participants.

The potential biases we identified that could play a role in the study sample were captured in the demographics questionnaire. Of the 33 selected participants, half ($n = 16$) work as an academic and the other half ($n = 17$) as a practitioner. Overall, participants have an average of 5.6 years of experience in process mining and have experience in data science ($n = 32$) and business intelligence ($n = 30$). Finally, participants hold diverse roles, such as process analyst, process mining consultant, product manager, senior researcher, and Ph.D. candidate.

3.2 Data Analysis

For the analysis of the interview data, we followed a qualitative coding approach [15]. The coding was performed by a team of three of the authors. First, all the members of the coding team individually studied the interview data to get an understanding of the content. Separately, they developed ideas for a possible

¹ In contrast to synthetic logs, real-life event logs are logs obtained from the execution of real-life processes, such as those provided by the IEEE Taskforce in Process Mining: <https://www.tf-pm.org/resources/xes-standard/about-xes/event-logs>.

coding structure. Then, a meeting was held to pitch the different coding structures and merge the ideas. The main coding structure that emerged revolved around steps that process analysts follow in formulating, answering, or refining questions while conducting an analysis in the context of a process mining project. Two authors were tasked with coding the different steps; one focused on the individual steps and the other focused on the relationships between them. The third coder verified the codes of the first two and disagreements were discussed.

As we aimed to study how process mining questions are developed but did not have an initial hypothesis or framework to start from, we used an *inductive approach*. From the transcripts, we used *open coding* [15] to arrive at an initial set of codes. The codes were mainly descriptive of the different steps observed or parts of them. We then used *thematic analysis* to organize our codes [15], as it helped us to identify clusters. Specifically, we clustered the codes based on whether they described a possible start point or endpoint for a process mining analysis. On the highest level, we identified two starting points: “Question” and “No Question”, and three endpoints “Not a process mining question”, “Question answered”, and “New question generated”. This formed the basis for our analysis. In the following step, we started from the codes representing the starting points and endpoints and iteratively searched for dependencies between them and the remaining codes. This led us to discover two high-level themes: analysis and analysis strategies. The *analysis* theme captured different kinds of process mining analyses that the interviewees performed in different settings. Examples are “exploratory analysis” and “pre-defined analysis”. The second theme includes *analysis strategies*, i.e., common approaches not specific to process mining that helped interviewees progress in a process mining project. For example, “evaluate hypothesis” and “explore beyond the question”.

More details on the participants and the data analysis can be found online on <https://doi.org/10.5281/zenodo.6984229>.

4 Results

In this section, we present the question development process emerging from the analysis of the interview data. First, we provide an overview of the whole process, which is depicted in Fig. 1. Then, we go into the details of the question formulation and refinement phases respectively in Sect. 4.1 and Sect. 4.2.

From our analysis, we learned that a question developed in the context of a process mining project can undergo three main phases: question formulation, refinement, and answering. Question formulation concerns posing a question about the process under analysis. Question refinement involves transforming an existing question into another one that can be more specific or easier to answer. Question answering deals with finding an answer to a given question.

Our analysis revealed that such phases originate from two different starting points, *No Question* and *Question*, depicted as orange-filled circles in Fig. 1. In the first case, process analysts do not have a question at hand and need to formulate one. Usually, they start by directly looking at the event log and gathering data-driven insights that can lead to questions, for example, with the help

of “Exploratory analysis”. In the second case, analysts start with a previously formulated question and plan their analysis based on it (“Plan analysis based on question”). Then, they engage in one or more iterations of “Process mining analysis”, to either refine or answer the question. Usually, analysts transition from refining a question to answering it based on the findings of their analyses.

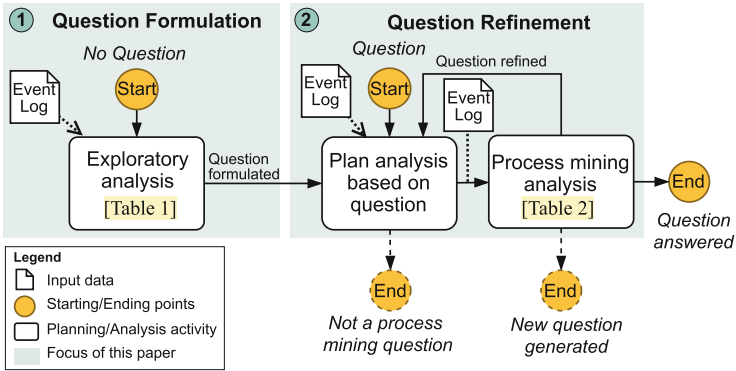


Fig. 1. Overview of the question development process showing different phases of developing a question in the context of a process mining project.

Similar to starting points, we identified three main endpoints, shown as orange-filled circles in Fig. 1: *Question answered*, *Not a process mining question*, and *New question generated*. The first one, *Question answered*, captures the expected end of a process mining analysis. The other two endpoints, depicted with dashed borders in Fig. 1, show possible “exits” from the process of developing one question. The endpoint *Not a process mining question* covers the situation in which a question is not suited to be answered with process mining analyses and, thus, is typically discarded or answered with other analyses. Instead, endpoint *New question generated* shows the generation of new questions, which can occur as a consequence of gaining process knowledge during the analysis.

In the remainder, we focus on the ① question formulation and ② refinement phases. We discuss what analysis activities can support them and how. For each phase, we report on input, factors, and typical steps using example statements from our interviewees. Input are the data at the disposal of the analysts. In this paper, we assume that analysts have an event log available for both phases. Factors are possible influences on question development, which can be the cause for a specific step (e.g., a low level of process thinking maturity) or can affect the choice of one step over another (e.g., the availability of domain knowledge). Steps are different analysis activities.

4.1 Question Formulation: From Event Logs to Questions

Based on our analysis, we define question formulation as the phase of question development that begins without question and concerns deriving and posing a question about the process under analysis.

Analysts start their analysis without question, for example, when business stakeholders are new to process mining and are curious to know what process mining technology can achieve. This is often prompted by the (broad) availability of event data, typical of data-driven projects [2], e.g., *“I have experienced a customer who has 40 GB of data: ‘See what you can find’”* (p11). Indeed, although process mining methodologies prescribe starting with concrete questions (cf. Sect. 2), formulating questions at the start of a project can be difficult [6]. This is partly due to the required participation of stakeholders. One interviewee explained that: *“it is very often hard to identify the correct question. So, sometimes the correct question is just given by process owners, stakeholders, etc., but other times we are just interested in finding out patterns in the event log”* (p36).

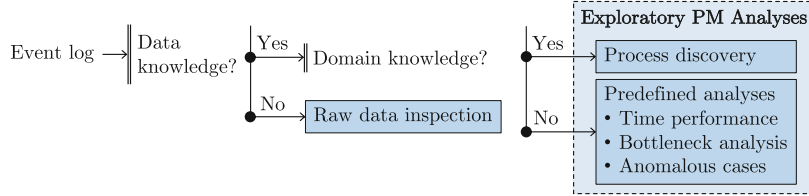
In this setting, process analysts usually start by analyzing an event log to derive data-driven insights or hypotheses that can be discussed with the stakeholders to formulate questions that are aligned with their expectations (cf. Table 1).

Factors. The absence of a question seems to be caused by low levels of process thinking and of process mining maturity, as p11 narrated *“it [the lack of questions] ‘happens more often if the customer is immature in process work and they do not invest any time in the process mining task’* (p11). It may occur in *“projects where people still have limited knowledge of what is possible”* (p39) or *“don’t have any clue of what the process is”* (p40). In such cases, stakeholders do not pose questions but are curious to know *“What is the process that I have?”* (p40). Moreover, at this stage, process analysts may have not yet gathered knowledge about the data and may have little domain knowledge and process knowledge to formulate process mining questions by themselves, especially if they are external to the organization. Thus, they follow different steps based on whether they know the data or have process and domain knowledge, e.g., through external resources such as documentation or from the stakeholders.

Steps. Interviewees reported following two main analysis steps when dealing with the absence of questions, namely (i) raw data inspection and (ii) exploratory analyses. Both steps are helpful in generating data-driven insights and hypotheses that can inspire new questions and assessing what kinds of analyses can be performed on the given event logs. Still, analysts seem to combine such steps based on the factors mentioned above, i.e., their knowledge of the data, the domain, and the process.

Some interviewees reported that, when lacking data and domain knowledge, it is easier to start from the “raw” event log to learn about the structure of the data, typical attribute values, and the underlying data models, if available. This raw data inspection helps analysts gather knowledge about the data and

Table 1. Question formulation. Starting with *No Question* and using “Raw data inspection” and “Exploratory PM Analyses” to derive insights and hypotheses.

Analyses Supporting Question Formulation	
Inputs	Event log
Factors	Process thinking maturity, process mining maturity, data knowledge, domain knowledge, process knowledge
Steps	 <pre> graph LR A[Event log] --> B{Data knowledge?} B -- Yes --> C{Domain knowledge?} B -- No --> D[Raw data inspection] C -- Yes --> E[Exploratory PM Analyses] C -- No --> D subgraph E [Exploratory PM Analyses] F[Process discovery] G[Predefined analyses • Time performance • Bottleneck analysis • Anomalous cases] end </pre>

estimate what analyses can be done on it. Interviewees also reported validating the data and its quality to ensure that it *“is really usable”*, and they can avoid *“working with information that’s completely useless”* (p15). Both data structure and quality can be a starting point for finding analysis questions related to data-driven issues or (new) data extractions. For example, p34 narrated that *“usually I will take time to analyze data quality. [...] the data quality step would help me to see a lot of problems and maybe guide me to the solution.”* (p34).

Moreover, analysts can conduct exploratory process mining (PM) analyses to understand the process and the context in which it is enacted and find insights from the data that lead to hypotheses that can *“inspire the stakeholders about what they could have as a question”* (p12). In process mining, a big part of exploratory analyses is covered by process discovery. In this setting, analysts often exploit the visual artifacts generated by process discovery algorithms as a basis for discussing with stakeholders and developing data-driven hypotheses and questions. One interviewee described this process as *“My way of doing process mining is to explore the dataset as I did now [process mining task] but with more time for reflections. I will take two or three hours in my office alone and try to make sense of the dataset. Next, I will get out some questions [...] and have interactive sessions with the data owner to understand things”* (p34).

Next to process discovery, analysts can resort to predefined analyses, i.e., ready-to-use or “standard” analyses aimed to gather information about process descriptives, which seem particularly helpful in case of limited domain knowledge. Such analyses include user-specified steps that analysts implement based on their experience or are provided by process mining vendors as *“sets of standard hypotheses and analyses behind”* that *“make your life easier so that you don’t start with an empty piece of paper. So, we look at the standard analyses, and that’s always something we bring to the first workshop”* (p33). Interviewees provided examples of predefined analyses, indicating time performance and bottleneck analysis as the most common ones, followed by anomalous, non-compliant cases, and control flow. In particular, process performance and anomalous cases seem to be *“the most common perspective that one can look at while doing process*

mining without having any additional information about the context” (p36) as opposed to, for example, analyzing resource behavior which *“does not give a lot of interesting insights in an exploratory setting. Because typically [...] they [the resources] are anonymized”* (p18). If process knowledge is available, the standard KPIs defined within the organization are also included in predefined analyses.

4.2 Question Refinement: Refining Questions with Process Mining

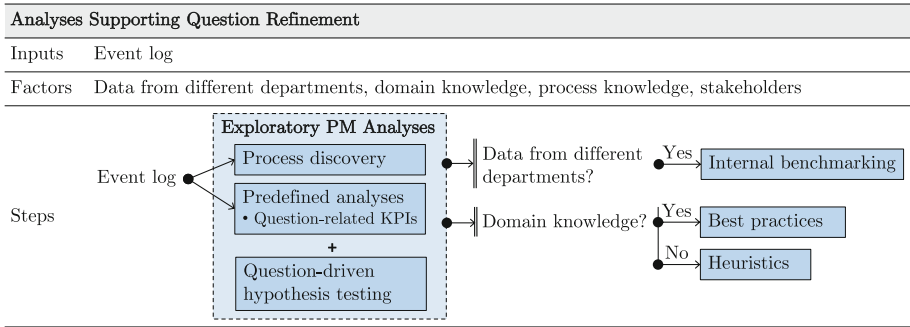
Based on our analysis, we define question refinement as the situation in which analysts start from a question, either posed by stakeholders or resulting from a previous question formulation phase, that needs to be refined. When provided with a question, analysts typically plan their analysis based on the kind of question being asked, as explained by one interviewee: *“the kind of analysis you do very much depends on the main question that is asked”* (p16).

In an ideal setting, analysts are provided with process-related questions, i.e., questions that presume an underlying notion of process control flow and can be suitably answered with process mining techniques. Process-related questions can be exploratory or confirmatory (cf. Sect. 2), which determines if questions need to be refined and, if so, with what analysis steps. Indeed, interviewees reported that *“it’s part of the strategy to adapt the analysis based on the question”* (p9).

However, our findings revealed that analysts can also find themselves dealing with data-related questions, i.e., questions that do not assume an underlying notion of process control flow. This happens, for example, when *“the problem is not directly a process problem but has a more statistical nature”* (p34). In this scenario, analysts can choose to refine the question or perform analyses other than process mining, as exemplified by the *Not a process mining question* endpoint in Fig. 1. Below, we discuss the different steps that analysts can follow to refine process-related questions, which are summarized in Table 2 together with related inputs and factors. We then provide examples of data-related questions.

Refining Process-Related Questions. Interviewees reported several examples of exploratory process-related questions, spanning from general ones such as *“What does this process look like?”* (p16), to questions aimed at investigating a particular process aspect, e.g., performance *“Can we check which machines have bottlenecks?”* (p26) or *“Where do cases spend a lot of time?”* (p19). Usually, exploratory questions require analysts to iteratively refine them, as opposed to confirmatory questions that can be answered more directly. To refine questions, analysts can narrow the analysis focus by *“building hypotheses related to the question that confirm or reject certain possible causes”* (p39) or by identifying patterns in the data that are linked to the question. Such approaches typically lead to “partial answers” that can be confronted with the stakeholders and potentially used to refine the question. One interviewee described the refinement of questions as follows: *“So, start with the first question. So, let’s get some data, get some partial answers, then refine the questions and do it again and maybe two or three or four times and then you converge to something that is robust, that makes sense and that can be used in a much more general way”* (p25).

Table 2. Question refinement. Starting with a *Question* to refine using different kinds of “Exploratory PM Analyses” to narrow the analysis space around the question.



Factors. The availability of stakeholders is a determinant factor for question refinement since business stakeholders often bring domain and process knowledge that analysts can combine with their prior experience to interpret the data and the question. Usually, process knowledge and domain knowledge are exchanged during interactive sessions, where analysts “*understand a lot of the business process with business people*” and learn how to avoid “*silly questions or putting down silly hypotheses*” (p25). Stakeholders also provide crucial feedback on the results and refinement steps. The availability of data from different departments can instead enable the use of benchmarks to narrow the analysis space, which we can also see as a form of refinement.

Steps. Not surprisingly, interviewees reported engaging in exploratory analyses to refine questions, as described in Sect. 4.1. However, given that analysts have both the event log and a question at their disposal, they can combine data-driven analyses, such as process discovery and predefined analyses with question-driven hypotheses to test on the data. Hypotheses are made in different ways, for example by “*finding positive and negative examples related to the questions*” (p39) or by following “*this CRISP thing, right? [...] picking out hypothesis right from the question and then creating something that I can reject or validate this hypothesis*” (p12). Compared to the analyses carried out in the absence of a question, which focus on “*finding interesting things in the data*” (p12), these exploratory analyses are driven by the need to identify parts of the event log that are relevant to the question. For example, predefined analyses can focus on KPIs associated with the question or can help test a given hypothesis.

Our interviewees reported different steps they use to narrow the analysis space around the question. Some interviewees mentioned following a data-driven approach, exploiting the data from different organizational departments for internal benchmarking. Internal benchmarks help narrow the analysis space by allowing analysts to identify critical steps on which to focus as one interviewee explained: “*you most of the time, already know from a benchmark where the critical process steps are, and you can deep dive into those few steps and see if it’s really an issue*” (p23). Other analysts rely on their own domain knowledge and

narrow the analysis space with the help of **best practices** and “good cases”. Best practices are particularly useful to refine questions around improvement opportunities, as they hint towards improvements, e.g., “*We look at the positive cases to see whether there’s a gap in the way our clients work. So, if there’s some best practices missing. So, if we notice that the clients we’re working with didn’t do a step that a lot of phone companies do, we say, ‘look, this would be a good idea for you.’ So, you can look for improvements.*” (p31). While best practices require domain knowledge, analysts can rely on general principles or heuristics to focus on specific parts of the event log. For example, some experts reported focusing on finding cases related to the question within “*the mainstream behavior [of the process] because it’s more supported and makes it easier to reject or validate a hypothesis*” (p41). The mainstream behavior can be separated with the help of heuristics such as the Pareto principle that allows focusing “*where you have more flesh on the bone. I use this 80/20 Pareto principle at the beginning because if you start looking at all the variants, you get lost*” (p26).

Refining Data-related Questions. Data-related questions are about event data but do not explicitly relate to the process control flow. One interviewee explained that data-related questions arise when stakeholders “*don’t have that ‘normal’ process idea, because they don’t get all the information of the status of all the activities in the process*” (p40).

From our interviews, we identified several examples of data-related questions asking, for instance, “*why there are data quality issues*” (p41) or “*what is the percentage of cases that do that*” (p19) or “*how many different activities*” (p37) a log contains. While the first exemplifies a data quality question, the other two require looking into specific data attributes or measurable KPIs. Such questions are often addressed with non-process mining analyses since “*Excel or SQL queries are just much more efficient than trying to do it with process mining*” (p19).

Still, stakeholders ask questions such as “*Could you predict if they are going to pay or not?*” (p7) that analysts can refine and transform into process-related questions. One interviewee described the iterative refinement of data-related questions into process-related ones as follows: “*So, the first thing is that they [the stakeholders] don’t know what ‘process’ means [...] So, all the questions are data-related. So, the first thing you need to do is drive them to the process-related questions. Of course, the data is also interesting, but the process-related analysis is what you can do. And then after that, when you show them some results, like process models, they start to understand what process-related analysis is. And then the questions start to shift. So, it’s never a one, two, or not even three iterations. The first one [iteration] I am sure that is going to be questions to be answered with predictive analysis or data mining, or machine learning*” (p7). In the interviews we found evidence that process discovery results, such as process models are used to refine data-related questions into process-related questions, but we couldn’t derive the detailed steps making up this scenario.

Overall, our findings provide evidence that process mining analyses are used to support question both formulation and refinement. One interviewee well-summarized the relevance of using process mining for question formulation saying that “the nice idea of process mining is that it allows us to detect new research questions” (p40), for example, based on the insights and hypotheses gathered through exploration. Another one remarked how question refinement is intrinsic to iterative analyses: “you start from a hypothesis, get some data, look at the data, go back, refine the hypothesis, get some additional data... So, you repeat the analyses for an entire year” (p25). However, in both settings, close interactions between process analysts and domain experts seem crucial because “without domain knowledge, you won’t achieve much or nothing at all” (p39), especially if organizations are less mature in process mining and thinking.

5 Discussion

In this section, we incorporate our findings into existing process mining methodologies and propose six recommendations (R1–R6) based on our findings with the aim to enhance the current body of knowledge from the perspective of question development. Figure 2 shows, in the dashed boxes connected by arrows, the phases that are typically prescribed by existing process mining methodologies (adapted from [8]). In the blue boxes, we illustrate how the results of Sect. 4 fit into existing methodologies. In addition, we depict as blue arrows with circled numbers recommendations R1–R4, which relate to newly added flows. Recommendations R5–R6 are more generic and, thus, not depicted.

R1 Use process mining to formulate questions. Our interviewees remarked that process mining can provide substantial value in *formulating* questions.

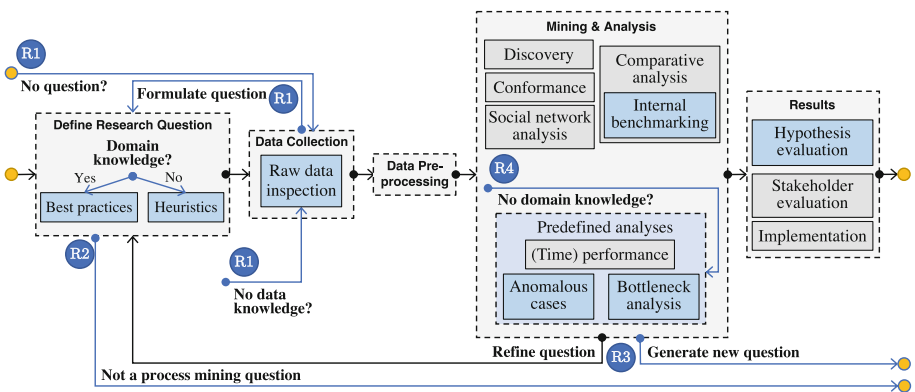


Fig. 2. Incorporation of our findings (blue boxes and arrows) into existing process mining methodologies. Start and end events are depicted as orange-filled circles. (Color figure online)

They reported that questions often emerge from discussions between analysts and stakeholders, who exchange knowledge with the help of process mining tools and artifacts. Process mining tools can be used by process analysts in an exploratory manner to inspect the raw data and “get a feeling” of what interesting directions to investigate are. Based on this, analysts can identify the most fruitful questions that can be formulated. Indeed, the experts in our study indicated that formulating questions without the data at hand is hard. On the other hand, process discovery tools and visual artifacts help spark process thinking among stakeholders. This eases the exchange of the domain and process knowledge required to formulate questions.

- R2 Evaluate if process mining techniques are appropriate to answer the formulated research question.** Our interviewees remarked that we should not always assume that a provided question is a process mining question. This is a good time to reflect on what to do next. As guidance, one might check whether the question at hand fits one of the following frequently asked questions in process mining [13]: (1) What happened? I.e., discovering the process model, (2) Why did it happen? I.e., finding the root causes for a particular situation, (3) What will happen? I.e., predicting process executions in specific circumstances, (4) What is the best that can happen? I.e., finding improvements. When the question does not match these templates, other types of analysis might be more suitable to answer it. For example, the experts indicated that some questions can be solved efficiently with the help of traditional data querying and manipulation languages.
- R3 Document the refinement of questions and the generation of new questions.** During the *Mining & analysis* phase, existing questions are often refined based on new insights. In such a case, it is important to revisit the earlier steps of question formulation and data collection, as also outlined in the PM² methodology [6]. The experts in our study also indicate that entirely new questions can be raised during this project phase. It is good practice to document the development of questions both to inform the stakeholders as well as for proper (academic) reporting. Indeed, keeping track of the questions and the analyses performed to answer them can help streamline the analysis process, identify cause-effect relationships among different questions and ease the answering of questions in future analyses.
- R4 Use predefined analyses to get started.** The use of predefined analyses can help generate questions and spark discussion on what to analyze more in-depth. Most process mining tools have predefined analyses built-in. Usually, these predefined analyses concern time performance, bottleneck analysis, and anomaly detection. Predefined analyses can be a good way to explore potential pain points or improvement opportunities and could be linked to “standard” process mining use cases [3] to help organizations with less process mining maturity get started with projects.
- R5 Value the collaboration between process analysts and stakeholders.** Process mining methodologies describe the value of the collaboration between process analysts and stakeholders in later project phases [6]. Our

results show that such collaboration brings value also in the *early* phase of question formulation. Experts do indicate that setting up a collaboration can be difficult, especially when projects are conducted in organizations with low process mining maturity. However, they advise working interactively and incorporating the stakeholders' knowledge to avoid "getting lost" and "putting down trivial questions". Later on, predefined analyses and, to some extent, discovery can also be done interactively. Experts described this phase as an agile collaboration, developing the work in a similar fashion as Scrum sprints.

R6 Align the question and the analysis. From our findings, we observe the importance of *aligning the question with the appropriate analyses*. Questions can include exploratory or confirmatory aspects, which may influence what analyses are conducted. This, in turn, can explain why process mining projects take a different course based on the application domain. *Exploratory* questions are usually formulated in contexts where prior knowledge of the process is scarce, and the main objective is to discover the process. We found evidence that in some domains such as healthcare [13] exploratory questions already bring much value in promoting process thinking since healthcare information systems are often not process-aware [11]. Instead, *confirmatory* questions intend to verify specific hypotheses, as we emphasized in Fig. 2 by adding a blue box in the *results*. Usually, formulating confirmatory questions requires deep process understanding, which is not always available. Our results reveal that such questions are common in domains such as auditing, where questions put a strong focus on the detection of non-compliance. Although we cannot claim that the course of a process mining project depends only on the questions and their nature, we believe that aligning questions with possible analyses and their outcomes could help organizations assess what they can and cannot achieve with process mining technology.

With these recommendations, we aim at enhancing existing methodologies with tangible examples that show how process mining analyses can support question development. We emphasize that process mining brings value not only for answering (concrete) questions but also for question formulation and refinement.

6 Conclusion

In this paper, we have looked into the development of questions within process mining projects. Drawing on 33 interviews with process mining experts, we have gained insights into how specific process mining analyses can support question formulation and refinement. Then, based on the interview findings, we have proposed six recommendations that enhance existing methodologies with concrete steps supporting question development within process mining projects.

Limitations. Our findings emerged from retrospective interviews and, therefore, are subject to validity threats typical of interview studies, such as reactivity, respondent bias, and researcher bias [12]. We mitigated these risks by:

(1) using a well-developed and pilot-tested interview guide, (2) coding the data with multiple authors, and (3) guaranteeing the anonymity of the participants. Moreover, we note that the set of recommendations presented in this paper may not be complete. Thus, we cannot exclude that additional ones emerge when asking different groups of experts. To mitigate this risk, we considered a sample size of 33 process mining experts with diverse backgrounds and we elaborated on themes that repeatedly emerged across the interviews.

Future Work. In the future, we will refine and extend the list of recommendations considering (i) factors that affect question development in specific settings and application domains and (ii) insights from literature on question development in the broader context of data analysis. We will also conduct a user evaluation to assess the generalizability and practical relevance of our findings.




References

1. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_19
2. Van der Aalst, W.M.: Process mining: discovering and improving Spaghetti and Lasagna processes. In: IEEE Symposium Computational Intelligence Data Mining (CIDM), pp. 1–7. IEEE (2011)
3. 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
4. Behrens, J.T.: Principles and procedures of exploratory data analysis. *Psychol. Methods* **2**(2), 131 (1997)
5. Bozkaya, M., Gabriels, J., Van der Werf, J.M.: Process diagnostics: a method based on process mining. In: International Conference on Information, Process, and Knowledge Management, pp. 22–27. IEEE (2009)
6. 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
7. Eindhoven University of Technology: Road traffic fine management process (2015). Data retrieved from 4TU ResearchData
8. Emamjome, F., Andrews, R., ter Hofstede, A.H.M.: A case study lens on process mining in practice. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) OTM 2019. LNCS, vol. 11877, pp. 127–145. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33246-4_8
9. Ericsson, K.A., Simon, H.A.: Protocol Analysis: Verbal Reports as data. MIT Press, Cambridge (1984)
10. Gurgen Erdogan, T., Tarhan, A.: A goal-driven evaluation method based on process mining for healthcare processes. *Appl. Sci.* **8**(6), 894 (2018)
11. Munoz-Gama, J., Martin, N., Fernandez-Llatas, C., et al.: Process mining for healthcare: characteristics and challenges. *J. Biomed. Inform.* **127**, 103994 (2022)
12. Padgett, D.K.: *Qualitative Methods in Social Work Research*, vol. 36. Sage, Thousand Oaks (2016)

13. Rojas, E., Munoz-Gama, J., Sepúlveda, M., Capurro, D.: Process mining in health-care: a literature review. *J. Biomed. Inform.* **61**, 224–236 (2016)
14. Rojas, E., Sepúlveda, M., Munoz-Gama, J., Capurro, D., Traver, V., Fernandez-Llatas, C.: Question-driven methodology for analyzing emergency room processes using process mining. *Appl. Sci.* **7**(3), 302 (2017)
15. Saldaña, J.: *The Coding Manual for Qualitative Researchers*. Sage, Thousand Oaks (2021)
16. Zerbato, F., Soffer, P., Weber, B.: Initial insights into exploratory process mining practices. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) *BPM 2021. LNBP*, vol. 427, pp. 145–161. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85440-9_9



Extracting Business Process Entities and Relations from Text Using Pre-trained Language Models and In-Context Learning

Patrizio Bellan^{1,2} , Mauro Dragoni¹ , and Chiara Ghidini¹ 

¹ Fondazione Bruno Kessler, Trento, Italy
{pbellan,dragoni,ghidini}@fbk.eu

² Free University of Bozen-Bolzano, Bolzano, Italy

Abstract. The extraction of business processes elements from textual documents is a research area which still lacks the ability to scale to the variety of real-world texts. In this paper we investigate the usage of pre-trained language models and in-context learning to address the problem of *information extraction from process description documents* as a way to exploit the power of deep learning approaches while relying on few annotated data. In particular, we investigate the usage of the native GPT-3 model and few in-context learning customizations that rely on the usage of conceptual definitions and a very limited number of examples for the extraction of typical business process entities and relationships. The experiments we have conducted provide two types of insights. First, the results demonstrate the feasibility of the proposed approach, especially for what concerns the extraction of *activity*, *participant*, and the *performs* relation between a participant and an activity it performs. They also highlight the challenge posed by control flow relations. Second, it provides a first set of lessons learned on how to interact with these kinds of models that can facilitate future investigations on this subject.

1 Introduction

Textual descriptions of business processes, contained for example in Standard Operating Procedure (SOP) documents, are ubiquitous in organizations. While the goal of these descriptions is that of being easy to understand and use, the actual exploitation of the information they contain is often hampered by having to manually analyze unstructured information. Similarly to what happens in e.g., ontology learning [21], techniques that facilitate the extraction of structured information from text can facilitate the usage of advanced techniques like verification, simulation or query answering on top of textual descriptions of processes as recently emphasized in [25].

Process (information) extraction from text can be regarded as the specific problem of finding algorithmic functions that transform textual descriptions of processes into structured representations of different expressivity, up to the entire formal process model diagram. The ambiguous nature of natural language, the multiple possible writing styles, and the great variability of possible domains of

application make this task extremely challenging. Indeed, recent papers on this topic [1, 6, 20] highlight that after more than ten years of research from the seminal work in [14], process extraction from text is a task far from being resolved. By looking at state of the art works, most of the existing approaches rely on template and rule-based approaches, which often lack the flexibility needed to fully cover the great variability of writing styles and process domains [2, 3, 12–14, 17, 23, 26]. Few recent works [16, 22] try to leverage modern Natural Language Processing (NLP) approaches based on deep learning, but they (somehow ironically) restrict the format of the source text to a structured text [16] or to sequential lists of tasks such as recipes or assembly instructions [22], thus avoiding the challenge posed by real world business process descriptions.

One of the problems of leveraging the potential of deep learning NLP is the **lack of the high quantities of carefully annotated data on textual descriptions** needed to make these techniques work, which newly available annotated datasets, such as PET [5], are not yet able to address. The problem is made even worse by the multi-perspective nature of process elements (activities, data objects, process participant (actors), resources, flow objects, and their mutual relations, among others), which require articulated set of annotation labels and the planning of laborious annotation campaigns. Recent advances in NLP, and in particular the availability of large pre-trained language models, and the introduction of novel *in-context learning strategies* that enable the customization of these models in a few shot fashion [8, 24], is opening up new perspectives on the construction of information extraction systems that support search and question answering (among other tasks) by means of multi-turn dialogs in written or spoken form.

In this paper we explore, for the first time in literature, the feasibility of using in-context learning over the pre-trained language models to perform process extraction from textual documents in a conversational fashion. In particular we focus on the GPT-3 (Generative Pre-trained Transformer 3) model [8], one of the state-of-the-art pre-trained large language models (PLM), and explore the feasibility of using it in its native form as well as with two customizations built by providing conceptual definitions of business process elements and a limited number of examples in a few shot learning fashion.

The experiments we have conducted provide two types of insights. First, the results demonstrate the feasibility of the proposed approach, especially for what concerns the extraction of *activities*, *participant*, and the *performs* relation between these two entities, while highlighting the challenge posed by control flow relations for which further training needs to be devised. Second, our experience provides a first set of lessons learned on how to interact with these kinds of models when performing process extraction from text. To the best of our knowledge, our work is the first attempt to use these models on this specific problem and therefore the results and lessons learned are likely to pave the way to future efforts, possibly involving different strategies, target entities and relations and also other pre-trained language models.

The paper is structured as follows. We first provide some required background (Sect. 2); then we describe our approach and its empirical investigation (Sects. 3

and 4); finally, we report insights and lessons learned (Sect. 5), related works (Sect. 6) and concluding remarks (Sect. 7).

2 Background

In this section we provide the main concepts needed for understanding the remainder of the paper.

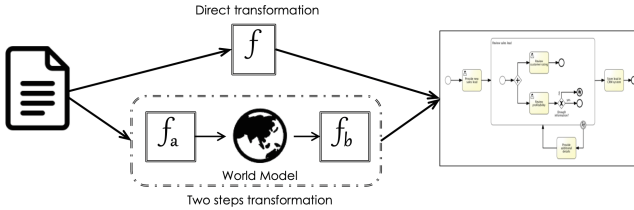


Fig. 1. Two approaches to perform process extraction from text.

2.1 Process Extraction from Text

Process information extraction from text can be regarded as the specific problem of finding algorithmic functions that transform textual descriptions of processes into structured representations of different expressivity, up to the entire formal process model diagram.

If we look at the way this challenge has been tackled in the literature, we can roughly divide the approaches in two big categories. The first one aims to *directly map* a process description into its process model representation via a single function f as graphically depicted in the top part of Fig. 1¹. In literature, function f is typically implemented via a *complex and ad-hoc tailored pipeline*. This approach has the advantage of defining a tailored transformation that can take into account all the available contextual information that can help solving the problem. Nonetheless, this advantage becomes a drawback when we need to devise general solutions or when the algorithmic function f is applied into different contexts. A further approach towards the implementation of a direct mapping f is the exploitation of Artificial Neural Networks. However, the huge quantity of data required to learn a model, and the small quantity of data available in this research domain makes this strategy rarely adopted.

The second approach found in the literature performs a *two-steps transformation approach with intermediate representation* to extract and create a process model. As illustrated in the bottom part of Fig. 1 the algorithmic function f is here considered a *compound* function $f_a \circ f_b$: first, function f_a extracts process elements from text and populates the intermediate representation (also called, world model), then function f_b builds the process model diagram starting from

¹ We have chosen BPMN as an illustrative example but the approach is clearly agnostic to the specific modeling language.

the structured representation of the elements contained in the world model. Often f_a and f_b are further broken down into smaller tasks that allows to better handle the problem complexity.

Similarly to other works (see Sect. 6), our investigation is performed building a computational function f_a , that aims at extracting key elements from text that can populate an intermediate structured representation, which could then be refined or used to build diagrammatic models.

2.2 In-Context Learning

Pre-trained language models (PLMs), such as GPT-3 [8] or BERT [11], are built by using an impressive amount of data and exploiting the advances of deep learning engineering and computational power [8, 24]. PLMs are becoming a hot topic in NLP as they can be adopted, and fine-tuned, to solve complex tasks in different domains, such as open question answering in prototypical common-sense reasoning [7].

While the fine tuning of PLMs for task-specific applications has become standard practice in NLP in the last few years, the advent of GPT-3 has pushed (or better, reduced) the amount of training to the extreme, suggesting a customization by instructions and, if necessary, few shots (a.k.a. few examples) of the task to solve, without the need of updating the parameters of the underlying model (that is, what is now called *in-context learning*). This approach has been shown to be extremely useful to address the training data challenge [27] and has been used to address topics ranging from medical dialogue summarization [9] to hate speech detection [10].

In-context learning relies on the provision of (somehow structured) instructions to the GPT-3 model. Usually, instructions are composed of some contextual knowledge², few examples, and the actual task to be solved. They are coded into a single natural-language template called *prompt*. Therefore, prompt *is* the input that is provided to the GPT-3 model. We illustrate the notion of prompt by showing an example similar to the ones we used in our experiments.

```

1 | Considering the context of Business Process Management and process modelling and
  | ↪ the following definitions:
2 | Activity:
3 | An activity is a unit of work that can be performed by an individual or a group.
  | ↪ It is a specific step in the process.
4 | Consider the following process:
5 | [ SAMPLE TEXT ]
6 | Q: lists the activities of the process
7 | A: [SAMPLE ANSWER]
8 | Consider the following process:
9 | [ ACTUAL TEXT ]
10 | Q: lists the activities of the process
11 | A:
```

Lines 1–3 describe the *contextual knowledge* component and provide the model with some contextual information that is used to narrow the model’s “reasoning ability” to the specific context at hand (BPM, in our case). This knowledge can

² The terminology for these instructions varies from paper to paper.

help the model to disambiguate the different meanings of a word (e.g., activity). In our example they are composed by the identification of the domain (Business Process Management) and a definition. Lines 4–7 describe the (few shots) *examples* component and provide examples of the task to be solved together with the solution. It is composed of a triple containing: (i) the example textual process description, (ii) the task instructions to be performed upon the text, and (iii) the correct answer(s). In our example it only contains one example. Lines 8–10 describe the *task instructions* component and provide the task instructions describing the actual problem to be solved and the *process description* where the task has to be performed upon. Finally, line 11 describes the *eliciting answer mark* component. It tells the model that the prompt is ended and to start producing an answer. At inference time, the prompt is inputted to the LLM in order to support the generation of the answer.

2.3 The PET Dataset

PET [5] is a novel dataset containing the human-annotated version of 45 textual process descriptions of process models. It is the only publicly available annotated dataset specific for process extraction from text tasks, and contains texts annotated with process models elements and relations. While the entire description of the dataset, the annotation guidelines, the annotation schema, and the annotation process are out of the scope of this paper³, here we only report the entire set of entities and relations used to annotate the texts (see Fig. 2) and the specific subset of annotations that we have adopted in our experiments (see the labels in red in Fig. 2). In this paper we make use of: (i) a concatenation of the “activity”⁴ and “activity data” labels to identify activities in the text; (ii) the “actor” process element (hereafter *participant*), that corresponds to the entity responsible for activities’ execution; (iii) the “Actor Performer” relationship between a participant and an activity; and (iv) the “flow” relation of the dataset to denote the (directly) follows relation between activities. Since PET annotates also gateways and guards within the control flow, we have considered here a simplified “flow” relation, implicitly subsumed by the annotations, that is obtained by directly connecting the appropriate activities and removing the extra control flow elements that we do not consider.

³ The interested reader can find all the PET-related resources at <http://huggingface.co/datasets/patriziobellan/PET>.

⁴ The “activity” label is used in PET only to represent the verbal component of what is usually denoted as business process activity.

3 Process Extraction from Text via In-Context Learning

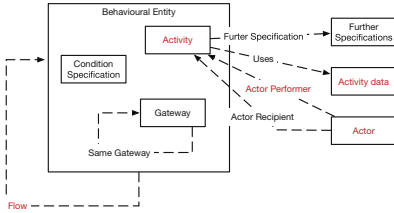


Fig. 2. PET annotation schema. (Color figure online)

of those specific entities and relationships. These questions become the specific tasks that GPT-3 has to solve with the help of specific prompts. Questions do not necessarily correspond to elements in a 1:1 manner, and answers to a question *can* be used as inputs to formulate further questions.

The second building block of the approach is the construction of prompts, and in particular the way we use in-context learning to define the prompts. They are generated starting from prompt templates, that are filled using two types of information: (i) *contextual knowledge* which enable GPT-3 to identify the specific domain at hand (BPM in our case) the elements to be extracted for the different tasks; and (ii) few shots *examples* of the task at hand. To produce the specific prompt the templates also need *task instructions*, and the specific *process description text* upon which to perform the task. Once ready the prompts are fed into the pre-trained model in order to generate the answer.

The third building block of our approach is the pre-trained model that can be used in a conversational manner. We decided to start from GPT-3 [8] since it is one of the state-of-the-art PLMs and it can be adopted without fine-tuning it toward a specific goal. Even though other transformer-like models such as BERT [11] or RoBERTa [18] could be adopted, they usually require more extensive training and fine tuning in order to exhibit acceptable performances. As a consequence, we have decided to directly start our investigation from GPT-3 and from the notion of *in-context learning* and to explore the potential of transformer-like models taking a few shot learning approach. Needless to say, this first investigation into the usage of pre-trained large language models for process extraction from text does not aim at saying the final word on this topic but, on the contrary, it aims at opening up the possibility to better investigate and exploit this kind of approach on other models.

We briefly describe, with the help of Fig. 3, the approach we have followed to perform the extraction of process entities and relations from text via in-context learning.

The starting point is a set of process elements (entities and/or relationships) we aim at extracting, and the first building block of the approach is the formulation of a series of incremental questions posed to the GPT-3 model in a sequential manner which enable the extraction

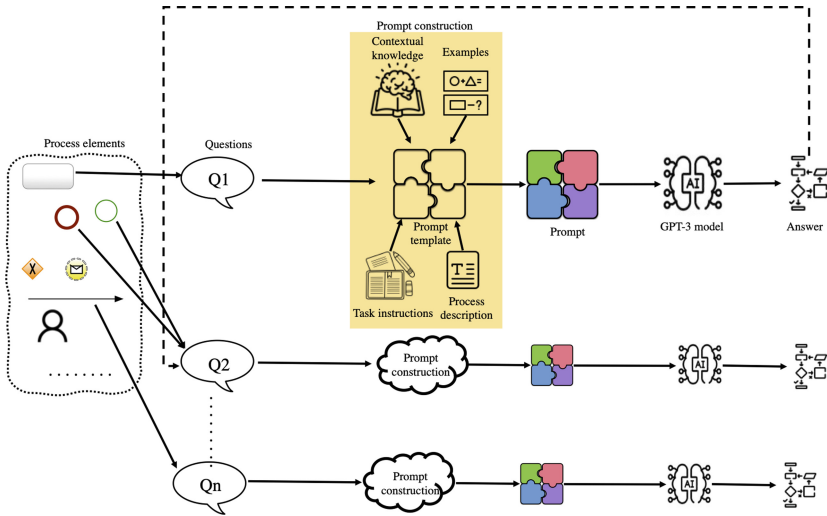


Fig. 3. The figure shows the overall extraction pipeline. The pipeline starts with the composition of the specific prompt for each customization (the specific question “to ask” to the model). Here, only task related process elements and relations definitions fill the contextual knowledge component of prompt-templates. The model receives the composed prompt and generates the specific answer.

3.1 Implementing the Approach

While the overall approach presented here does not depend upon the particular process elements we extract, in this paper we have decided to use it for the extraction of *activities*, *participants*, the *performing* relation between a participant and the activity(ies) it performs, and the sequence relation between activities (hereafter *directly follow* relation). We focus on these four elements as they constitute the basic building blocks on any business process model, they enable the construction of a basic structured representation.

Asking the questions. The questions used to extract *activities*, *participant*, the *performing* relation, and the *directly follow* relation from text are reported in Fig. 4. As we can see, these questions are posed in an incremental manner: first, we ask questions about the process activities (Q_1), then we enrich the activities with the participants performing them (Q_2), and finally we ask about the precedence relation among activities (Q_3). Note that question Q_2 is used to retrieve both the participant and the performing relationship between activities.

While the incremental order of the questions was a way to break down the complex task of extracting complex process diagrams from text into its smaller components, and other orderings or styles of questions can be obviously explored from here, its usage is interesting because it can be used to mimic the way we often build conceptual models using follow up questions. Moreover, incremental questions can be easily automatized to support, e.g., a conversational information seeking system.

- Q₁:** Lists the activities of the process;

Q₂: Who is the participant performing activity X in the process model?
for each activity returned by **Q₁**;

Q₃: Considering the list of process activity described in the text, does activity X immediately follow activity Y in the process model?.,
for each pairs of different activities X and Y returned by **Q₁**.

Fig. 4. The questions adopted as *task instructions* in prompts.

An important aspect in setting up the questions is their correspondence with the elements to be extracted and the specific wording to be used, that is, learning to ask the *right* question. This first work does not aim at investigating this aspect in depth. Nonetheless, Sect. 5 reports some insights we gained in our work. We are aware that there is a growing literature corpus on prompt-based fine-tuning, as described, e.g., in [15], but an investigation into the most efficient prompt is out of scope for this paper.

Building the prompts with in-context learning customizations. Our in-context learning approach exploits two sources of “information”: *contextual knowledge* and few *examples* related to the task at hand.

In this work the *contextual knowledge* component contains the identification of the specific domain and *intensional definitions* of the elements to be extracted that are specific for the different tasks. For this specific paper it consists in the text in Fig. 5: a preamble identifying the BPM context, common for all prompts, plus the definitions of Activity, Participant, Process Model, Flow and Sequence Flow, that are used for answering the different questions (in the Figure each definition is labelled with the question it was used for). The decision if investigating the usage of intensional definitions is based on the fact that this information is available in the BPM community and also it may reduce the dependency from examples. The definitions of Activity, Participant, and Process Model are slight rewordings of the answers obtained by asking the question “what is an activity (resp. participant, process model) in the context of Business Process Management?” to GPT-3 itself. The definitions of Flow and Sequence Flow instead were extracted from the annotation instructions of the PET dataset that we use for the empirical assessment. These choices were made to minimize the external knowledge inserted in our assessment and - at the same time - provide a first empirical assessment of using intentional definitions in the customization of pre-trained models.⁵

⁵ Several definitions exist of many business process elements (see e.g., www.businessprocessglossary.com), but they often present different wordings and even conflicting characteristics [4]. A thorough investigation of the impact of different definitions of business process elements is out of the goal of this paper and is left for future works.

Considering the context of Business Process Management and process modelling and the following definitions:

Activity: An activity is a unit of work that can be performed by an individual or a group. It is a specific step in the process.

Participant: A participant is any individual or entity that participates in a business process. This could include individuals who initiate the process, those who respond to it, or those who are affected by it.

Process Model: A process model is a model of a process in terms of process activities and their sequence flow relations.

Flow: A flow object captures the execution flow among the process activities. It is a directional connector between activities in a Process. It defines the activities' execution order.

Sequence Flow: A Sequence Flow object defines a fixed sequential relation between two activities. Each Flow has only one source and only one target. The direction of the flow (from source to target) determines the execution order between two Activities. A sequence relation is an ordered temporal relation between a source activity and the activity that immediately follow it in the process model.

Fig. 5. Contextual knowledge provided to the pre-trained model.

Examples of the task at hand are provided in a few shot fashion, that is by using an extremely limited number of examples. The examples are composed by textual descriptions of processes together with the pairs of all questions and correct answers. For example, a sample for question **Q₁** would be a process textual description together with the list of activities we can find in that text.

4 Empirical Assessment

We provide below the process we performed to evaluate the proposed implemented approach. We start by introducing the research questions, the tasks, the adopted settings, and then we illustrate the obtained results. In the next section we discuss the gathered insights. We start by assessing the feasibility of using GPT-3 with in-context learning to extract process entities and relations from texts by means of the following research questions:

RQ1 Is it possible to use the GPT-3 pre-trained language model in its native form to extract specific process entities and relations from a textual process description?

RQ2 Are in-context learning customizations (prompts) that rely on contextual BPM knowledge and few examples more effective than the native model in solving the tasks? Are there differences between providing contextual BPM knowledge and the examples?

We perform all the experiments adopting the *text-davinci-001* engine and set the model's parameters to 0.0. We want to remark here that the comparison among different model's configurations it is reserved for future investigation, but it is out of the scope of this paper.

4.1 The Tasks

We evaluated the proposed approach by extracting activities, participants, the performing relation, and the directly follow relation as illustrated in Sect. 3.1. We do it by asking GPT-3 to perform three tasks: (**T₁**) extract activities, using question **Q₁** as task instruction in the prompt-template; (**T₂**) extract participants and the performs relation, using a customized question **Q₂** for each input activity as task instruction in the prompt-template; (**T₃**) extract the directly follow relation using a customized question **Q₃** for each pair of input activities as task instruction in the prompt-template. Since the second and third tasks need activities as an input, we have decided to split them into two different versions: one in which we use the activities extracted in task **T₁** as an input, and another in which we use the activities that are present in the golden standard texts we did use for the validation (see Sect. 4.3) as input. The first version gives an evaluation of **T₂** and **T₃** as incremental follow up tasks of **T₁**, while the second enables us to evaluate the capability to extract participants, and the two relations as stand alone tasks. In the results we use the suffix *ex*(tracted) for the results of the first version and the suffix *gs* (gold standard) for the results of the second version of tasks **T₂** and **T₃**.

4.2 Experimental Setting

We evaluated the proposed approach with four experimental settings, which we describe using the terminology described in Sects. 2.2 and 3.

RAW: the GPT-3 model has been used as it is provided by the maintainers without any customization. This setting works as baseline to observe the capability of pre-trained language models of working within complex scenarios. It was created by providing the prompt-template only with the task instructions and the process description text.

DEFS: the GPT-3 model was provided with definitions to observe the capability of the system to exploit some conceptual knowledge over the domain. This setting is created on top of RAW by adding to the prompt of the RAW setting contextual knowledge in the form of the contextual information and the definitions shown in Fig. 5.

2SHOTS: the GPT-3 model model has been enhanced by providing few shots examples to observe the capability of the system to learn from a very limited number of samples. This setting is created on top of RAW by adding to the prompt of the RAW setting the examples provided by using the gold standard annotations of documents doc-2.2 and doc-10.9 of the PET dataset. In order to avoid the injection of non-essential information, the examples for each task did pertain only the output required for the task. E.g., in order to perform task **T₁** the prompt did contain only the activities contained in documents doc-2.2 and doc-10.9.

DEFS+2SHOTS: the GPT-3 model has been enhanced by using both strategies described above. This setting is created on top of RAW by building a prompt that contains both the DEFS and 2SHOTS customizations illustrated above.

The reader may find all the prompts and the material related to this paper at http://pdi.fbk.eu/pet/edoc2022/edoc2022_material.zip

4.3 Validation Dataset

Table 1. The characteristics of the documents.

Text	word#	activity#	participant#	follow#	perform#
doc-1.2	100	10	2	10	10
doc-1.3	162	11	5	11	12
doc-3.3	71	7	2	6	4
doc-5.2	83	7	3	6	4
doc-10.1	29	4	2	4	4
doc-10.6	30	4	2	4	4
doc-10.13	39	3	2	2	3

perform and follow relations are reported in Table 1.

We are aware that the analysis of seven documents has limitations from a statistical significance point of view. However, the rationale behind this empirical evaluation is two-fold. First, since this is a first observational study of a promising groundbreaking strategy, we decided to select documents having specific characteristics in order to perform an ad-hoc analysis about how the pre-trained language model worked on them. Second, the application of the proposed approach passed through several refinements round before to be tested since we had to understand how the pre-trained language model actually works. Hence, to better understand the impact of the information provided by us to enrich the pre-trained language model, the most suitable way was to observe such behaviors on a small but characteristic subset of document.

4.4 Answering the Research Questions

Table 2 provides the results of the empirical assessment performed on the validation dataset. The first column contains the document identifier, the second column contains the elements or the relations for which the measures are provided, and the remaining of the columns describe the results for the different settings⁶.

The results obtained highlight few interesting patterns. Concerning **RQ₁** we can see that strategy RAW provides unsatisfactory results, thus highlighting that the GPT-3 raw model is not able to address the task of extracting different process elements from text in a satisfactory manner. In general, therefore, we can answer “NO” to **RQ₁**. Few observations can nonetheless be made. Overall,

⁶ In few cases the model was able to provide semantically correct answers which did not match the exact PET labels. A paradigmatic case is the answer “check and repair the computer” as a single activity, instead of the two separate ones which are reported PET, as required by its specific annotation guidelines. We have carefully considered these few cases and decided to evaluate the semantically correct answers as correct answers.

Table 2. The results for the four settings.

Text ID	Element	RAW			DEFS			2SHOTS			DEFS+2SHOTS		
		prec	rec	F1	prec	rec	F1	prec	rec	F1	prec	rec	F1
doc-1.2	Activity	1.00	0.50	0.67	1.00	0.50	0.67	0.75	0.90	0.82	0.75	0.90	0.82
	Participant (gs)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Participant (ex)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Follows (gs)	0.00	0.00	0.00	0.33	0.10	0.15	0.67	0.20	0.31	0.40	0.20	0.27
	Follows (ex)	0.00	0.00	0.00	0.00	0.00	0.00	0.40	0.29	0.33	0.31	0.57	0.40
	Performs (gs)	0.20	1.00	0.33	0.60	1.00	0.75	1.00	1.00	1.00	1.00	1.00	1.00
	Performs (ex)	0.33	1.00	0.50	1.00	1.00	1.00	0.70	1.00	0.82	0.70	1.00	0.82
doc-1.3	Activity	1.00	0.69	0.82	1.00	0.73	0.85	1.00	0.87	0.93	1.00	0.87	0.93
	Participant (gs)	1.00	1.00	1.00	1.00	0.60	0.75	1.00	1.00	1.00	1.00	1.00	1.00
	Participant (ex)	1.00	0.60	0.75	1.00	0.60	0.75	1.00	0.80	0.89	1.00	0.80	0.89
	Follows (gs)	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.73	0.29	0.18	0.73	0.29
	Follows (ex)	0.00	0.00	0.00	0.13	0.07	0.09	0.22	0.75	0.34	0.19	0.38	0.25
	Performs (gs)	1.00	1.00	1.00	0.91	0.91	0.91	0.97	1.00	0.98	0.97	1.00	0.98
	Performs (ex)	0.78	1.00	0.88	0.82	1.00	0.90	0.92	1.00	0.96	0.92	1.00	0.96
doc-3.3	Activity	1.00	0.57	0.73	0.86	1.00	0.92	1.00	0.86	0.92	0.88	1.00	0.93
	Participant (gs)	0.67	1.00	0.80	0.67	1.00	0.80	0.67	1.00	0.80	0.67	1.00	0.80
	Participant (ex)	0.67	1.00	0.80	0.50	1.00	0.67	0.67	1.00	0.80	0.67	1.00	0.80
	Follows (gs)	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.50	0.24	0.16	0.50	0.24
	Follows (ex)	0.00	0.00	0.00	0.00	0.00	0.00	0.24	0.80	0.36	0.15	0.67	0.25
	Performs (gs)	0.57	1.00	0.73	0.57	1.00	0.73	0.57	1.00	0.73	0.57	1.00	0.73
	Performs (ex)	0.75	1.00	0.86	0.43	1.00	0.60	0.67	1.00	0.80	0.50	1.00	0.57
doc-5.2	Activity	0.00	0.00	0.00	1.00	0.57	0.73	1.00	0.86	0.92	1.00	0.86	0.92
	Participant (gs)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Participant (ex)	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Follows (gs)	0.00	0.00	0.00	0.00	0.00	0.00	0.23	0.83	0.36	0.24	0.83	0.37
	Follows (ex)	0.00	0.00	0.00	0.00	0.00	0.00	0.24	0.80	0.36	0.25	0.80	0.38
	Performs (gs)	0.57	1.00	0.73	0.57	1.00	0.73	0.48	1.00	0.64	0.43	1.00	0.60
	Performs (ex)	0.00	0.00	0.00	0.75	1.00	0.86	0.39	1.00	0.56	0.33	1.00	0.50
doc-10.1	Activity	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
	Participant (gs)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Participant (ex)	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
	Follows (gs)	0.00	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.67	0.38	1.00	0.55
	Follows (ex)	0.00	0.00	0.00	0.00	0.00	0.00	0.33	1.00	0.50	0.43	1.00	0.60
	Performs (gs)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Performs (ex)	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
doc-10.6	Activity	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
	Participant (gs)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Participant (ex)	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
	Follows (gs)	0.00	0.00	0.00	0.00	0.00	0.00	0.60	1.00	0.75	0.43	1.00	0.60
	Follows (ex)	0.00	0.00	0.00	0.00	0.00	0.00	0.60	1.00	0.75	0.43	1.00	0.60
	Performs (gs)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Performs (ex)	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
doc-10.13	Activity	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	0.60	1.00	0.75
	Participant (gs)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.67	1.00	0.50	0.67
	Participant (ex)	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	0.5	0.67
	Follows (gs)	0.67	1.00	0.80	0.00	0.00	0.00	0.29	1.00	0.44	0.25	1.00	0.40
	Follows (ex)	0.00	0.00	0.00	0.00	0.00	0.00	0.29	1.00	0.44	0.22	1.00	0.36
	Performs (gs)	1.00	1.00	1.00	1.00	1.00	1.00	0.67	1.00	0.80	0.67	1.00	0.80
	Performs (ex)	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	0.40	1.00	0.57
Average	Activity	0.43	0.25	0.32	0.55	0.40	0.45	0.96	0.93	0.94	0.89	0.95	0.91
	Participant (gs)	0.95	1.00	0.97	0.95	0.94	0.94	0.95	0.93	0.92	0.95	0.93	0.92
	Participant (ex)	0.38	0.37	0.36	0.50	0.51	0.49	0.95	0.97	0.96	0.95	0.90	0.91
	Follows (gs)	0.10	0.14	0.11	0.05	0.01	0.02	0.38	0.75	0.44	0.29	0.75	0.39
	Follows (ex)	0.00	0.00	0.00	0.02	0.01	0.01	0.33	0.81	0.44	0.28	0.77	0.41
	Performs (gs)	0.76	1.00	0.83	0.81	0.99	0.87	0.81	1.00	0.88	0.81	1.00	0.87
	Performs (ex)	0.27	0.43	0.32	0.43	0.57	0.48	0.81	1.00	0.88	0.69	1.00	0.77

RAW provides good results for the extraction of participants (gs) and performs (gs). This may show that information about participants performing something is somehow part of the general knowledge that a pre-trained linguistic model has. This is even more evident if compared with the performance of RAW in extracting the follow (gs) relation, which emphasizes a difficulty of GPT-3 in understanding even basic temporal relations. In texts doc-1.2, doc-1.3 and doc-3.3, RAW provides also acceptable results for activity, thus highlighting that the RAW model works well with complex texts. This may be a consequence of the fact that longer texts provide a more complete context of the process description. Hence, such a context better helps the RAW model in understanding which are the most relevant elements to detect.

Concerning **RQ₂** we can see that in-context learning approaches that rely on contextual BPM knowledge and few examples more effective in providing answers than the native GPT-3 model and can lead to good results. In particular 2SHOTS appears to be the best overall strategy. Adding contextual BPM knowledge, and in particular definitions, is useful in specific cases - thus hinting to a possible positive effect - but does not appear to be a valid general strategy, neither when it is provided alone (as in DEFS) not when it complements the examples (as in DEFS+2SHOTS). Whether this is the effect of the definitions we tested or definitions in general is left for future work to assess. The result we can report here is that adding the *right* contextual BPM knowledge may be a non-trivial problem that needs to be carefully investigated. A exception to the overall satisfactory performance of 2SHOTS is given by the *Follow* relationship both in its (ex) and (gs) settings. In fact, while providing the two examples is useful to increase the performances w.r.t. the RAW baseline, the gain is often limited. This result may indicate that the knowledge of temporal relationships of GPT-3 is insufficient for the BPM domain and better ad-hoc training is needed. To sup up, we can positively answer **RQ₂** saying that in-context learning in a few-show fashion can be used to extract process information from text, with the need to better investigate the extraction of the follows relationship.

Finally, we observed how the performance are not related to the length of the documents. Indeed, from Table 1, we may observe that we worked with long documents (i.e., Texts doc-1.2, and doc-1.3), medium-size documents (i.e., Texts doc-3.3 and doc-5.2), and short documents (i.e., Texts doc-10.1, doc-10.6 and doc-10.13). By observing the behavior of each setting reported in Table 2, no relevant differences may be observed from the metrics. This aspect is particularly interested since it means that the proposed strategy may be applied in different scenarios without considering the documents length as a criticality to address.

5 Further Insights

This work represents a first attempt towards the use of in-context learning techniques for the extraction of process model elements from text. From this experience we gathered several insights, that encompass the results reported in Table 2. We report them below as they will trigger future investigation. We may group

such insights within three main categories: (i) the type of interaction between the user and the model; (ii) the parameters to adopt for understanding the behavior of the model; and (iii) the understanding of which information better benefits the effectiveness of the in-context learning.

Interaction between the user and the model. The first important aspect to consider is how the interaction between the user and the machinery occurs. It has to be as much simple, and complete, as possible. Indeed, the model (independently by the prompt adopted) is not able to provide proper answers to generic and complex questions. In our preliminary tests, we asked the model: *List all the activities and their precedence relations*. Here, two information are requested at the same time and having the second one that is semantically dependent by the first one. We observed that this type of requests led to an empty or, in general, not significant outcome. The same issue occurs when questions are not complete. For example, if we omit the word *performer* when we ask for who is the participant performing a specific activity, the model tends to generate a wrong answer. This is an important aspect to consider since the model has a limited inference capability with respect to the human-like cognitive process.

Parameters to understand model behavior. During preliminary tests, i.e., before to run the empirical assessment, we tested different prompt templates. This type of test was necessary to observe if the format of the template may affect the quality and variety of the results. This preliminary investigation confirmed this hypothesis since, depending on the prompt format, results are different.

Understanding of which information better benefits the effectiveness of the in-context learning. The third aspect relates to the selection of information to provide for the in-context learning phase in order to improve the model capability in answering correctly. A straightforward aspect we validated is that the choice of examples used for the few-shot customization has a relevant impact on the results. During the preliminary tests we chosen different customizations to observe the inference capability of the system in order to tune the queries. As example, the use of the term *Participant* instead of *Actor* led to better results. The hypothesis is that the term *Actor* may be considered a more general term used in several contexts. While, the word *Participant* brings the semantic meaning of someone having an active role in a task. This aspect will be further investigated in the future. Another case was instead related to ambiguous knowledge that may be previously loaded into the model. Indeed, we observed that, concerning the concept of *Activity* different definitions of such a concept were already included in the model. Hence, in order to preserve the capability of acting in an effective way within the business process domain, we adopted the prefix text *Consider the context of Business Process Management and Process modeling* before queries related to the extraction of activities from the text. This triggers a higher effectiveness of the model.

Potential impact on annotation campaigns. The performance obtained by the proposed approach highlight how in-context learning techniques may be used side-by-side with human experts within annotation campaigns [28]. In general, the annotation of business process model texts is a complicated task to be per-

formed from scratch. A potential impact of our approach is to support the annotation task by providing candidate annotations. This is a preliminary analysis of this aspect: future investigations will consider how to integrate our approach in annotation campaigns.

The analysis we performed on the behavior of the pre-trained model and on the in-context learning customizations paved the way to further explore the approach proposed in this paper. The outcomes of the performed empirical assessment demonstrated the suitability of the proposed approach that has the potential of becoming a groundbreaking strategies for the expert-supervised incremental building of business process models from texts by exploiting, e.g., conversational information seeking paradigm.

6 Related Work

We can group existing work on extracting process elements or an entire process model from text in three different streams.

In the first stream, process extraction from text is addressed through a direct mapping function. [3] proposes to learn how to build up a process model diagram through user's feedbacks. Whenever a sentence is provided in input to the tool, the engine checks if a corresponding mapping rule exist. If there are no rules to describe a process element, the system exploits the user's feedbacks to create one. So, the the user has to manually map a text fragment into its process model element. Then, the system creates the corresponding mapping rule. The works of [2] and [19] target the extraction of declarative constraints expressed in DECLARE and DCR graphs respectively in a rule-based fashion. The extraction is performed from single sentences and focuses on the identification of roles, activities, and a specific set of DECLARE (resp. DCR graph) constraints. Finally, the recent work of [16] proposes a deep learning solution to the problem of automatically discovering a business process and its corresponding process model diagram (represented in BPMN) without extra human labeled knowledge. The generality of this approach is hampered by the restriction of the input to heavily structured Process Definition Documents.

The second stream contains works that address process extraction from text as a two-steps transformation approach with intermediate representation [12,14,17,26]. All these works produce BPMN process model diagrams by exploiting rules or templates. The seminal work of [14] exploits templates and a slight modification of a CREWS intermediate World Model to extract BPMN diagrams from text without making assumptions regarding the input text. It also introduces a publicly available dataset of 47 pairs composed of a process model description, and a process model diagram for performing the evaluation, which is still considered as one of the reference datasets up-to-date. Despite still being one of the reference works on this topic, the work in [14] failed to address the variability of real-world texts and was not developed further. [26] presents a method to create a process model diagram and its associated ontology from semi-structured use case descriptions (descriptions on how an entity should cooperate

with other entities), while [12] focuses on semi-structured textual descriptions of process executions in the archaeological domain. Finally, [17] exploits a syntactic analysis of the input text to extract Subject-Verb-Object constructs and keywords-based extraction for the identification of Gateways.

The third stream of work instead tackles only the f_a function of the two-steps transformation approach, thus extracting information as lists of tags from a text. In particular, the works in [13] and [23] follow a rule-based approach to extract mainly control and data flows in a procedural and declarative fashion, respectively. Also, the recent work of [22] proposes a hierarchical neural network approach, called Multi-Grained Text Classifier (MGTC), to tackle the problem of classifying sentences in procedural texts without engineering any features. Unfortunately, the approach is only applied to sequential lists of tasks such as recipes or assembly instructions.

In the context of process extraction from text, our work belongs to the third stream above. Differently from state-of-the-art research, and to the best of our knowledge, this is the first research endeavor aimed at extracting process information from text using a deep learning method based on pre-trained language representation models which aims at dealing with an entire textual description without making assumptions regarding the input text. It also does it in an incremental and flexible conversational fashion so as to extract the required information via question and answering dialogues.

7 Conclusion

In this paper we have investigated the feasibility of leveraging GPT-3 language model and in-context learning approach to perform process information extraction from textual documents in an incremental question and answering manner. The results highlighted the potential of the in-context learning approach which can substantially address the “training data challenge” of deep-learning-based NLP techniques in the BPM field. The results show the feasibility of our proposed methodology. This opens the possibility to use this technique to address the construction of business process model by starting from natural language text in scenarios where it is necessary to manage the low-resources issues and by exploiting the human-in-the-loop paradigm given the role of the domain expert in processing the information provided by the model. We also reported a suite of lessons learned from this experience that will drive the development of further research.



References

1. van der Aa, H., Carmona, J., Leopold, H., Mendling, J., Padró, L.: Challenges and opportunities of applying natural language processing in business process management. In: COLING 2018 Proceedings of 27th International Conference on Computational Linguistics, pp. 2791–2801. ACL (2018)
2. van der Aa, H., Di Ciccio, C., Leopold, H., Reijers, H.A.: Extracting declarative process models from natural language. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 365–382. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_23
3. Ackermann, L., Volz, B.: model[NL]generation: natural language model extraction. In: Proceedings of the 2013 ACM workshop DSM@SPLASH 2013, pp. 45–50. ACM (2013)
4. Adamo, G., Di Francescomarino, C., Ghidini, C.: Digging into business process meta-models: a first ontological analysis. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) CAiSE 2020. LNCS, vol. 12127, pp. 384–400. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49435-3_24
5. Bellan, P., van der Aa, H., Dragoni, M., Ghidini, C., Ponzetto, S.P.: PET: an annotated dataset for process extraction from natural language text tasks. In: Proceedings of the BPM 2022 First Workshop on Natural Language Processing for Business Process Management (NLP4BPM) co-located with the 20th conference Business Process Management, CEUR Workshop Proceedings. CEUR-WS.org (2022)
6. Bellan, P., Dragoni, M., Ghidini, C.: Process extraction from text: state of the art and challenges for the future. CoRR abs/2110.03754 (2021)
7. Boratko, M., Li, X., O’Gorman, T., Das, R., Le, D., McCallum, A.: ProtoQA: a question answering dataset for prototypical common-sense reasoning. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, pp. 1122–1136. ACL (2020)
8. Brown, T.B., et al.: Language models are few-shot learners. In: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020 (2020)
9. Chintagunta, B., Katariya, N., Amatriain, X., Kannan, A.: Medically aware GPT-3 as a data generator for medical dialogue summarization. In: Proceedings of the 6th Machine Learning for Healthcare Conference, Proceedings of Machine Learning Research, vol. 149, pp. 354–372. PMLR (2021)
10. Chiu, K., Alexander, R.: Detecting hate speech with GPT-3. CoRR abs/2103.12407 (2021)
11. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT 2019, Vol. 1, pp. 4171–4186. ACL (2019)
12. Epure, E.V., Martín-Rodilla, P., Hug, C., Deneckère, R., Salinesi, C.: Automatic process model discovery from textual methodologies. In: 9th IEEE International Conference on Research Challenges in Information Science, RCIS 2015, pp. 19–30. IEEE (2015)
13. Ferreira, R.C.B., Thom, L.H., Fantinato, M.: A Semi-automatic approach to identify business process elements in natural language texts. In: ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, Vol. 3, pp. 250–261. SciTePress (2017)
14. Friedrich, F., Mendling, J., Puhmann, F.: Process model generation from natural language text. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS,

- vol. 6741, pp. 482–496. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21640-4_36
15. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. In: Proceedings of ACL/IJCNLP 2021, pp. 3816–3830. ACL (2021)
 16. Han, X., et al.: A-BPS: automatic business process discovery service using ordered neurons LSTM. In: 2020 IEEE International Conference on Web Services, ICWS 2020, pp. 428–432. IEEE (2020)
 17. Honkisz, K., Kluza, K., Wiśniewski, P.: A concept for generating business process models from natural language description. In: Liu, W., Giunchiglia, F., Yang, B. (eds.) KSEM 2018. LNCS (LNAI), vol. 11061, pp. 91–103. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99365-2_8
 18. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. ArXiv abs/1907.11692 (2019)
 19. López, H.A., Debois, S., Hildebrandt, T.T., Marquard, M.: The process highlighter: from texts to declarative processes and back. In: Proceedings of Dissertation Award, Demo, and Industrial Track, BPM 2018. CEUR Workshop Proceedings, vol. 2196, pp. 66–70. CEUR-WS.org (2018)
 20. 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
 21. Petrucci, G., Rospocher, M., Ghidini, C.: Expressive ontology learning as neural machine translation. *J. Web Semant.* **52–53**, 66–82 (2018)
 22. Qian, C., et al.: An approach for process model extraction by multi-grained text classification. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) CAiSE 2020. LNCS, vol. 12127, pp. 268–282. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49435-3_17
 23. Quishpi, L., Carmona, J., Padró, L.: Extracting annotations from textual descriptions of processes. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 184–201. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_11
 24. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**, 140:1–140:67 (2020)
 25. Sánchez-Ferreres, J., Burattin, A., Carmona, J., Montali, M., Padró, L., Quishpi, L.: Unleashing textual descriptions of business processes. *Softw. Syst. Model.* **20**(6), 2131–2153 (2021). <https://doi.org/10.1007/s10270-021-00886-x>
 26. Sawant, K.P., Roy, S., Sripathi, S., Plesse, F., Sajeew, A.S.M.: Deriving requirements model from textual use cases. In: 36th International Conference on Software Engineering, ICSE 2014, Proceedings, pp. 235–244. ACM (2014)
 27. Scao, T.L., Rush, A.M.: How many data points is a prompt worth? In: Proceedings of NAACL-HLT 2021, pp. 2627–2636. ACL (2021)
 28. Wang, S., Liu, Y., Xu, Y., Zhu, C., Zeng, M.: Want to reduce labeling cost? GPT-3 can help. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 4195–4205. ACL (2021)



Discovering Sound Free-Choice Workflow Nets with Non-block Structures

Tsung-Hao Huang^(✉)  and Wil M. P. van der Aalst 

Process and Data Science (PADS), RWTH Aachen University, Aachen, Germany
{[tsunghao.huang](mailto:tsunghao.huang@pads.rwth-aachen.de),[wvdaalst](mailto:wvdaalst@pads.rwth-aachen.de)}@pads.rwth-aachen.de

Abstract. Process discovery aims to discover models that can explain the behaviors of event logs extracted from information systems. While various approaches have been proposed, only a few guarantee desirable properties such as soundness and free-choice. State-of-the-art approaches that exploit the representational bias of process trees to provide the guarantees are constrained to be block-structured. Such constructs limit the expressive power of the discovered models, i.e., only a subset of sound free-choice workflow nets can be discovered. To support a more flexible structural representation, we aim to discover process models that provide the same guarantees but also allow for non-block structures. Inspired by existing works that utilize synthesis rules from the free-choice nets theory, we propose an automatic approach that incrementally adds activities to an existing process model with predefined patterns. Playing by the rules ensures that the resulting models are always sound and free-choice. Furthermore, the discovered models are not restricted to block structures and are thus more flexible. The approach has been implemented in Python and tested using various real-life event logs. The experiments show that our approach can indeed discover models with competitive quality and more flexible structures compared to the existing approach.

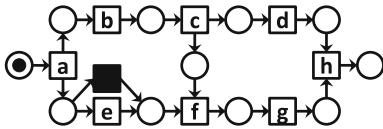
Keywords: Process discovery · Free-choice net · Synthesis rules

1 Introduction

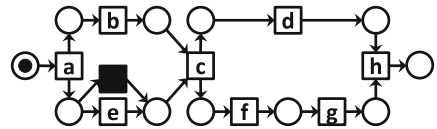
Process discovery aims to construct process models that reflect the behaviors of a given event log extracted from information systems [2]. As it is a non-trivial problem, many challenges remain. In most cases, the one and only “best model” does not exist as there are trade-offs among the four model quality metrics, namely fitness, precision, generalization, and simplicity [2]. In addition to the quality metrics, there exist properties that one would like to have for the discovered models. One of the important properties is being a sound workflow net as soundness ensures the absence of deadlocks, proper completion, etc. [1] and it is a prerequisite for many crucial automated analyses such as conformance checking. The other desirable structural property is being free-choice [3]. In free-choice nets, choices and synchronizations are separated. This provides an easy conversion between the discovered models and many process modeling languages such

as Business Process Modeling Notation (BPMN) since the equivalent constructs (dedicated split and join connectors) are naturally embedded. Furthermore, free-choice nets have been studied extensively and thus supported by an abundance of theories [10], which provide efficient analysis techniques.

While various discovery algorithms have been proposed, only a handful of them provides such guarantees. State-of-the-art discovery algorithms like the Inductive Miner (IM) [15] are able to discover sound free-choice workflow nets by exploiting its representational bias. However, due to the same reason, the discovered models are constrained to be block-structured. This limits the expressive power of such models, i.e., only a subset of the sound free-choice workflow nets can be discovered. As an example, Fig. 1a shows a sound free-choice workflow net (with non-block structures) discovered by our approach¹. The same language can never be expressed by the model discovered by IM, as shown in Fig. 1b.



(a) A model discovered by our approach. The same language cannot be expressed by the models discovered using the Inductive Miner, which uses process trees internally.



(b) A model discovered by the IM using the log generated by the model in (a). The two branches before c need to be synchronized first before d can be executed.

Fig. 1. Examples showing the need for the non-block process models discovery. Note that the trace $\langle a, b, c, d, e, f, g, h \rangle$ that is possible in (a) cannot be replayed by (b).

In this paper, we aim to discover sound free-choice workflow nets with non-block structures. Inspired by the interactive process discovery approach in [11, 12], we develop an automatic process discovery algorithm that incrementally adds activities to an existing net using synthesis rules [10]. Since checking the feasibility for the application of the synthesis rules is computationally expensive, we use log heuristics to locate the most possible position for the to-be-added activity on the existing process model instead of evaluating all possible applications of synthesis rules as in [11]. Additionally, we identify the need for an additional rule and extend the set of patterns introduced in [12].

Playing by the rules ensures that the discovered process models by our approach are guaranteed to be sound free-choice workflow nets [10, 11]. Moreover, the discovered models are not constrained to block structures. Last but not least, the level of replay fitness is guaranteed via a threshold set by the users. The approach has been implemented in Python and evaluated using various public-available real-life event logs. The evaluation shows that our approach is able to discover non-block structured models with competitive qualities compared to the state-of-the-art discovery algorithm.

¹ The proposed approach has dedicated silent transitions for start and end as defined later in Definition 5. We dropped them here for ease of comparison.

The remainder of the paper is organized as follows. We review the related work in Sect. 2 and introduce necessary concepts in Sect. 3. Section 4 introduces the approach. Section 5 presents the experiment and Sect. 6 concludes the paper.

2 Related Work

An overview of process discovery is out of the scope of this paper, we refer to [7, 14] for more details. In this section, we focus on process discovery techniques that guarantee soundness (and free-choice) properties. Approaches like [6, 8] can discover non-block structured models but cannot guarantee both properties. While Split Miner discovers models that are deadlock-free, they are not necessarily sound [8].

The family of Inductive Miner (IM) algorithms [15] guarantees sound and free-choice of the discovered models by exploiting the representational bias of the process tree. By design, a process tree represents a sound workflow net. It is a rooted tree where the leaf nodes are activities and the non-leaf nodes are the operators. The hierarchical representation has a straightforward translation to Petri net. However, the resulting models are limited to being block-structured as a process tree can only represent process models that can be separated into parts that have a single entry and exit [15]. Consequently, process trees can only represent a subset of sound workflow nets. The same arguments hold for approaches that are based on process trees such as the Evolutionary Tree Miner (ETM) [9] and the recently developed incremental process discovery approach [16].

Applying the synthesis rules [10], the interactive process discovery approaches developed in [11–13] ensure soundness and free-choice properties. A semi-automatic interactive tool, ProDiGy, is proposed in [12] to recommend the best possible ways to add an activity to an existing model.

Our approach differs from [11–13] in several ways. First, we adopt an automatic setting as the order of adding activities is predetermined and the best modification to the existing net is selected based on the model quality. Second, we use log heuristics to locate the most suitable position for adding the new activity instead of evaluating all the possibilities of synthesis rules applications. Moreover, we identify the need for a new rule as the desired models often cannot be discovered without going back and forth by a combination of reduction and synthesis rules [13]. Lastly, the set of patterns is extended and formally defined.

3 Preliminaries

We denote the set of all sequences over some set A as A^* , the power set of A as $\mathcal{P}(A)$, and the set of all multisets over A as $\mathcal{B}(A)$. For some multiset $b \in \mathcal{B}(A)$, $b(a)$ denotes the number of times $a \in A$ appears in b . For a given sequence $\sigma = \langle a_1, a_2, \dots, a_n \rangle \in A^*$, $|\sigma| = n$ is the length of σ and $dom(\sigma) = \{1, 2, \dots, |\sigma|\}$ is the domain of σ . $\langle \rangle$ is the empty sequence. $\sigma(i) = a_i$ denotes the i -th element of σ . Given sequences σ_1 and σ_2 , $\sigma_1 \cdot \sigma_2$ denotes the concatenation of the two. Let A be a set and $X \subseteq A$ be a subset of A . For $\sigma \in A^*$ and $a \in A$,

we define $\downarrow_X \in A^* \rightarrow X^*$ as a projection function recursively with $\langle \rangle \downarrow_X = \langle \rangle$, $\langle a \rangle \cdot \sigma \downarrow_X = \langle a \rangle \cdot \sigma \downarrow_X$ if $a \in X$ and $\langle a \rangle \cdot \sigma \downarrow_X = \sigma \downarrow_X$ if $a \notin X$. For example, $\langle x, y, x \rangle \downarrow_{\{x,z\}} = \langle x, x \rangle$. Projection can also be applied to multisets of sequences, e.g., $[\langle a, b, a \rangle^6, \langle a, b, c \rangle^6, \langle b, a, c \rangle^2] \downarrow_{\{b,c\}} = [\langle b \rangle^6, \langle b, c \rangle^8]$.

Definition 1 (Trace, Log). A trace $\sigma \in \mathcal{U}_A^*$ is a sequence of activities, where \mathcal{U}_A is the universe of activities. A log $L \in \mathcal{B}(\mathcal{U}_A^*)$ is a multiset of traces.

Definition 2 (Log Properties). Let $L \in \mathcal{B}(\mathcal{U}_A^*)$ and $a, b \in \mathcal{U}_A$.

- $\#(a, L) = \sum_{\sigma \in L} |\{i \in \text{dom}(\sigma) \mid \sigma(i) = a\}|$ is the times a occurred in L .
- $\#(a, b, L) = \sum_{\sigma \in L} |\{i \in \text{dom}(\sigma) \setminus \{|\sigma|\} \mid \sigma(i) = a \wedge \sigma(i+1) = b\}|$ is the number of direct successions from a to b in L .
- $\text{caus}(a, b, L) = \begin{cases} \frac{\#(a,b,L) - \#(b,a,L)}{\#(a,b,L) + \#(b,a,L) + 1} & \text{if } a \neq b \\ \frac{\#(a,b,L)}{\#(a,b,L) + 1} & \text{if } a = b \end{cases}$ is the strength of causal relation (a, b) .
- $A_c^{\text{pre}}(a, L) = \{a_{\text{pre}} \in \mathcal{U}_A \mid \text{caus}(a_{\text{pre}}, a, L) \geq c\}$ is the set of a 's preceding activities, determined by threshold c .
- $A_c^{\text{fol}}(a, L) = \{a_{\text{fol}} \in \mathcal{U}_A \mid \text{caus}(a, a_{\text{fol}}, L) \geq c\}$ is the set of a 's following activities, determined by threshold c .
- $A^s(L) = \{\sigma(1) \mid \sigma \in L \wedge \sigma \neq \langle \rangle\}$ is the set of start activities in L .
- $A^e(L) = \{\sigma(|\sigma|) \mid \sigma \in L \wedge \sigma \neq \langle \rangle\}$ is the set of end activities in L .

Definition 3 (Petri Net, Labeled Petri Net). A Petri net $N = (P, T, F)$ is a tuple, where P is the set of places, T is the set of transitions, $P \cap T = \emptyset$, and $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs. A labeled Petri net $N = (P, T, F, l)$ is a Petri net (P, T, F) with a labeling function $l \in T \rightarrow \mathcal{U}_A$ that maps a subset of transitions to activities. A $t \in T$ is called invisible if t is not in the domain of l .

For any $x \in P \cup T$, $\bullet^N x = \{y \mid (y, x) \in F\}$ denotes the set of input nodes and $x^N \bullet = \{y \mid (x, y) \in F\}$ denotes the set of output nodes. The superscript N is dropped if it is clear from the context. The notation can be generalized to set. For any $X \subseteq P \cup T$, $\bullet X = \{y \mid \exists x \in X (y, x) \in F\}$ and $X \bullet = \{y \mid \exists x \in X (x, y) \in F\}$.

Definition 4 (Free-choice Net). Let $N = (P, T, F)$ be a Petri net. N is a free-choice net if for any $t_1, t_2 \in T$: $\bullet t_1 = \bullet t_2$ or $\bullet t_1 \cap \bullet t_2 = \emptyset$.

Definition 5 (Workflow Net (WF-net) [1,11]). Let $N = (P, T, F, l)$ be a labeled Petri net. $W = (P, T, F, l, p_s, p_e, \top, \perp)$ is a WF-net iff (1) it has a dedicated source place $p_s \in P$: $\bullet p_s = \emptyset$ and a dedicated sink place $p_e \in P$: $p_e \bullet = \emptyset$ (2) $\top \in T$: $\bullet \top = \{p_s\} \wedge p_s \bullet = \{\top\}$ and $\perp \in T$: $\perp \bullet = \{p_e\} \wedge p_e \bullet = \{\perp\}$ (3) every node x is on some path from p_s to p_e , i.e., $\forall x \in P \cup T (p_s, x) \in F^* \wedge (x, p_e) \in F^*$, where F^* is the reflexive transitive closure of F .

Definition 6 (Short-circuited WF-net [1]). Let $W = (P, T, F, l, p_s, p_e, \top, \perp)$ be a WF-net. The short-circuited WF-net of W , denoted by $SC(W)$, is constructed by $SC(W) = (P, T \cup \{t'\}, F \cup \{(\perp, t'), (t', \top)\}, l, p_s, p_e, \top, \perp)$, where $t' \notin T$.

Definition 7 (Paths, Elementary Paths). A path of a Petri net $N = (P, T, F)$ is a non-empty sequence of nodes $\rho = \langle x_1, x_2, \dots, x_n \rangle$ such that $(x_i, x_{i+1}) \in F$ for $1 \leq i < n$. ρ is an elementary path if $x_i \neq x_j$ for $1 \leq i < j \leq n$.

Definition 8 (Incidence Matrix [10]). Let $N = (P, T, F)$ be a Petri net. The incidence matrix $\mathbf{N} : (P \times T) \rightarrow \{-1, 0, 1\}$ of N is defined as

$$\mathbf{N}(p, t) = \begin{cases} 0 & \text{if } (p, t) \notin F \wedge (t, p) \notin F \vee ((p, t) \in F \wedge (t, p) \in F) \\ -1 & \text{if } (p, t) \in F \wedge (t, p) \notin F \\ 1 & \text{if } (p, t) \notin F \wedge (t, p) \in F \end{cases}$$

For a Petri net $N = (P, T, F)$ and its corresponding incidence matrix \mathbf{N} , we use $\mathbf{N}(p)$ to denote the row vector of the corresponding $p \in P$ and $\mathbf{N}(t)$ to denote the column vector of the corresponding $t \in T$.

Definition 9 (Linearly Dependent Nodes [10]). Let $N = (P, T, F)$ be a Petri net. \mathbb{Q} is the set of rational numbers. A place p is linearly dependent if there exists a row vector $\mathbf{v} : P \rightarrow \mathbb{Q}$ such that $\mathbf{v}(p) = 0$ and $\mathbf{v} \cdot \mathbf{N} = \mathbf{N}(p)$. A transition t is linearly dependent if there exists a column vector $\mathbf{v} : T \rightarrow \mathbb{Q}$ such that $\mathbf{v}(t) = 0$ and $\mathbf{v} \cdot \mathbf{N} = \mathbf{N}(t)$.

Definition 10 (Synthesis Rules [10,11]). Let W and W' be two free-choice workflow nets, and let $SC(W) = (P, T, F, l, p_s, p_e, \top, \perp)$ and $SC(W') = (P', T', F', l', p_s, p_e, \top, \perp)$ be the corresponding short-circuited WF-nets:

- Linear Dependent Place Rule ψ_P : W' is derived from W using ψ_P , i.e., $(W, W') \in \psi_P$ if (1) $T' = T$, $P' = P \cup \{p\}$ and $p \notin P$ is linear dependent in $SC(W')$, $F' = F \cup \tilde{F}$ where $\tilde{F} \subseteq ((\{p\} \times T) \cup (T \times \{p\}))$ (2) Every siphon in $SC(W')$ contains p_s .
- Linear Dependent Transition Rule ψ_T : W' is derived from W using ψ_T , i.e., $(W, W') \in \psi_T$ if $P' = P$, $T' = T \cup \{t\}$ and $t \notin T$ is linear dependent in $SC(W')$ and $F' = F \cup \tilde{F}$ where $\tilde{F} \subseteq ((P \times \{t\}) \cup (\{t\} \times P))$, and $\forall t \in T \cap T'. l(t) = l'(t)$.
- Abstraction Rule ψ_A : $(W, W') \in \psi_A$ if (1) there exists a set of transitions $R \subseteq T$ and a set of places $S \subseteq P$ such that $(R \times S \subseteq F) \wedge (R \times S \neq \emptyset)$. (2) $SC(W')$ is constructed by adding an additional place $p \notin P$ and a transition $t \notin T$ such that $P' = P \cup \{p\}$, $T' = T \cup \{t\}$, $F' = (F \setminus (R \times S)) \cup ((R \times \{p\}) \cup (\{p\} \times \{t\}) \cup (\{t\} \times S))$, and $\forall t \in T \cap T'. l(t) = l'(t)$.

Applying the three synthesis rules (ψ_P, ψ_T, ψ_A) to derive W' from a sound free-choice workflow net W ensures that W' is also sound [11,13]. Three properties need to be hold for a WF-net to be sound (1) safeness: places cannot hold multiple tokens at the same time (2) option to complete: it is always possible to reach the marking in which only the sink place is marked. (3) no dead transitions. Next, we introduce the initial net [11] and show some examples of synthesis rules applications.

Definition 11 (Initial Net [13]). Let $W = (P, T, F, l, p_s, p_e, \top, \perp)$ be a free-choice WF-net. W is an initial net if $P = \{p_s, p_1, p_e\}$, $T = \{\top, \perp\}$, $F = \{(p_s, \top), (\top, p_1), (p_1, \perp), (\perp, p_e)\}$.

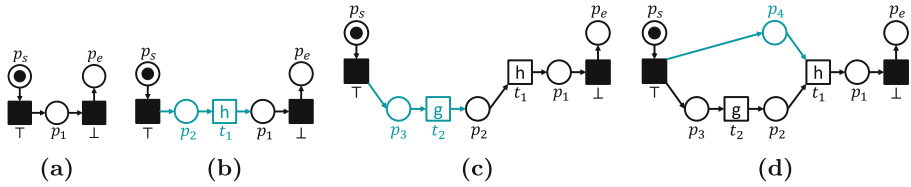


Fig. 2. Examples of synthesis rules applications starting from (a) The initial net. (b) Using ψ_A , p_2 and t_1 are added to the initial net with $R = \{T\}$ and $S = \{p_1\}$. (c) Using ψ_A , p_3 and t_2 are added to previous net with $R = \{T\}$ and $S = \{p_2\}$. (d) p_4 is added using ψ_p as p_4 is a linear combination of p_3 and p_2 .

The initial net is shown in Fig. 2a. Clearly, it is a sound free-choice workflow net. Starting from the initial net, one can incrementally add additional nodes according to the synthesis rules. Figure 2 shows example applications of synthesis rules starting from the initial net.

4 Approach

With the necessary concepts introduced, we are now ready to introduce the approach. We start by showing the basic idea of the approach with the help of Fig. 3 before diving into each step in detail. Internally, the approach incrementally adds a new activity to an existing net. The figure shows a single iteration. In each iteration, we have an existing model from the previous iteration and a log projected on the already added activities so far and the to-be-added one.

We start by locating the most likely position to add the new activity determined by log heuristics. The result of this step is a subset of nodes of the existing model. The set of nodes will then be used to prune the search space. Then, the predefined patterns are applied to the existing net to get a set of candidate nets. Lastly, we select the best net (next existing net) out of the candidates in terms of fitness and precision. Note that the existing net in the first iteration is initiated by the initial net (Definition 11). As a running example, consider the corresponding log that is used to discover the Petri net in Fig. 1 by our approach: $L_s = [\langle a, b, c, d, f, g, h \rangle^{22}, \langle a, b, c, f, d, g, h \rangle^{14}, \langle a, e, b, c, d, f, g, h \rangle^{13}, \langle a, e, b, c, f, d, g, h \rangle^{13}, \langle a, e, b, c, f, g, d, h \rangle^{10}, \langle a, b, c, f, g, d, h \rangle^{10}, \langle a, b, e, c, d, f, g, h \rangle^6, \langle a, b, e, c, f, g, d, h \rangle^3, \langle a, b, e, c, f, d, g, h \rangle^3, \langle a, b, c, d, e, f, g, h \rangle^2, \langle a, b, c, e, d, f, g, h \rangle^2, \langle a, b, c, e, f, g, d, h \rangle^1, \langle a, b, c, e, f, d, g, h \rangle^1]$. The instances provided in Fig. 3 shows the 3rd iteration for the running example L_s . In the following subsections, we introduce the details of each step.

4.1 Ordering Strategies for Adding Activities

Before starting any iteration, we need to come up with an order for adding activities based on a given log L . It is important as the quality of the discovered models often depends on the order of adding activities [11]. Moreover, in combination with the search space pruning, it can influence the computation time for

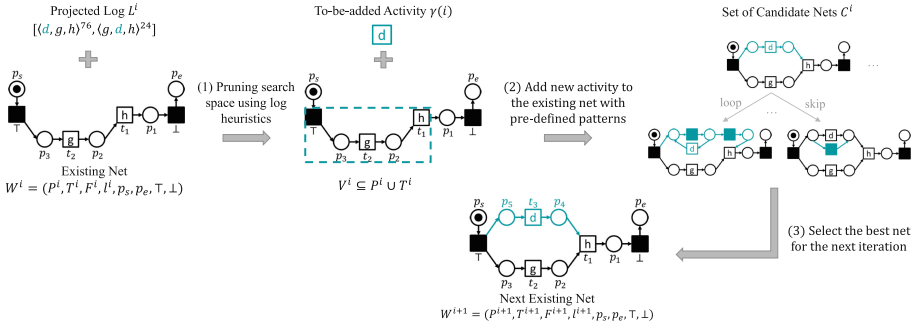


Fig. 3. An example of a single iteration of our approach.

each iteration significantly. In this paper, we introduce two ordering strategies. The first one is relatively straightforward. The activities in L are simply ordered by their frequency.

Definition 12 (Activities-Adding Order, Frequency-Based Ordering). Let $L \in \mathcal{B}(\mathcal{U}_A^*)$ and $A = \bigcup_{\sigma \in L} \{a \in \sigma\}$. $\gamma \in A^*$ is an activities-adding order for L if $\{a \in \gamma\} = A$ and $|\gamma| = |A|$. The frequency-based ordering is $order_{freq}(L) = \gamma$ such that γ is an activities-adding order and $\forall_{1 \leq i < j \leq |\gamma|} \#(\gamma(i), L) \geq \#(\gamma(j), L)$.

The second ordering strategy is similar to the Breadth-first Search (BFS) algorithm. The advantage of this is that it also considers the closeness between activities in the log, rather than just frequency. To explain this ordering strategy, we first define a sub-function.

Definition 13 (Directly-Precedes Activities Sorting). Let $L \in \mathcal{B}(\mathcal{U}_A^*)$ and $a \in \mathcal{U}_A$. $A = \{b \in \mathcal{U}_A \mid \#(b, a, L) > 0\}$ is the set of activities directly-precede a in L at least once and $\sigma \in A^*$. Directly-precedes activities sorting is $sortPreceded(a, L) = \sigma$ such that $\{b \in \sigma\} = A$ and $|\sigma| = |A|$ and $\forall_{1 \leq i < j \leq |\sigma|} \#(\sigma(i), a, L) \geq \#(\sigma(j), a, L)$.

The function $sortPreceded$ takes an activity a and a log L to return a sequence of a 's directly-preceded activities b that are sorted by the frequency of $\#(b, a, L)$. Finally, we can define the BFS-based ordering strategy.

Definition 14 (Breadth-First-Search-Based Ordering). Let $L \in \mathcal{B}(\mathcal{U}_A^*)$ and $A = \bigcup_{\sigma \in L} \{a \in \sigma\}$. BFS-based ordering is defined as $order_{BFS}(L) = \gamma$, where γ is an activities adding order for L and $\gamma = \gamma_1 \cdot \gamma_2 \cdot \dots \cdot \gamma_{|\gamma|}$, for each $1 \leq j \leq |\gamma|$,

$$\gamma_j = \begin{cases} order_{freq}(L \upharpoonright_{A^e(L)}) & \text{if } j = 1 \\ sortPreceded(\gamma(j-1), L) \upharpoonright_{A \setminus \{\gamma(1), \gamma(2), \dots, \gamma(j-1)\}} & \text{otherwise} \end{cases}$$

The function starts by sorting the end activities $A^e(L)$ according to their frequency in the log. Then, it enumerates through the sequence γ and sorts the preceded activities of $\gamma(j-1)$ by the frequency of direct successions. The projection function in the second case of Definition 14 filters out the activities that are already in γ .

Compared to the frequency-based ordering, the BFS-based ordering considers the closeness of the activities. This allows us to add activities that are close together. Together with the effect of the search space pruning, it is expected that BFS-based ordering would have less computation time. Applying the function $order_{BFS}$ to our running example, L_s , we get the activities adding order as $\gamma = order_{BFS}(L_s) = \langle h \rangle \cdot \langle g, d \rangle \cdot \langle f \rangle \cdot \langle c, e \rangle \cdot \langle \rangle \cdot \langle b \rangle \cdot \langle a \rangle \cdot \langle \rangle$. γ is then used to determine the order of adding activities. Given the activities adding order γ , we define the artifacts for each iteration i as followed.

Definition 15 (Projected Log). *Let $L \in \mathcal{B}(\mathcal{U}_A^*)$ and γ be a activities adding order for L . The projected log for L in the i -th iteration is $L^i = L \upharpoonright_{\{\gamma(1), \gamma(2), \dots, \gamma(i)\}}$.*

For instance, the projected log for the running example L_s for the 3rd iteration is then $L_s^3 = L_s \upharpoonright_{\{h, g, d\}} = [\langle d, g, h \rangle^{76}, \langle g, d, h \rangle^{24}]$, as shown in Fig. 3. The to-be-added activity is denoted as $\gamma(i)$, which is $\gamma(3) = d$ for the 3rd iteration. Also, we denote the existing sound free-choice workflow net for iteration i as W^i . Note that for the running example, W^1 , W^2 , and W^3 are visualized in Fig. 2a, 2b, and 2c, respectively.

4.2 Search Space Pruning

As checking the feasibility of applying linear dependent rules ψ_T, ψ_P is computationally expensive [11], it is impractical to compute all possible applications of the synthesis rules. Also, some of them are not of interest. For example, as shown in Fig. 3, it is clear that the to-be-added activity d never happens after h in the projected log. Using such information, we can already eliminate the constructs (applications of synthesis rules) that allow activity d to be executed after h . Therefore, in each iteration i we start by locating the most likely position to add $\gamma(i)$. This helps us to restrict the application of synthesis rules on only a subset of nodes, denoted as $V^i \subseteq P^i \cup T^i$, in the existing net W^i . To do that, we first identify the set of preceding and following activities of $\gamma(i)$ in the projected log L^i , which would be $A_c^{pre}(\gamma(i), L^i)$ and $A_c^{fol}(\gamma(i), L^i)$ respectively. Recall that c is a threshold for the causal relation and can be given by users as an input. We use $c = 0.9$ as the default value. Then, the corresponding labelled transitions are identified in W^i . Finally, V^i is the set of all the nodes on the elementary paths from the preceding transitions to the following transitions. If $A_c^{pre}(\gamma(i), L)/A_c^{fol}(\gamma(i), L)$ is an empty set, we use the \top/\perp transitions. For instance, in Fig. 3, we identify that $A_c^{pre}(d, L_s^3) = \emptyset$ and $A_c^{fol}(d, L_s^3) = \{h\}$. Therefore, we find all the nodes on the elementary paths between every node in $\{\top\}$ and every node in $\{t_1\}$. As a result, the set $V^3 = \{\top, p_3, t_2, p_2, t_1\}$ is used to prune the search space, i.e., to constrain the application of synthesis rules.

Constraining Synthesis Rules. For the abstraction rule ψ_A , this means that the set of transitions R and the set of places S used as the preconditions for applying ψ_A need to be a subset of V , i.e., $S \subseteq V \wedge R \subseteq V$. For the linear dependent rules ψ_P/ψ_T , the new place/transition (p'/t') cannot have arcs connected to any node outside V . This shortens the computation time as certain rules applications can be removed and there is no need to check their feasibility.

4.3 Patterns

In this section, we introduce the patterns that are used to add activity $\gamma(i)$ to the existing free-choice workflow net W^i . First, we motivate the need for an additional rule.

The Need for an Additional Rule. It is proven that any sound free-choice workflow net can be constructed by the three synthesis rules ψ_A, ψ_P, ψ_T [10,11]. However, when applying to discover process models, the desirable model is often not possible to derive due to the existing construct. An example is shown in Fig. 4. While it is possible to add a transition labeled by a in Fig. 4a, it is not possible to derive the same net in Fig. 4b as there is no rule allowing such transformation. One possible workaround is to go back and forth by a combination of reduction and synthesis rules as suggested in [13]. However, once the existing net becomes more complex, such a solution becomes infeasible to track.

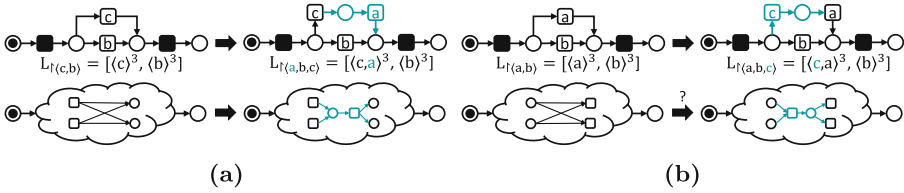


Fig. 4. Examples showing the motivation for the dual abstraction rule ψ_D . Although the desirable nets on the right-hand side of (a) and (b) are semantically the same, the existing synthesis rules only allow the transformation in (a). There is no rule defined for the transformation in (b).

We observe that in many situations (including the example in Fig. 4b), the desired models cannot be constructed because there is no rule allowing one to introduce a new transition t and a new place p in between a set of places S and a set of transitions R that are fully connected, i.e., $S \times R \subseteq F$. Therefore, we define the dual abstraction rule to allow such construct.

Definition 16 (Dual Abstraction Rule ψ_D). Let $W = (P, T, F, l, p_s, p_e, \top, \perp)$ and $W' = (P', T', F', l', p_s, p_e, \top, \perp)$ be two free-choice workflow nets. $(W, W') \in \psi_D$ if (1) there exists a set of places $S \subseteq P$ and a set of transitions $R \subseteq T$ such that $S \times R \subseteq F \wedge S \times R \neq \emptyset$. (2) W' is constructed by adding an additional transition $t \notin T$ and a place $p \notin P$ such that $P' = P \cup \{p\}, T' = T \cup \{t\}, F' = (F \setminus (S \times R)) \cup ((S \times \{t\}) \cup (\{t\} \times \{p\}) \cup (\{p\} \times R))$, and $\forall t \in T \cap T' l(t) = l'(t)$.

As we are only interested in sound free-choice workflow nets, we need to make sure that the dual abstraction rule ψ_D preserves soundness.

Proposition 1 (ψ_D preserves soundness). Let $W = (P, T, F, l, p_s, p_e, \top, \perp)$, $W' = (P', T', F', l', p_s, p_e, \top, \perp)$ be free-choice workflow nets, and W' is derived from W using ψ_D , i.e., $(W, W') \in \psi_D$. Then W' is sound if W is sound.

Proof. Let $t' \in T' \setminus T$ and $p' \in P' \setminus P$ be the new transition and place in W' . Let $R = p' \bullet$ and $S = \bullet t'$. The new net W' is free-choice in only two cases. Either $S = \overset{W}{\bullet} R$ or $R = S \bullet$. In either case, any reachable marking in $(W, [p_s])$ that does not need to fire $t_R \in R$ is still reachable in $(W', [p_s])$. Also, the reachable markings in $(W, [p_s])$ that need to fire $t_R \in R$ can be reached in $(W', [p_s])$ as one can just add t' somewhere before t_R in the corresponding firing sequence. Then, it is trivial to see that W' fulfils the three conditions of soundness if W is also sound. \square

Next, we extend the linear dependent place rule ψ_P . As we aim to add a transition labeled by $\gamma(i)$ to the existing labeled free-choice workflow net W^i , only adding a place p' by ψ_P does not suffice. Hence, in our approach, an application of ψ_P is always coupled with a directly followed application of abstraction rule ψ_A to include a transition. ψ_A is applied between the added place p' and its preset $\bullet p'$. This is possible because every transition in $\bullet p'$ is connected to every place in $\{p'\}$ by definition, which satisfies the precondition of ψ_A . An example is shown in Fig. 5, p_5 and t_3 are added by ψ_A directly after the addition of p_4 by ψ_P . To be more precise, we define the extended rule, ψ'_P , that describes the pattern.

Definition 17 (Extended Linear Dependent Place Rule ψ'_P). Let $W=(P, T, F, l, p_s, p_e, \top, \perp)$ and $W''=(P'', T'', F'', l'', p_s, p_e, \top, \perp)$ be free-choice workflow nets. $(W, W'') \in \psi'_P$ if (1) $\exists W'=(P', T', F', l', p_s, p_e, \top, \perp) (W, W') \in \psi_P \wedge (W', W'') \in \psi_A$ and (2) $\exists ! p^* \in P'' (\{p^*\} = P' \setminus P) \wedge ((T'' \setminus T') \times \{p^*\}) \subset F'' \wedge ((T' \times \{p^*\}) \not\subset F'')$.

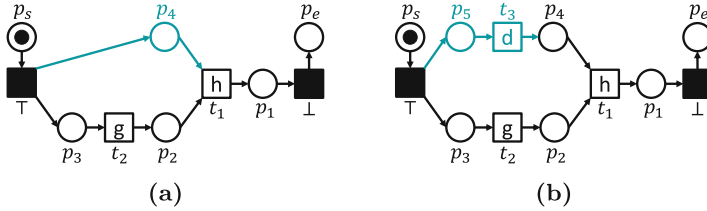


Fig. 5. (a) ψ_P adds a place p_4 . (b) As every transition in $\bullet p_4$ has an arc to every place in $\{p_4\}$, one can directly apply ψ_A to add p_5 and t_3 .

Then, we define the set of nets constructed by every possible single application of the rules $\psi_A, \psi'_P, \psi_T, \psi_D$.

Definition 18 (Base Candidates Set). Let $W=(P, T, F, l, p_s, p_e, \top, \perp)$, $W'=(P', T', F', l', p_s, p_e, \top, \perp)$ be free-choice workflow nets. Let $X=(P' \cup T') \setminus (P \cup T)$, $V \subseteq P \cup T$, $V'=(P \cup T) \setminus V$, and let $a \in \mathcal{U}_A$ be an activity label. The base candidates set is $base(W, V, a) = \{W' | ((W, W') \in (\psi_A \cup \psi_T \cup \psi'_P \cup \psi_D)) \wedge (\nexists x \in X (\{x\} \times V') \cup (V' \times \{x\}) \subseteq F') \wedge (l' = l \cup ((T' \setminus T) \times \{a\}))\}$.

The base candidates set $C_{base}^i = base(W^i, V^i, \gamma(i))$ consists of the nets that are constructed by every possible single application of the rules $\psi_A, \psi'_P, \psi_T, \psi_D$ to add a transition labeled by $\gamma(i)$ to W^i considering the constraints on V^i .

Next, we introduce three patterns that make a transition skippable, in a strict loop, or in an optional (tau) loop. A transition in a strict loop means that the execution of the transition is required, otherwise it is an optional loop.

Definition 19 (Pattern-Building Functions). *Let $W=(P, T, F, l, p_s, p_e, \top, \perp)$ and $W'=(P', T', F', l', p_s, p_e, \top, \perp)$ be two free-choice workflow nets. Let $a \in \mathcal{U}_A$ be an activity label and $t_a \in T : l(t_a) = a$ be the corresponding transition in W . We define the three pattern-building functions² as*

- $skip(W, a) = W'$ such that
 - $(W, W') \in \psi_T$
 - $F' = F \cup (\{t'\} \times t_a \bullet) \cup (\bullet t_a \times \{t'\})$ (where $t' \in T' \setminus T$)
 - $l' = l$ (t' is a silent transition)
- $loop_s(LW, a)$ is defined by two cases:
 1. if $\nexists t^* \in ((t_a \bullet) \bullet) (|\bullet t^*| > 1) \wedge (\bullet t^* \setminus t_a \bullet \neq \emptyset)$, then $loop_s(W, a) = W'$ such that
 - $(W, W') \in \psi_T$
 - $F' = F \cup (t_a \bullet \times \{t'\}) \cup (\{t'\} \times \bullet t_a)$ (where $t' \in T' \setminus T$)
 - $l' = l$ (t' is a silent transition)
 2. otherwise, return $loop_s(W', a)$ such that
 - $(W, W') \in \psi_A$
 - $((\{t_a\} \times (P' \setminus P)) \in F') \wedge ((\{t_a\} \times P) \notin F')$
 - $l' = l$
- $loop_\tau(W, a)$ is defined by two cases:
 1. if $\nexists t^* \in ((t_a \bullet) \bullet) (|\bullet t^*| > 1) \wedge (\bullet t^* \setminus t_a \bullet \neq \emptyset)$, then $loop_\tau(W, a) = W'$ such that
 - $(W, W') \in \psi_T$
 - $F' = F \cup (t_a \bullet \times \{t'\}) \cup (\{t'\} \times \bullet t_a)$ (where $t' \in T' \setminus T$)
 - $l' = (l \setminus \{(t_a, a)\}) \cup \{(t', a)\}$ (the labels of t_a and t' are swapped)
 2. otherwise, return $loop_\tau(W', a)$ such that
 - $(W, W') \in \psi_A$
 - $((\{t_a\} \times (P' \setminus P)) \in F') \wedge ((\{t_a\} \times P) \notin F')$
 - $l' = l$

The second case of the loop functions is there to keep the free-choice property. To illustrate the ideas using the running example, consider the net shown in Fig. 6a as the input net W and t_3 (labeled by d) is the transition for which we are going to apply the functions to derive patterns. Figure 6b shows that function $skip(W, d)$ simply adds a silent transition t_4 with the same connection as t_3 to W . Figure 6c and 6d show an application of $loop_s(W, d)$ and illustrate the need for the two cases for the loop functions. As shown in Fig. 6c, the second case of $loop_s$ is applied since there exists a transition $t_1 \in ((t_3 \bullet) \bullet)$ with more than one place in its preset ($|\bullet t_1| > 1$) and $\bullet t_1 \setminus t_3 \bullet \neq \emptyset$. Therefore, W' (Fig. 6c) is first constructed by adding p_6 and t_4 . Then, the function returns $loop_s(W', d)$. Now, the first case should be applied. In this case, t_5 is added with the reverse connections of t_3 . As indicated, the second case in the loop functions helps to keep the

² The input/output nodes notations (\bullet) used in Definition 19 refer to the input net W . We drop the superscript for readability.

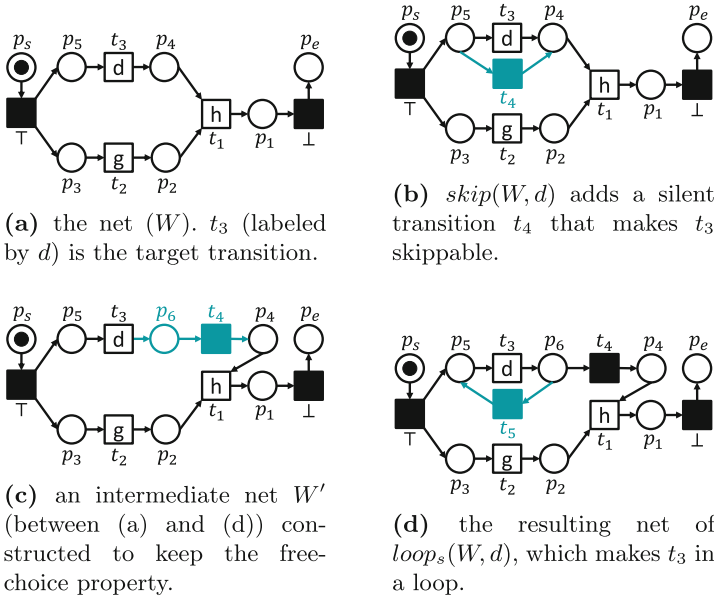


Fig. 6. Examples showing how the functions are applied to derive patterns.

free-choice property. Imagine a net that is constructed by adding t' to the net in Fig. 6a with connections (p_4, t') and (t', p_5) . Such a net makes t_3 in a loop but it is no longer a free-choice net. The constructs of $loop_s$ and $loop_\tau$ are almost the same, the difference is that the labels of t_3 and the silent transition t_5 are swapped.

Finally, to get the set of candidate nets C^i , we apply the three pattern-building functions to every net $W \in C_{base}^i$. Observe that all the nets in Fig. 6 are elements of C^3 .

4.4 Selection and Fall-Through

Selection. In the last step, we select the next existing net W^{i+1} from the set of candidates C^i evaluated by the projected $\log L^i$. The selection is done in a stepwise manner. We first try to filter out the candidates that do not reach a user-defined replay fitness threshold θ and then select the best net out of the rest in terms of F1 score, which is calculated as the harmonic mean of fitness and precision. We use alignment-based fitness [4] and precision [5].

Fall-Through. If none of the nets in C^i reach the fitness threshold θ , we adopt a fall-through. This is done by going back to Step 2, where $\gamma(i)$ is added to $W^i = (P^i, T^i, F^i, l^i, p_s, p_e, \top, \perp)$, but without the constraints of V^i . This can also be seen as setting $V^i = P^i \cup T^i$. In this case, a new place p' with arcs $\{(\top, p'), (p', \perp)\}$ can be always added by ψ_P as p is linear dependent on p_s and p_e . Then, the patterns building functions can be applied to ensure that the fitness threshold θ is guaranteed in every iteration.

5 Evaluation

In this section, we present the experiments conducted to evaluate our approach. The presented approach in this paper is implemented in Python using PM4Py³ and can be accessed here⁴. As mentioned, the algorithm takes as inputs a log and three parameters including two thresholds θ, c , and the types of ordering strategy. Using this implementation, we conduct three experiments to address the following questions (1) How effective are the pre-defined patterns? (2) What are the effects of the ordering strategy on the model quality and the execution time? (3) Can the model quality be improved by the non-block structures?

5.1 Experiment Setup

Dataset: We use four public available real-life event logs, which are BPI2017⁵, helpdesk⁶, hospitalBilling⁷, and traffic⁸ respectively. BPI2017 is split into two sub logs, BPI2017A and BPI2017O, using the event prefixes. To focus on the mainstream behaviors, the logs are filtered to include at least 95% of the cases.

Experiment 1 (Effectiveness of Patterns): The first experiment aims to evaluate how effective are the pre-defined patterns. As our approach is based on [11], this can be evaluated by comparing the quality of the intermediate models of our approach to the ones from ProDiGy [12], which adopts a similar setting. To conduct the experiment, we follow the top recommendation of ProDiGy in every step to get the intermediate models and compare the models' quality with ours. We use the projected log of every iteration to evaluate the model obtained after adding additional activity to the model. To have a fair comparison, we force our approach to use the same order of adding activities from ProDiGy.

Experiment 2 (Effects of Ordering Strategy & Search Space Pruning): The order of adding activities to the log is crucial to our approach as model quality is highly dependent on the order [11]. Moreover, the order can influence the execution time due to its influence on the search space pruning. Therefore, we would like to investigate the effects of the ordering strategy on the model quality and the execution time. To set up the experiment, we apply the approach to the five event logs using the two different ordering strategies while keeping the other two parameters at the same values. We evaluate the model quality in terms of fitness, precision, and F1 score. In addition, we keep track of the ratio of the reduced nodes, which is calculated by $\frac{|V^i|}{|P^i \cup T^i|}$. This gives us an indication of the effectiveness of search space pruning.

³ <https://pm4py.fit.fraunhofer.de/>.

⁴ <https://github.com/tsunghao-huang/synthesisRulesMiner>.

⁵ <https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f>.

⁶ <https://doi.org/10.4121/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb>.

⁷ <https://doi.org/10.4121/uuid:76c46b83-c930-4798-a1c9-4be94dfef741>.

⁸ <https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5>.

Experiment 3 (Effects of Non-block Structures): In this experiment, we compare our approach to the state-of-the-art: Inductive Miner - Infrequent (IMf) [15]. As the models discovered by IMf are guaranteed to be sound free-choice workflow net as well, comparing our approach with IMf enables us to see if the models can benefit from the non-block structures discovered by our approach. For each event log, we apply IMf using five different values ($[0.1, 0.2, 0.3, 0.4, 0.5]$) for the filter threshold and choose the best model (by F1 score) to compare the quality with the ones discovered by our approach in experiment 2.

For all the experiments, we use the alignment-based approaches to calculate fitness [4] and precision [5]. We also calculate the F1 score as the harmonic mean of the fitness and precision.

5.2 Results

Effectiveness of Patterns. Figure 7 shows the result of the comparison. The fitness and precision are the average values of the five event logs. As one can see from the figures, both approaches can capture the behaviors quite well for the first three activities added. When adding more activities to the model, our approach has consistently higher values for both fitness and precision than ProDiGy. One might think that this is expected as we extend the set of patterns used in ProDiGy. However, note that ProDiGy evaluates every possible synthesis rules applications while we only focus on a subset of the nodes using log heuristics. There is a trade-off between optimal solution and time in our approach. Nevertheless, the results show that the extended patterns enable us to discover models with higher quality compared to the existing approach, ProDiGy, while limiting the search space.

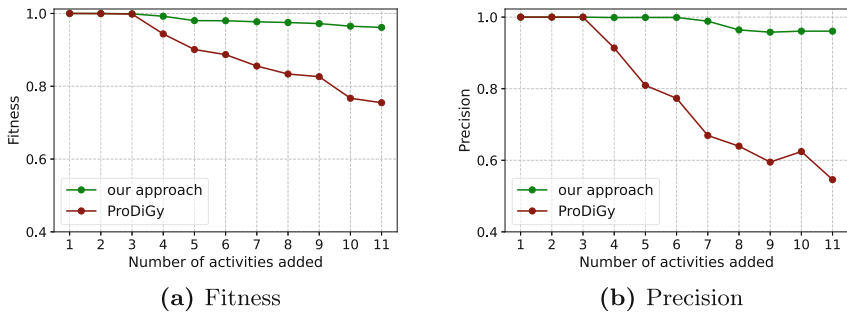


Fig. 7. Results on fitness and precision comparison for the effectiveness of patterns

Effects of Ordering Strategy and Search Space Pruning. Table 1 shows the results of experiments 2 and 3. We observe that the BFS-based ordering strategy performs better than the frequency-based strategy (in terms of F1 score and time) for four of the five logs. We further investigate the reason for the shorter

execution time of BFS-based ordering. As shown in Fig. 8, it turns out that the BFS-ordering strategy is more effective (lower $\frac{|V^i|}{|P^i \cup T^i|}$) in reducing the search space at the later stage of the discovery process. As the model grows, checking the preconditions of an application for the linear dependent place or transition rule becomes more expensive. Reducing the search space more effectively at the later stage is more beneficial in terms of execution time in most cases. BFS-based ordering achieves this by considering the closeness of activities in the process. In such case, activities that are closer together are added first and it is more likely for BFS-based ordering to focus on a smaller subset of nodes on the existing net when pruning the search space compared to the frequency-based one.

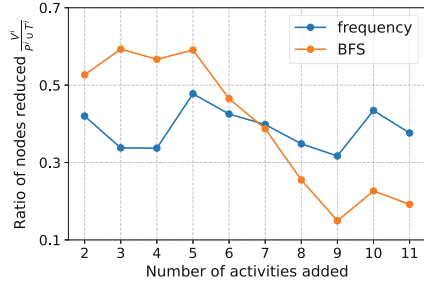


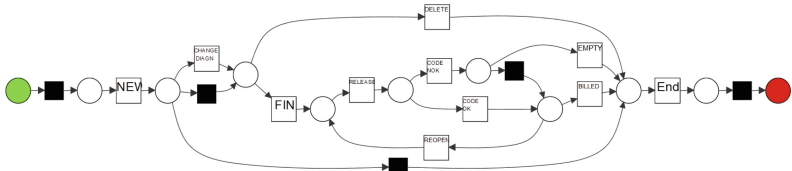
Fig. 8. Comparison of the to-be-considered nodes ratio for each iteration between the two ordering strategies.

Table 1. Results about effects of ordering strategy and comparison to IMf

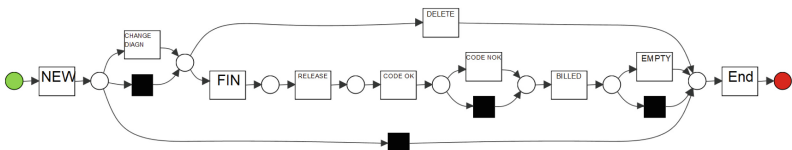
Log	Miner	Ordering Strategy	IMf filter	Fitness	Precision	F1	Time (s)
BPI2017A	Ours	Frequency	–	0.970	0.947	0.958	734
	Ours	BFS	–	0.989	0.935	0.961	342
	IMf	–	0.2	0.999	0.936	0.967	10
BPI2017O	Ours	Frequency	–	0.994	0.962	0.978	560
	Ours	BFS	–	0.989	1.000	0.994	240
	IMf	–	0.2	0.997	0.907	0.950	7
Helpdesk	Ours	Frequency	–	0.972	0.984	0.977	54
	Ours	BFS	–	0.981	0.976	0.978	44
	IMf	–	0.2	0.967	0.950	0.958	1
Hospital billing	Ours	Frequency	–	0.961	0.810	0.879	567
	Ours	BFS	–	0.989	0.935	0.961	407
	IMf	–	0.2	0.982	0.906	0.943	45
Traffic	Ours	Frequency	–	0.960	0.930	0.945	321
	Ours	BFS	–	0.964	0.720	0.825	427
	IMf	–	0.4	0.904	0.720	0.801	28

Effects of Non-block Structures. Table 1 shows that compared to IMf, the models discovered by our approach have higher F1 scores for four of the five logs. Note that the fitness values of the models discovered by our approach are all higher than the defined threshold 0.95. In general, IMf tends to discover models with higher fitness values while our approach discovers models with higher

precision. In IMf, one can use the filter threshold to balance fitness and precision. This is also the case in our approach, the user can set a lower fitness threshold to include more candidate nets that are less fitting but more precise. Figure 9 shows the discovered models from the two approaches for the hospitalBilling log. While the overall structure of Fig. 9a is similar to its counterpart in Fig. 9b, our approach discovered non-block structures at the later stage of the process. Such construct is not possible to model by IMf. The result shows that our approach can discover sound free-choice workflow nets with non-block structures and produce competitive model quality as the state-of-the-art algorithm.



(a) The discovered model using our approach. Due to the more flexible structure, one can execute *EMPTY*, *BILLED*, or *REOPEN* after *CODE NOK* while only *BILLED* or *REOPEN* are executable after *CODE OK*. The construct is not discoverable by IMf.



(b) The discovered model using IMf. Note that activity *REOPEN* is dropped by the filter of IMf.

Fig. 9. The models discovered by our approach and IMf for the hospitalBilling log.

6 Conclusion and Future Work

In this paper, we present a discovery algorithm that aims to discover sound free-choice workflow nets with non-block structures. The algorithm utilizes the synthesis rules to incrementally add activities with predefined patterns to discover models that are guaranteed to be sound and free-choice. Moreover, a certain level of replay fitness is guaranteed by a user-defined threshold.

The approach has been implemented and evaluated using various real-life event logs. The results show that the process models discovered by our approach have higher model quality (in terms of both replay fitness and precision) than the existing approach [12], which also depends on synthesis rules. Moreover, our approach produces competitive model quality compared to the state-of-the-art: Inductive Miner - infrequent. For future work, we plan to explore more advanced ordering strategies and investigate their influences on the model quality and computation time. The other direction is to further speed up the approach as the long execution time is a clear limitation. This could be done by exploiting the log-based heuristics further.

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

References

1. van der Aalst, W.M.P.: The application of Petri nets to workflow management. *J. Circuits Syst. Comput.* **8**(1), 21–66 (1998)
2. van der Aalst, W.M.P.: *Process Mining - Data Science in Action*, 2nd edn. Springer, Cham (2016)
3. van der Aalst, W.M.P.: Using free-choice nets for process mining and business process management. In: *FedCSIS 2021*, vol. 25, pp. 9–15 (2021)
4. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. *WIREs Data Min. Knowl. Discov.* **2**(2), 182–192 (2012)
5. Adriansyah, A., Munoz-Gama, J., Carmona, J., Van Dongen, B.F., van der Aalst, W.M.P.: Measuring precision of modeled behavior. *Inf. Syst. E Bus. Manag.* **13**(1), 37–67 (2015)
6. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L., Bruno, G.: Automated discovery of structured process models from event logs: the discover-and-structure approach. *Data Knowl. Eng.* **117**, 373–392 (2018)
7. Augusto, A., et al.: Automated discovery of process models from event logs: review and benchmark. *IEEE Trans. Knowl. Data Eng.* **31**(4), 686–705 (2019)
8. Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Polyvyanyy, A.: Split miner: automated discovery of accurate and simple business process models from event logs. *Knowl. Inf. Syst.* **59**(2), 251–284 (2018). <https://doi.org/10.1007/s10115-018-1214-x>
9. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: A genetic algorithm for discovering process trees. In: *CEC 2012*, pp. 1–8. IEEE (2012)
10. Desel, J., Esparza, J.: *Free Choice Petri Nets*. No. 40, Cambridge University Press, Cambridge (1995)
11. Dixit, P.M.: *Interactive process mining*. Ph.D. thesis, Technische Universiteit Eindhoven (2019)
12. Dixit, P.M., Buijs, J.C.A.M., van der Aalst, W.M.P.: Prodigy : human-in-the-loop process discovery. In: *RCIS 2018*, pp. 1–12. IEEE (2018)
13. Dixit, P.M., Verbeek, H.M.W., Buijs, J.C.A.M., van der Aalst, W.M.P.: Interactive data-driven process model construction. In: Trujillo, J.C., et al. (eds.) *ER 2018*. LNCS, vol. 11157, pp. 251–265. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_19
14. van Dongen, B.F., de Medeiros, A.K.A., Wen, L.: Process mining: overview and outlook of petri net discovery algorithms. *Trans. Petri Nets Other Model. Concurr.* **2**, 225–242 (2009)
15. 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 (2016). <https://doi.org/10.1007/s10270-016-0545-x>
16. Schuster, D., van Zelst, S.J., van der Aalst, W.M.P.: Incremental discovery of hierarchical process models. In: Dalpiaz, F., Zdravkovic, J., Loucopoulos, P. (eds.) *RCIS 2020*. LNBIP, vol. 385, pp. 417–433. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50316-1_25



Shape Your Process: Discovering Declarative Business Processes from Positive and Negative Traces Taking into Account User Preferences

Federico Chesani¹(✉), Chiara Di Francescomarino², Chiara Ghidini², Giulia Grundler¹, Daniela Loreti¹, Fabrizio Maria Maggi³, Paola Mello¹, Marco Montali³, and Sergio Tessaris³

¹ DISI - University of Bologna, Bologna, Italy
federico.chesani@unibo.it

² Fondazione Bruno Kessler, Trento, Italy

³ Free University of Bozen/Bolzano, Bolzano, Italy

Abstract. Process discovery techniques focus on learning a process model starting from a given set of logged traces. The majority of the discovery approaches, however, only consider one set of examples to learn from, i.e., the log itself. Some recent works on declarative process discovery, instead, advocated the usefulness of taking into account two different sets of traces (a.k.a. positive and negative examples), with the goal of learning a set of constraints that is able to discriminate which trace belongs to which set. Sometimes, however, too many possible sets of constraints might be available, thus nullifying the discovery effort. Therefore, some preference criteria would be helpful to guide the discovery process towards a set of constraints among the many. In this work, we present an approach for the discovery of declarative models providing the possibility, from the user viewpoint, of specifying preferences on activities and constraint templates to be used to build the final set of constraints. Such preferences are used to guide the discovery process, so that the output set will include, if possible, the preferred constraints, thus exploiting some expert knowledge about the desired outcome. The approach is grounded in a logic-based framework that provides a sound and formal meaning to the notion of expert preferences.

1 Introduction

Process discovery is one of the most investigated process mining techniques [43]. It deals with the automatic learning of a process model from a given set of logged traces, each one representing the digital footprint of the execution of a case.

If we focus on the way process discovery techniques see the model-extraction task, we can divide them into two broad categories. The first category is constituted by works that tackle the problem of process discovery with one-class supervised learning techniques (see, e.g., [1, 3, 4, 23, 44]). These works are driven by the assumption that all available log traces are instances of the process to be

discovered and constitute the vast majority of works in the process discovery spectrum. The second category comprises works that intend model-extraction as a two-class supervised task, which is driven by the possibility of partitioning the log traces into two sets according to some business or domain-related criteria. Usually, these sets are referred to as *positive* and *negative* examples, and the goal is to learn a model that characterizes one set w.r.t. the other. These works are traditionally less represented (see [16, 21, 31]). Nonetheless, few recent works [15, 26, 39] have highlighted the importance of performing model-extraction as a two-class supervised task with different motivations: first, the actual existence of *positive* and *negative* examples in real use cases [26, 39]; second, the need to balance *accuracy* and *recall* [39]; and third, the need to discover a particular process variant (e.g., the process characterizing “fast” traces) against the one that characterizes other variants, thus using the labels *positive* and *negative* to distinguish between two classes of examples [15]. Hereafter, we refer to miners of the first and second category as *unary* and *binary* miners, respectively.

A problem that remains unsolved in process discovery, in general, and in binary miners, in particular, is the need to select, among all possible discovered models, the ones that fit better the expectations of expert users, that is, users who are knowledgeable about the specific domain and because of this have specific desiderata and expectations. This is true for the traditional discovery of procedural and declarative models, where the discovered model that accepts all the positive examples is usually too complex (e.g., too spaghetti like), and mechanisms are introduced to “select” specific behaviors. Examples of criteria for this selection can be the frequency of a certain element (e.g., an activity or a path), or the presence of certain modeling patterns (e.g., a specific declarative pattern). The problem becomes even more compelling when we approach process discovery as a two-class supervised task. In fact, as recently shown in [39], perfect binary miners, able to discover models that accept all positive examples and none of the negative examples, do not necessarily exist. In such cases, many sub-optimal models can be returned, leading to the issue of identifying criteria for preferring one model or the other.

In this paper, we address the problem of inserting expert user preferences, (hereafter expert preferences) in the discovery of declarative process models as a two-class supervised task. We start from a recent work [15] that introduces the `NegDis` binary miner for the `Declare` modeling language [36] (introduced in Sect. 2). Being `NegDis` based on the logic-based framework `ASP` [12], it provides a formal framework with a clear semantics that allows the users to “prioritize” the discovery results. In this work, we extend `NegDis` by introducing the `ASPrin` tool [11], so as to support the notion of expert preferences while remaining within the context of a formal, logic-based semantics. The following contributions are provided:

- (i) we introduce and motivate two types of expert preferences: the first one on the `Declare` patterns to be used in the discovery task, and the second one on the activities appearing in the output model. Moreover, we discuss also a third type of preference coming from the combination of the first two (Sect. 3).

- (ii) we extend the original mechanism of **NegDis** (Sect. 4), by incorporating the *ASPrin* tool [11] into it. This allows us to integrate within a single framework both the expert preferences, as well as the original **NegDis** mechanism based on *model subsumption* (that is treated as a preference as well). In this way, we retain the original ability of obtaining models that vary in generality/specificity, or simplicity.
- (iii) we provide some hints about the implementation (Sect. 5);
- (iv) we report on exploratory experiments applying an instantiation of **NegDis** to the data sets used in [24, 39] (Sect. 6).

Related works (Sect. 7) and final considerations (Sect. 8) conclude this work.

2 The Modeling Language

The discovery approach we introduce in this paper is based on **Declare**, a language for describing declarative process models first introduced in [36]. A **Declare** model consists of a set of constraints applied to (atomic) activities. Constraints are, in turn, based on templates. Templates are abstract parameterized patterns and constraints are their concrete instantiations on real activities. Templates have a graphical representation and their semantics can be formalized using different logics, the main one being LTL for finite traces, making them verifiable and executable. Each constraint inherits the graphical representation and semantics from its template. The major benefit of using templates is that analysts do not have to be aware of the underlying logic-based formalization to understand the models. They work with abstract representations of templates, while the underlying formulas remain hidden. Table 1 summarizes the main **Declare** constructs used in this paper. The reader can refer to [36] for a full description of the language.

Table 1. Declare templates

Template	Explanation
<code>existence(n, A)</code>	A occurs at least n times
<code>absence(m + 1, A)</code>	A occurs at most m times
<code>responded_existence(A, B)</code>	If A occurs, then B occurs
<code>response(A, B)</code>	If A occurs, then B occurs after A
<code>alternate_response(A, B)</code>	Each time A occurs, then B occurs afterwards, before A recurs
<code>precedence(A, B)</code>	B occurs only if preceded by A
<code>chain_precedence(A, B)</code>	Each time B occurs, then A occurs immediately before
<code>co_existence(A, B)</code>	If B occurs, then A occurs, and vice versa
<code>not_succession(A, B)</code>	A never occurs before B
<code>not_chain_succession(A, B)</code>	A and B occur if and only if the latter does not immediately follow the former

3 Why Preferences on the Discovered Models?

Users look for discovering models for a variety of reasons. A common one is related to the need of having a description/explanation of a process. Other reasons might be, for example, the need for detecting process deviations or process drifts. Or, as in the case of `NegDis`, expert users might be interested in understanding, from a model viewpoint, what distinguishes one set of traces from another.

Depending on the discovery technique and the target language, many alternative models might describe the same process. For example, both `BPMN` and `Declare` allow us to describe the same process using different constructs or templates. However, not all the discovered models are equivalent¹, and even when they are equivalent, there could be too many models to choose from.

Since the availability of many models might, in turn, hinder the usefulness of the discovery approach, the expert user would need a criterion for selecting few models among the many discovered. Preferences on the discovered models represent then a way for prioritizing the discovered models based on the expert user's needs. In particular, we envisage three different types of preferences: preferences over activities, preferences over templates, and a combination of both.

3.1 Preferences over Process Activities

A first type of preferences on the discovered models is strictly related to the application domain. Indeed, depending on the expert user's goals, models that focus more on certain activities might be preferable.

Example 1. Let us consider a “loan scenario”, where a bank receives a request for a loan, evaluates it, and provides an answer. Let us assume that process instances have been classified into two sets, for example including successful and unsuccessful applications. The bank employee will look then for a model that helps her to understand the differences. Of course, the employee will not directly look into the logs, which, for simplicity, we can suppose to be as follows:

$$L^+ = \{ \langle \text{loanRequest}, \text{requestEval}, \text{notifyOutcome} \rangle \}$$

$$L^- = \{ \langle \text{requestEval}, \text{loanRequest} \rangle \}$$

where the positive example set L^+ contains only one trace (composed of three activities), and the negative example set L^- contains a single trace as well.

If the employee is an employee working in the marketing department, she could have in mind the bank slogan “we always answer our customers”. Hence, she would be surely interested in the `notifyOutcome` activity. By specifying such

¹ Roughly speaking, two models are *equivalent* if they accept and reject the same traces. Such a notion of equivalence hints to the possibility that given two models M_1 and M_2 , opting for the former or the latter will not change which traces will be accepted or rejected.

preference, the discovery algorithm would return two models both involving the preferred activity²:

$$M_1 = \{\text{response}(\text{requestEval}, \text{notifyOutcome})\}$$

$$M_2 = \{\text{existence}(\text{notifyOutcome})\}$$

□

Generally speaking, being able to specify a preference for models that refer to specific activities allows expert users to answer the question “*Is it possible to discriminate between two sets of traces by looking at certain activities?*”. The discovery process becomes, in this way, domain-driven: many models describe the process, but those ones that focus on certain domain aspects should be returned before others.

3.2 Preferences over Declare Templates

Process description languages like, e.g., BPMN and **Declare**, are quite rich in their expressiveness, and allow us to describe a process using different constructs or templates. This leads to the availability of alternative models that could be equivalent or not. Unfortunately, even when restricting our attention to equivalent models only, it is easy to see that they might not convey the information in exactly the same way to users.

Case 1: Equivalent models. Let us consider first the case where a discovery algorithm provides as output two equivalent models. If from a “conformance viewpoint” nothing changes, from a high-level viewpoint different models might bear subtle meaning distinctions, as shown in the following example.

Example 2. Let us assume to have the following log, whose traces have been classified into two sets:

$$L^+ = \{ \langle a, b \rangle, \langle b, a \rangle \} \quad L^- = \{ \langle a \rangle, \langle b \rangle \}$$

Alternative models allowing us to represent the traces that belong to L^+ and exclude the ones that belong to L^- are:

$$M_1 = \{\text{existence}(a), \text{existence}(b)\}$$

$$M_2 = \{\text{existence}(a), \text{responded_existence}(a, b)\}$$

$$M_3 = \{\text{existence}(b), \text{responded_existence}(b, a)\}$$

$$M_4 = \{\text{existence}(a), \text{co_existence}(a, b)\}$$

$$M_5 = \{\text{existence}(b), \text{co_existence}(a, b)\}$$

² Other models exist, of course, but, for the sake of clarity, we only mention two of them.

From a logical viewpoint, models M_1 – M_5 are equivalent. However, models M_2 – M_5 suggest that what distinguishes the traces in L^+ from the traces in L^- is a relation between activities **a** and **b**: indeed, these models contain a binary constraint, whose purpose is to highlight a relation between these two activities. Model M_1 , instead, does not tell us anything about possible links between activities **a** and **b**, and a user might conclude that no relation exists between them. \square

Declare binary templates, by their nature, suggest a link between activities. Hence, a discovery algorithm that would return models with relation constraints would emphasize such links. The user would be left with the burden of understanding if such links are mere coincidences or artifacts of the discovery technique, or if rather some new knowledge has been discovered about the process.

We can imagine scenarios where expert users prefer models containing the minimum number of binary templates, so as not to incur into the risk of perceiving in-existent relations. On the other hand, we can easily think about situations where an expert user is actively looking for relations. In both cases, preferences on which **Declare** templates should be preferably included into a model would allow the expert user to tailor the discovery process to her needs.

Notice also that Example 2 might mislead the reader to think that preferences over templates is a matter of unary vs. binary constraints only. This is not the case, since equivalence is a logic property that stems from the interplay between all the constraints within each single model. Models with many binary constraints might be proved to be equivalent, as shown in the following example.

Example 3. Let us consider the following log:

$$L^+ = \{ \langle \mathbf{a}, \mathbf{b} \rangle, \langle \mathbf{a}, \mathbf{b}, \mathbf{c} \rangle, \langle \mathbf{a}, \mathbf{c}, \mathbf{b} \rangle \} \quad L^- = \{ \langle \mathbf{a} \rangle, \langle \mathbf{a}, \mathbf{c} \rangle \}$$

Two alternative models that accept the positive traces and reject the negative ones are:

$$M_1 = \{ \text{absence2}(\mathbf{a}), \text{response}(\mathbf{a}, \mathbf{b}) \} \quad M_2 = \{ \text{absence2}(\mathbf{a}), \text{alternate_response}(\mathbf{a}, \mathbf{b}) \}$$

Models M_1 and M_2 are equivalent due to the interplay of the constraint **absence2** with the the **response** and the **alternate_response** constraints: roughly speaking, being activity **a** forbidden to appear more than once, the effects of the stricter constraint **alternate_reponse** are nullified. \square

Case 2: Non-equivalent models. Let us consider now the case where alternative non-equivalent models are discovered. This might happen because a log is usually a partial view of all the possible execution traces. Not-yet-seen traces are *unknown* w.r.t. the classification, but different models could classify them in different manners. Different models would *shape the unknown* differently.

Example 4. Let us consider the following log:

$$L^+ = \{ \langle \mathbf{a}, \mathbf{b} \rangle, \langle \mathbf{b}, \mathbf{a} \rangle \} \quad L^- = \{ \langle \mathbf{a} \rangle \}$$

Alternative models that accept the traces in L^+ and discard the ones in L^- are:

$$M_1 = \{\text{existence(a), existence(b)}\} \quad M_2 = \{\text{responded_existence(a, b)}\}$$

Let us consider then the trace $\langle b \rangle$, that was not recorded in the log. Model M_1 would reject it, whereas model M_2 would accept it. \square

Example 4 shows how traces not appearing in the log used for the discovery might be classified differently by the discovered models. A preference elicitation mechanism would allow the expert user to decide how the not-yet-seen traces would be classified, in a restricting or in a broader way. Another example is given below.

Example 5. Let us consider the following log:

$$L^+ = \{ \langle a, b \rangle, \langle a, b, c \rangle, \langle a, c, b \rangle \} \quad L^- = \{ \langle a \rangle, \langle a, c \rangle \}$$

Two non-equivalent models that accept the positive examples and reject the negative ones are:

$$M_1 = \{\text{response(a, b)}\} \quad M_2 = \{\text{alternate_response(a, b)}\}$$

\square

Both models in Example 5 suffice to classify a trace into one or the other class. However, model M_2 is *stricter*, since it accepts less traces and rejects more traces than M_1 . A expert user might express her preference for stricter or more general models.

3.3 Preferences over both Activities and Templates

The third type of preferences on the discovered models is a straightforward combination of the preference types introduced in Subsects. 3.1 and 3.2. Domain-related knowledge would drive the attention to certain activities, and preferences over templates would allow focusing on certain relation types.

Example 6. Let us consider again the “loan scenario” and the log:

$$L^+ = \{ \langle \text{loanRequest, requestEval, notifyOutcome} \rangle \}$$

$$L^- = \{ \langle \text{requestEval, loanRequest} \rangle \}$$

Let us consider now the viewpoint of an employee working in the internal auditing department. Given that the wrong execution order of certain activities might be a symptom of some fraud, the employee would like to focus the attention over templates of type **response** and/or **precedence**, and, in particular, over those constraints involving the **requestEval** activity. The discovery algorithm would exploit such preference by looking for models with the elicited features, and would provide in output:

$$M = \{\text{precedence(requestEval, loanRequest)}\}$$

\square

Notice that Example 6 shares the exact same log as Example 1. However, the output is completely different: the preferences are used, indeed, to guide the search for a model, which is of interest for the expert user.

4 Discovering Business Processes from Positive and Negative Traces

Our approach is based on the `NegDis` binary miner [15], which, given two input sets of positive and negative examples, aims at extracting a model accepting all positive traces and rejecting all negative ones. In this work, we enrich `NegDis` with the possibility to express domain-dependent preferences on the discovered models. Therefore, we report some definitions and explanations from [15] that are useful to understand our approach.

`NegDis` starts from a certain *language bias*: given a set of `Declare` templates D and a set of activities A , we indicate with $D[A]$ the set of all possible groundings of templates in D w.r.t. A , i.e., all the constraints that can be built using activities in A .

We respectively denote with L^+ and L^- the sets of positive and negative examples in the input event log. `NegDis` starts by considering a, possibly empty, initial model P , that is a set of `Declare` constraints known to characterize the examples in L^+ . The goal of `NegDis` is to refine P taking into account both the positive and the negative examples.

Definition 1. *Given the initial model P , a candidate solution for the discovery task is any set of constraints $S \subseteq D[A]$ s.t. (i) $P \subseteq S$; (ii) $\forall t \in L^+$ we have $t \models S$; (iii) S maximizes the set $\{t \in L^- \mid t \not\models S\}$.*

`Declare` templates can be organized into a hierarchy of *subsumption* [19] according to the logical implications derivable from their semantics. Consistently with this concept, we introduce the following definition of *generality* relation between models.

Definition 2. *A model $M \subseteq D[A]$ is more general than $M' \subseteq D[A]$ (written as $M \succeq M'$) when for any $t \in A^*$, $t \models M' \Rightarrow t \models M$, and strictly more general (written as $M \succ M'$) if M is more general than M' and there exists $t' \in A^*$ s.t. $t' \not\models M'$ and $t' \models M$.*

`NegDis` integrates the *subsumption* rules introduced in [19], into the *deductive closure operator*.

Definition 3. *Given a set R of subsumption rules, a deductive closure operator is a function $cl_R : \mathcal{P}(D[A]) \rightarrow \mathcal{P}(D[A])$ that associates any set $M \in D[A]$ with all the constraints that can be logically derived from M by applying one or more deduction rules in R .*

For brevity, in the rest of the paper, we will omit the set R and we will simply write $cl(M)$ to indicate the deductive closure of M . The complete set of employed deduction rules is available in the source code [42].³

Conceptually, the `NegDis` approach can be seen as a two-step procedure: in the first step, a set of candidate constraints is built, and then solutions are selected among subsets of candidates via an optimization algorithm. The set of candidate constraints is composed of those in $D[A]$ that accept all positive examples and reject at least a negative one. To build this set, `NegDis` constructs a *compatibles* set, i.e., the set of constraints that accept all traces in L^+ :

$$compatibles(D[A], L^+) = \{c \in D[A] \mid \forall t \in L^+, t \models c\} \quad (1)$$

Then, it defines the *sheriffs* function to associate to any trace t in L^- the constraints of *compatibles* that reject t :

$$sheriffs(t) = \{c \in compatibles \mid t \not\models c\} \quad (2)$$

The *sheriffs* function is used to construct the set of all candidate constraints from which a discovered model is derived, i.e., the set $\mathcal{C} = \bigcup_{t \in L^-} sheriffs(t)$ of all the constraints in $D[A]$ accepting all positive traces and rejecting at least one negative trace. The solution space is therefore:

$$\mathcal{Z} = \{M \in \mathcal{P}(\mathcal{C}) \mid \forall t \in L^- t \not\models M \cup P \text{ or } sheriffs(t) = \emptyset\} \quad (3)$$

Due to the fact that not all the pairs of negative and positive sets of traces can be perfectly separated using `Declare` [39], there can be traces in L^- for which the *sheriffs* is empty, meaning that those traces cannot be excluded by any model that guarantees the acceptance of all the positive ones.

The second step of `NegDis` uses an optimization strategy to identify the solutions; in [15], two different criteria were taken into account: *generality* (or conversely, *specificity*), and *simplicity*. If the user is interested in the most general model, then `NegDis` employs the closure operator cl to select the models $S \in \mathcal{Z}$ with the less restrictive behavior. If the user wants the simplest model, `NegDis` looks for the solutions with minimal closure size. In case of ties, the solution with the minimal size is preferred.

5 Adding Preferences to Process Discovery: An Implementation Through *ASPrin*

5.1 Specifying the Preferences

As discussed in Sect. 3, in this work we support three different types of preferences. Preferences over domain activities are simply expressed through ASP Prolog facts of the type:

```
good_action(X).
```

³ The file `declare_rules.txt` can be found in the `data` directory.

where X is a placeholder for an activity name. The intended meaning is that models containing `Declare` constraints about the the action X should be preferred to models that do not contain it.

Analogously, to specify a preference for a `Declare` template, we simply add a sentence of the type:

```
good_constraint(X).
```

where X now is a placeholder for a template name. Again, the intended meaning is about preferring models containing the specified templates, to other models.

Finally, to express a preference that is a combination of the previous types, we allow the user to write facts like:

```
good_constraint_action(decl(Template, Action1 [, Action2])).
```

where `Template` is a placeholder for the `Declare` template name, `Action1` is the placeholder for the activity name and, in case the preferred template is a binary one, `Action2` is the placeholder for the second activity. It is worthy to mention that it is possible to express several preferences at the same time: the intended meaning is that models satisfying more preferences are preferable to models that satisfy less preferences.

5.2 Exploiting *ASPrin* for Searching Preferred Models

At a first glance, one could think that the *sheriffs* function in Eq. 2 includes all that we need to generate “preferred” models. Indeed, a naive idea would be to select exactly one constraint from each $sheriffs(t) \neq \emptyset$ for $t \in L^-$ according to some preferences. However, this solution does not take into account the interplay among constraints. In particular, some constraints might be more general than others, or even there might be cases in which two constraints imply the validity of a third one. This would clearly interfere with the validity of the specified preferences.

For this reason, we cannot use any combinatorial optimizer to enforce the preferences, but we need a system enabling some form of constraint propagation. In [15], we use Answer Set Programming (ASP) by leveraging the underlying rule based formalism enabling propagation, and *weak constraints* for optimization [12, 20]. The encoding of the optimization problem follows the *Generate and Test* ASP paradigm where part of the rules select a candidate ASP model (e.g., a subset of \mathcal{C}) and a set of constraints filters only the relevant models (e.g., those “rejecting” all the negative examples). Weak constraints are used to assign a preference value to any ASP model, i.e., a violated weak constraint does not reject the model but assigns a penalty to it. In [15], simple weak constraints were used to implement subsumption preferences; however, specifying more complex preferences between ASP models (like the ones presented so far) using weak constraints would become unmanageable and error-prone.

To tackle this issue, in this work, we exploit the *ASPrin* tool [11], which layers upon the *clingo* ASP solver [20], enabling the specification of complex preference relations through user-defined types and their arguments. *ASPrin* provides

a general framework for optimizing qualitative and quantitative preferences in ASP. While *ASPrin* comes with a library of predefined preference types (subset, pareto, lexicographic, etc.), it is readily extensible by new customized preference types. Preferences can be defined and aggregated by means of higher level types, making *ASPrin* the perfect tool to support the preferences introduced in Sect. 3. Moreover, *ASPrin* provides a simple way to implement the “generality” and “simplicity” criteria discussed in Sect. 4.

Describing the *ASPrin* language and the precise encoding of the optimization problem is outside the scope of this paper (the full code is available in [42] while a detailed description of how we encoded the process discovery problem using the *ASPrin* framework is available in [14]). However, in abstract wording, we use two different predicates, which are explicitly represented by means of their template and activities as predicate arguments. This enables the characterization of the ASP models, e.g., by prioritizing those in which specific templates and/or activities are selected or excluded. These preferences can be combined with domain-independent preferences, e.g., on the size of the discovered models to provide a fine-grained ordering among them.

6 Evaluating the Discovery

In Sect. 3, we introduced the preference types through simple toy-like examples. The interested reader, however, might wonder about the usability and efficacy of our approach when applied to real-life cases. We explored the applicability of our approach using two real-life event logs, namely DREYERS (492 positive traces and 208 negative ones) and CERV (55 positive traces and 102 negatives traces).⁴ In both cases, we were able to find ten models satisfying the given preferences in a computation time between 1 and 3s, using a normally-equipped laptop.

6.1 The DREYERS Log

The DREYERS log describes the Dreyer Foundation’s processes pertaining to their support to legal and architectural projects, and it has been used in [17, 39]. Each application to request the Foundation’s support goes through a pre-screen that can lead to an initial rejection. The remaining applications undergo a review, in which at least one of the reviewers must be a lawyer or an architect, depending on the application type. The review phase is followed by a board meeting, where applications to be supported are selected and eventually funded. Two sets of log traces are available in the dataset: a positive one collecting executions that did not fail and a negative one representing executions that were reset due to a system failure.

Using this dataset, we played a sort of “investigation game”, and explored the hypothesis that the type of application (architect- or lawyer-

⁴ For reproducibility purposes the source code is available in [42], the DREYERS event log can be found in [17, 39], while the CERV event log is a proprietary dataset.

Table 2. Traces ruled out by each constraint of model M_1 .

Constraints	Traces #	Variants #
alternateresponse(Undo payment, First payout)	2	2
chainprecedence(Fill out application, Initial Rejection)	3	2
choice(Round ends, Change phase to Abort)	195	17
notchainsuccession(Receive final report, First payout)	1	1
notchainsuccession(Change phase to Preparation, Approve application)	1	1
notchainsuccession(Change phase to Preparation, Execute Pre decision)	2	2
notchainsuccession(Set to Pre approved, Round Ends)	2	2
notsuccession(Architect Review, Approval on to the board)	1	1
Traces not ruled out by the model	3	2
Total	208	30

type) might affect the process outcome. To this end, we initially specified a preference `good_action(Lawyer Review)`, and later on a preference `good_action(Architect Review)`. In both cases, more than one model satisfying the preferences were found. However, the two sets of models are identical (except for the architect/lawyer activity), showing that the process is independent of the application domain. We report an example of a model obtained when specifying the preference for models containing activity `Architect Review`:

$$M_1 = \{ \text{alternateresponse(Undo payment, First payout)}$$

$$\text{chainprecedence(Fill out application, Initial Rejection)}$$

$$\text{choice(Round ends, Change phase to Abort)}$$

$$\text{notchainsuccession(Receive final report, First payout)}$$

$$\text{notchainsuccession(Change phase to Preparation, Approve application)}$$

$$\text{notchainsuccession(Change phase to Preparation, Execute Pre decision)}$$

$$\text{notchainsuccession(Set to Pre approved, Round Ends)}$$

$$\text{notsuccession(Architect Review, Approval on to the board)} \}$$

Notably, as shown in Table 1, this model is able to discriminate between positive and negative examples except for three negative traces (two variants), that cannot be ruled out without discarding also some positive examples.

We continued our investigation by focusing on activity `Initial Rejection`. We report here one of the returned models:

$$M_2 = \{ \text{absence2(Initial rejection)}$$

$$\text{choice(Round Ends, Applicant informed)}$$

$$\text{notchainsuccession(Set to Pre approved, Round Ends)}$$

$$\text{notchainsuccession(Receive final report, First payout)}$$

$$\text{notchainsuccession(Change phase to Preparation, Approve application)}$$

$$\text{notchainsuccession(Change phase to Preparation, Execute Pre decision)}$$

$$\text{notsuccession(Lawyer Review, Change phase to review)}$$

$$\text{response(Undo payment, First payout)} \}$$

Model M_2 highlights the fact that some negative traces can be distinguishable from the positive ones because of the repetition of **Initial Rejection**: some traces, indeed, reported the execution of the activity twice, thus indicating an attention point for the process analyst.

Finally, we did compare the effect of discovering models with or without the two preferred activities. For this we asked **NegDis** to extract 10 optimal models with no preferences, 10 with the **Architect Review** preference and 10 with the **Initial Rejection** preference, and we pairwise compared the models with no preference and the ones with a preferred activity. When imposing no preferences, activity **Architect Review** shows up in only 4 of the 10 discovered models. Imposing the usage of **Initial Rejection** is instead “unnecessary” (a posteriori), as this activity is also present in all 10 models discovered without specifying any preference.

6.2 Evaluation on the CERV Log

CERV is an event log that describes a process pertaining to the cervical cancer screening in an Italian screening center, and it has been used in previous works [15, 24]. The screening program is composed of five phases, organized sequentially: screening planning, invitation management, first level test with pap-test, second level test with colposcopy (only if the first test is positive), and eventually biopsy (if the second test gives a positive response). Several subjects do not respect the planned protocol: e.g., subjects might not show up at the first test, even if they have chosen a time slot. Moreover, a number of subjects prefer to consult physicians they trust more, in case of a positive response. As it commonly happens in socio-technical systems, a large variety of process instances appear in the log, not all them being compliant with the protocol. Hence, the traces have been labeled by a domain expert as belonging either to the positive or the negative set, depending on their compliance with the adopted protocol.

We investigated the log by eliciting two preferences over the precedence and succession templates:

```
good_constraint(precedence) .
good_constraint(succession) .
```

The first two returned models are:

$$M_1 = \{ \text{alternateresponse}(\text{send positive pap test result, take a colposcopy examination}) \\ \text{chainprecedence}(\text{invite, take a pap test examination}) \\ \text{exclusivechoice}(\text{send pap test sample, reject}) \\ \text{precedence}(\text{send colposcopy uncertain result, send biopsy sample}) \}$$

$$M_2 = \{ \text{alternateresponse}(\text{send positive pap test result, take a colposcopy examination}) \\ \text{chainprecedence}(\text{invite, take a pap test examination}) \\ \text{exclusivechoice}(\text{send pap test sample, reject}) \\ \text{succession}(\text{send colposcopy uncertain result, send biopsy sample}) \}$$

In model M_1 , the precedence constraint implies that if a biopsy is executed, then the colposcopy examination has provided an uncertain result before. The second model is identical to the first one, except for the constraint related to our preference. Interestingly, the succession relates the same activities involved in the precedence constraint in the first model. The difference between the two models lies in the logical relation between precedence and succession: a trace that violates the former will always violate the latter (but not vice versa). It is then up to the domain expert to prefer a stricter or a more general behavior.

Finally, we did compare the effect of discovering models with or without the two preferred templates, by extracting 10 optimal models with no template preferences, and 10 each with the Precedence and Succession preference respectively. Interestingly enough, imposing the Precedence preference results in being extremely useful in this scenario. In fact, none of the 10 models discovered with no preferred template did contain a Precedence pattern. Similarly with Succession, which appears in only 1 of the 10 models discovered without specifying any preference.

7 Related Work

When processes are loosely-structured, procedural discovery could produce spaghetti-like models [15, 28]. In that case, declarative approaches are more suitable for the purpose since they briefly list all the required or prohibited behaviors without explicitly specifying all possible process paths.

Over the last decade, several works focused on declarative process discovery [18, 19, 30, 38]. In [18, 30], the authors propose to build the set of all possible candidate **Declare** constraints considering all the activities that appear in the log, and check them against the whole log until certain levels of recall and specificity are reached. Techniques to refine the business model excluding vacuously satisfied constraints are the focus of the subsequent works by Schunselaar et al. [38], whereas Di Ciccio et al. [19] propose an approach to filter out frequent redundancies and inconsistencies. All the cited declarative approaches do not deal with negative examples. Nonetheless, interestingly from our point of view, in [29], the authors present an approach to specify “crisp” preferences that filter out constraints (discovered from positive examples only) that are not in line with some user knowledge. Differently from this approach, our approach allows the user to use preferences to “prioritize” the discovered models without necessarily filter out the ones that do not agree with the specified preferences.

Negative examples are instead actively employed in the declarative discovery approaches [7, 8, 15, 16, 24, 25, 39]. The technique by Lamma et al. [24, 25] learns integrity constraints expressed as logical formulas, and translates them into the equivalent DecSerFlow constructs [2]. Bellodi et al. [7, 8] employ the same approach and automatically convert the results into Markov Logic formulas—statistical relational learning is used to determine the weight of each formula. Analogously, Chesani et al. [16] propose to learn a set of SCIFF rules [5] and

translate them into ConDec constraints [35]. The approach that we adopt in this work instead, is the one presented in [15], which directly learns **Declare** constraints without any intermediate language. This approach is grounded on a SAT-based solver analogously to the works in [13, 32, 37], where simple Linear Temporal Logic (LTL) formulas are generated to analyze sets of positive and negative examples. Particularly relevant for our work is the contribution by Slaats et al. [39], which proposes a binary classification procedure for process discovery evaluated on a set of real-life logs with negative examples from industry.

Our notion of negative example is similar to the definitions of syntactical and semantic noise of [22] since our approach is able to extract both the syntactic information that characterizes the positive examples w.r.t. negative ones, and the relevant semantic difference between traces that have been partially or totally modified at a certain point in time. In this sense, our work is also closely related to deviance mining approaches [33], i.e., techniques to extract the relevant details characterizing those traces that deviate from the expected behavior. Whereas some deviance mining approaches [6, 40] focus on the differences between models discovered from deviant and non-deviant traces, others [9, 10, 27, 34, 41] intend deviance mining as a sort of sequence classification for the discovery of activity patterns discriminating between different sets of traces.

8 Conclusions

In this paper, we address the problem of inserting expert preferences in the discovery of declarative process models as a two-class supervised task. In particular, we extend the **NegDis** binary miner for the **Declare** modeling language with preferences over **Declare** templates and activities appearing in the model (plus a combination of the two). The computation of the preferred models, which take into account the preferences posed by the expert user, is performed using *ASP_{Prin}*, a general framework for computing optimal ASP models with preferences. The provided approach is described by means of motivating examples and an application to the real-life event logs DREYERS and CERV shows how to describe - in a discriminative manner - execution traces on the basis of preferred activities and **Declare** patterns. Future works will include a wider evaluation, which will also involve end-users. This will enable the assessment of the potential benefits of involving users (through their preferences) in the loop of process discovery.

Acknowledgments.. This work has been partially supported by the European Union’s H2020 projects HumaneAI-Net (g.a. 952026), StairwAI (g.a. 101017142), and TAILOR (g.a. 952215).

References

1. van der Aalst, W.M.P., De Masellis, R., Di Francescomarino, C., Ghidini, C.: Learning hybrid process models from events. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNCS, vol. 10445, pp. 59–76. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_4
2. van der Aalst, W.M.P., Pesic, M.: DecSerFlow: towards a truly declarative service flow language. In: Bravetti, M., Núñez, M., Zavattaro, G. (eds.) WS-FM 2006. LNCS, vol. 4184, pp. 1–23. Springer, Heidelberg (2006). https://doi.org/10.1007/11841197_1
3. van der Aalst, W.M.P., Rubin, V.A., Verbeek, H.M.W., van Dongen, B.F., Kindler, E., Günther, C.W.: Process mining: a two-step approach to balance between underfitting and overfitting. *Softw. Syst. Model.* **9**(1), 87–111 (2010)
4. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **16**(9), 1128–1142 (2004)
5. Alberti, M., Chesani, F., Gavaneli, M., Lamma, E., Mello, P., Torroni, P.: Verifiable agent interaction in abductive logic programming: the SCIFF framework. *ACM Trans. Comput. Log.* **9**(4), 29:1–29:43 (2008)
6. 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
7. Bellodi, E., Riguzzi, F., Lamma, E.: Probabilistic logic-based process mining. In: CILC. CEUR Workshop Proceedings, vol. 598. CEUR-WS.org (2010)
8. Bellodi, E., Riguzzi, F., Lamma, E.: Statistical relational learning for workflow mining. *Intell. Data Anal.* **20**(3), 515–541 (2016)
9. Bergami, G., Di Francescomarino, C., Ghidini, C., Maggi, F.M., Puura, J.: Exploring business process deviance with sequential and declarative patterns. *CoRR* abs/2111.12454 (2021)
10. Bose, R.P.J.C., van der Aalst, W.M.P.: Discovering signature patterns from event logs. In: CIDM, pp. 111–118. IEEE (2013)
11. Brewka, G., Delgrande, J.P., Romero, J., Schaub, T.: asprin: customizing answer set preferences without a headache. In: AAI, pp. 1467–1474. AAI Press (2015)
12. Brewka, G., Eiter, T., Truszczyński, M.: Answer set programming at a glance. *Commun. ACM* **54**(12), 92–103 (2011)
13. Camacho, A., McIlraith, S.A.: Learning interpretable models expressed in linear temporal logic. In: ICAPS, pp. 621–630. AAI Press (2019)
14. Chesani, F., et al.: Optimising business process discovery using answer set programming. In: Proceedings of the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022) (2022 To appear)
15. Chesani, F., et al.: Process discovery on deviant traces and other stranger things. *IEEE Trans. Knowl. Data Eng.* (2021), under review
16. Chesani, F., Lamma, E., Mello, P., Montali, M., Riguzzi, F., Storari, S.: Exploiting inductive logic programming techniques for declarative process mining. *Trans. Petri Nets Other Model. Concurr.* **2**, 278–295 (2009)
17. Debois, S., Slaats, T.: The analysis of a real life declarative process. In: IEEE Symposium Series on Computational Intelligence, SSCI 2015, Cape Town, South Africa, 7–10 December 2015, pp. 1374–1382. IEEE (2015)



18. Di Ciccio, C., Maggi, F.M., Mendling, J.: Efficient discovery of target-branched declare constraints. *Inf. Syst.* **56**, 258–283 (2016)
19. Di Ciccio, C., Maggi, F.M., Montali, M., Mendling, J.: Resolving inconsistencies and redundancies in declarative process models. *Inf. Syst.* **64**, 425–446 (2017)
20. Gebser, M., Kaminski, R., Kauffman, B., Schaub, T.: Multi-shot asp solving with clingo. *Theor. Pract. Logic Progr.* **19**(1), 27–82 (2019)
21. Goedertier, S., Martens, D., Vanthienen, J., Baesens, B.: Robust process discovery with artificial negative events. *J. Mach. Learn. Res.* **10**, 1305–1340 (2009)
22. Günther, C.W.: Process Mining in flexible environments. Ph.D. thesis, Technische Universiteit Eindhoven (2009)
23. 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
24. Lamma, E., Mello, P., Montali, M., Riguzzi, F., Storari, S.: Inducing declarative logic-based models from labeled traces. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 344–359. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_25
25. Lamma, E., Mello, P., Riguzzi, F., Storari, S.: Applying inductive logic programming to process mining. In: Blockeel, H., Ramon, J., Shavlik, J., Tadepalli, P. (eds.) *ILP 2007*. LNCS (LNAI), vol. 4894, pp. 132–146. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78469-2_16
26. de León, H.P., Nardelli, L., Carmona, J., vanden Broucke, S.K.L.M.: Incorporating negative information to process discovery of complex systems. *Inf. Sci.* **422**, 480–496 (2018)
27. Lo, D., Khoo, S., Liu, C.: Efficient mining of iterative patterns for software specification discovery. In: *KDD*, pp. 460–469. ACM (2007)
28. Maggi, F.M., Bose, R.P.J.C., van der Aalst, W.M.P.: Efficient discovery of understandable declarative process models from event logs. In: Ralyté, J., Franch, X., Brinkemper, S., Wrycza, S. (eds.) *CAiSE 2012*. LNCS, vol. 7328, pp. 270–285. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31095-9_18
29. Maggi, F.M., Bose, R.P.J.C., van der Aalst, W.M.P.: A knowledge-based integrated approach for discovering and repairing declare maps. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) *CAiSE 2013*. LNCS, vol. 7908, pp. 433–448. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38709-8_28
30. Maggi, F.M., Di Ciccio, C., Di Francescomarino, C., Kala, T.: Parallel algorithms for the automated discovery of declarative process models. *Inf. Syst.* **74**(Part), 136–152 (2018)
31. Maruster, L., Weijters, A.J.M.M., van der Aalst, W.M.P., van den Bosch, A.: A rule-based approach for process discovery: dealing with noise and imbalance in process logs. *Data Min. Knowl. Discov.* **13**(1), 67–87 (2006)
32. Neider, D., Gavran, I.: Learning linear temporal properties. In: *FMCAD*, pp. 1–10. IEEE (2018)
33. Nguyen, H., Dumas, M., La Rosa, M., Maggi, F.M., Suriadi, S.: Business process deviance mining: review and evaluation. *CoRR* abs/1608.08252 (2016)
34. Partington, A., Wynn, M.T., Suriadi, S., Ouyang, C., Karnon, J.: Process mining for clinical processes: a comparative analysis of four Australian hospitals. *ACM Trans. Manag. Inf. Syst.* **5**(4), 19:1–19:18 (2015)
35. 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

36. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), pp. 287–300. IEEE Computer Society (2007)
37. Riener, H.: Exact synthesis of LTL properties from traces. In: FDL, pp. 1–6. IEEE (2019)
38. Schunselaar, D.M.M., Maggi, F.M., Sidorova, N.: Patterns for a log-based strengthening of declarative compliance models. In: Derrick, J., Gnesi, S., Latella, D., Treharne, H. (eds.) IFM 2012. LNCS, vol. 7321, pp. 327–342. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30729-4_23
39. Slaats, T., Debois, S., Back, C.O.: Weighing the pros and cons: process discovery with negative examples. In: Polyvyanyy, A., Wynn, M.T., Looy, A.V., Reichert, M. (eds.) Business Process Management - 19th International Conference, BPM 2021, vol. 12875, pp. 47–64 (2021)
40. Suriadi, S., Mans, R.S., Wynn, M.T., Partington, A., Karnon, J.: Measuring patient flow variations: a cross-organisational process mining approach. In: Ouyang, C., Jung, J.-Y. (eds.) AP-BPM 2014. LNBIP, vol. 181, pp. 43–58. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08222-6_4
41. Suriadi, S., Wynn, M.T., Ouyang, C., ter Hofstede, A.H.M., van Dijk, N.J.: Understanding process behaviours in a large insurance company in Australia: a case study. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 449–464. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38709-8_29
42. Tessaris, S., Di Francescomarino, C., Chesani, F.: Negdis: code for the experiments (Aug 2021). <https://doi.org/10.5281/zenodo.5158527>
43. 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
44. Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering workflow models from event-based data using little thumb. *Integr. Comput. Aided Eng.* **10**(2), 151–162 (2003)

Process-Driven Applications



Semi-automated Test Migration for BPMN-Based Process-Driven Applications

Konrad Schneid¹ , Sebastian Thöne¹, and Herbert Kuchen² 

¹ Münster University of Applied Sciences, Münster, Germany
{konrad.schneid, sebastian.thoene}@fh-muenster.de

² University of Münster, Münster, Germany
kuchen@uni-muenster.de

Abstract. Automated regression tests are a key enabler for applying popular continuous software engineering techniques. This paper focuses on testing BPMN-based Process-Driven Applications (PDA). When evolving PDAs, the affected test cases must be identified and co-evolved as well. In this process, affected test cases can be overlooked, misunderstandings may occur during communication between different roles involved, and implementation errors can arise. Regardless of possible error sources, the entire test migration process is time-consuming. This paper presents a new semi-automated test migration process for PDAs. The concept builds on previous work on creating regression tests using a no-code approach. Our approach identifies the modifications of the PDA and classifies their impact on previously defined tests. The classification indicates whether existing test code can be migrated automatically or whether a manual revision becomes necessary. During an AB/BA experiment, the concept and the developed prototype proved a more efficient test migration process and a higher test quality.

Keywords: Test migration · Software evolution · Process-driven application · BPMN

1 Introduction

Automated regression tests are essential in developing applications, especially when practicing continuous software engineering techniques with short release intervals [5]. The development of such tests is very time-consuming and cost-intensive on the one hand and error-prone on the other. However, automated regression tests provide a significant return on investment by saving the costs of manual test repetitions [6]. Once the application is rolled out, development does not stop here. Software evolution implies a continual software adaptation, e.g., due to new requirements, new regulatory specifications, or process optimization [13]. In addition to the adaptation of the software itself, the migration of the existing regression tests cannot be neglected.

This paper elaborates a semi-automated test migration approach for Process-Driven Applications. These are hybrid systems which play an important role in business process management. PDAs are not only based on classic source code, but also on an executable business process model, typically using the ISO standard *BPMN (Business Process Model and Notation)* [7,8]. Running such applications requires a process engine such as *Camunda BPM* or *Activiti* [1,4]. Figure 1 shows two versions of an executable BPMN model, which we use as running example.

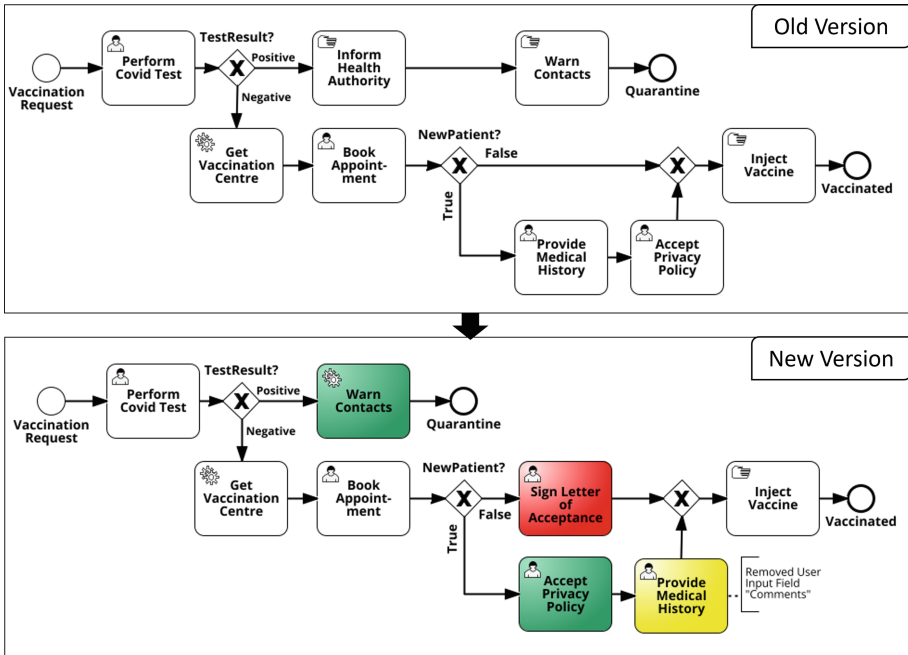


Fig. 1. Example of a BPMN-based PDA evolution for the treatment of a Covid vaccination. (Color figure online)

The process addresses a request for a Covid vaccination and consists of three distinct process paths: The first upper path ends in quarantine after a positive Covid test, whereby the middle process path ends with vaccination at a vaccination center where the patient is already known. The latter path also ends with a vaccination, but this time in a medical centre where the patient is new. The new version of the process model incorporates the following modifications: While the health authority was informed about positive tests in the old version, this task is removed in the new one. Furthermore, the task type of *Warn Contacts* has changed from manual to service task. The user task *Sign Letter of Acceptance* has been added to the middle path, requiring new user inputs. Within the latter path, the

order of user tasks *Accept Privacy Policy* and *Provide Medical History* has been swapped, and a user input field has been removed from *Provide Medical History*.

Typically, such changes cause a test migration which involves two roles with different skills and backgrounds: The *process analyst*, responsible for business logic and test specification, identifies affected test cases and adapts the corresponding test specification. The *process engineer*, responsible for implementing the tests, rewrites the corresponding test code.

However, the process analyst could overlook affected tests or misinterpret the modification's impact. Technical details such as variable declarations, evaluable expressions, and references to user forms or software modules are stored in the underlying XML representation of the process model. As BPMN does not provide any visual representation for these technical configurations, this information might not be accessible by the process analyst. Since the process analyst typically lacks the programming skills to adjust the test code, the task must be delegated to the process engineer. There is a risk that information is misinterpreted or even lost during communication. Finally, the developer may make implementation mistakes when manually adapting the test code. Regardless of possible sources of error, the entire test migration process is very time-consuming.

To improve the situation described above, our research pursues the following objectives:

1. The solution should automatically identify affected regression tests after modification of PDA artifacts.
2. The impact on the affected tests should be classified into three categories: a) The test can automatically be migrated and requires no further intervention. b) The test can automatically be migrated, but a manual release is recommended due to the impact of the modification. c) The test becomes incompatible to the new PDA release and must be adapted manually.
3. The manual test migration process should be wizard-based and allow the user to specify test cases without any programming knowledge. The required test code has to be generated based on the adapted test specification. In this manner, the test migration process is manageable by the process analyst without support by the process engineer.
4. The approach should support at least the BPMN extended core elements according to RECKER, which occur in more than every fourth model [15]. Additionally, decision models in DMN notation and linked views should be considered, too. Source code referenced in service tasks is not (yet) analyzed in this state.

Our solution builds on prior work for test specification and subsequent test code generation using a no-code approach [17]. To achieve our objective, we must first identify which kinds of modifications can occur in PDAs. In the second step, we need to analyze the impact of the identified modification types on already existing tests. This allows us to classify tests that have already been created whether they can be migrated automatically or whether manual changes are necessary.

Our approach not only saves time and hence costs, but also increases the test quality. We have demonstrated the desired effects in an AB/BA experiment.

This paper is structured as follows: Sect. 2 gives an outline of our previous no-code test specification and generation approach, which serves as the starting point of this paper. While Sect. 3 summarizes related work, Sect. 4 provides a general overview of our approach. Section 5 deals with the comparison of two versions of a business process. In the next section, we discuss in detail the possible types of modifications and their potential impact on existing tests. The final semi-automated test migration process is explained in Sect. 7. Section 8 covers the implementation of our prototype and the evaluation. Finally, in Sect. 9 we conclude and point out future work.

2 No-Code Approach for Creating Regression Tests

In a previous paper [17], we addressed the problem that implementing automated regression tests for PDAs is time-consuming and error-prone. Since this article builds on that approach, we provide a brief introduction.

Our no-code approach for a wizard-based specification of test cases and subsequent automated test code generation can be divided into three phases, as illustrated in Fig. 2. In the first phase, the PDA artifacts such as the process model, decision tables, or referenced views are analyzed to obtain all possible end-to-end execution paths of the business process. During the analysis, all data fields that determine the control flow along such a path are identified. Furthermore, corresponding value suggestions are elaborated for the data fields by looking at constraints for the considered path to support the later specification.

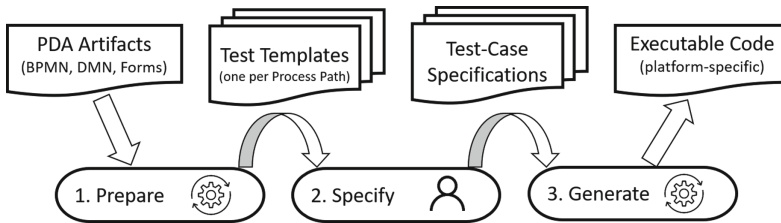


Fig. 2. Overview of our no-code approach for semi-automated tests creation adapted from [17].

A depth-first search for identification of possible process paths is applied, whereby special treatment of constructs such as parallel gateways and loops is necessary. For each identified path, a so-called *test template* is created. In addition to the sequence of BPMN elements of the end-to-end process path, the test template contains all data fields identified on that path. As a third component, the template includes the elaborated value suggestions, presumably leading to

the corresponding process path during execution. The template contains all information about required test data for one end-to-end process path, which must be specified with concrete values to create one or more corresponding tests. The test templates are persisted in a platform-independent, domain-specific language (DSL). Listing 1 reveals an excerpt with elements of the middle process path of our running example, leading to vaccination in some known vaccination center.

```

1 Flow Flow_1 for process
2 with elements : [ VaccinationRequest, PerformCovidTest,
    ↪ GatewayTestResultStart, GetVaccinationCentre,
    ↪ BookAppointment, GatewayNewPatientStart,
    ↪ GatewayNewPatientEnd, InjectVaccine, Vaccinated ]
3 without elements : [ ... ] ;

```

Listing 1. Excerpt of a Test Template in the DSL Definition.

In the second phase, the process analyst refines the generated test templates to concrete regression tests by specifying all user inputs and expected outputs for each selected process path. This task is supported by a wizard which proposes suitable test values elaborated during the analysis. The resulting test specification is persisted in a platform-independent DSL format again (see Listing 2).

```

1 Test Vaccinated for Flow_1
2 with variables for VaccinationRequest : [ Name = "John_Smith"
    ↪ , DateOfBirth = "01.05.1990" ],
3 with variables for BookAppointment : [ NewPatient = false ,
    ↪ Appointment = "01.05.2022-15:00" ];

```

Listing 2. Exemplary Test Specification for the Test Template from Listing 1.

In the third phase, the code generator comes into action. The platform-independent test specifications from phase two are automatically transferred into test code for a specific process engine. In our case, the concept was prototypically implemented for the Camunda BPM platform. The generated JUnit tests start a new process instance and mock all tasks with user interaction on the considered path with the specified test data. The test also includes assertions which check whether the intended path is traversed during execution. This no-code approach enables the process analyst to create test code independently and efficiently without programming knowledge.

3 Related Work

Several approaches already exist in verification, testing, and evaluation of PDAs. However, these do not fully meet our objectives so far. In this Section, we give a brief overview of related work.

Extensive research has been conducted on the verification of process models [12]. By using static analysis techniques, structural and behavioral correctness can be checked at an early stage during design time. E.g., we dealt with uncovering inconsistencies between process model and dependent source code artifacts

such as broken code references or data-flow anomalies using statical analysis in [16, 18]. The interaction between executable process model, referenced source code, and third-party APIs plays an essential role in the case of PDAs. However, third-party services are often black boxes that cannot be analyzed statically. Therefore, dynamic tests such as executable regression tests are necessary, which is the focus of this paper.

Testing of PDAs at runtime has also received attention in recent research [25]. DE MOURE ET AL. present a concept for generating test tables based on BPMN process models [14]. These can be further refined for generating initial test scripts for Selenium and Cucumber. The concept does neither consider DMN models nor support the specification process by value-help. YOTYAWILAI and SUWANASART's solution generates test templates based on BPMN models [24], which can be specified by the process analyst and used for test code generation as well. This approach does not address constructs such as parallelism, loops, or DMN models. Besides a lack of value-help during the specification process, a prototypical implementation and an evaluation are not provided. Due to this research gap, we developed the concept presented in Sect. 2 in an earlier iteration.

Research was also conducted in the area of evaluation of PDAs and their process models. The concept of KHERBOUCHE ET AL. aims to support change management [9]. Their dependency-centric approach analyzes the impact during business process evolution. The analysis provides metrics to evaluate the impact during change management more profoundly. The detection and manual resolution of differences between process model versions is examined by KÜSTER ET AL. [10]. This approach is based on the SESE fragment decomposition and is intended to support variant and version management. However, testing is not considered in these two concepts.

During PDA evolution, not only the software itself must be considered, but also tests that have already been created before. The efficient selection of test cases of the process model is investigated by BÖHMER and RINDERLE-MA [3]. They combine generic metrics based on the process model with historical data to reduce the execution of all tests available to a minimum. WANG ET AL. present a modification impact-based test prioritization for PDAs [21]. Running the test in a specific order should detect failure as early as possible. Their prioritization is based on the fault propagation behavior of adaptations and the internal structure of the PDA. Neither paper addresses the migration of test code.

TIPPAPHARAT AND SUWANNASART have developed a concept for a test impact analysis of BPMN models [19]. This approach focuses on the modification of user input fields. Affected test cases are identified and, if necessary, re-specified with random test data. A test code generation is missing along with examining other modification types on PDA artifacts. BEHRANG AND ORSO examine automated test migration for mobile apps. The authors consider tests for similar functions used in different apps [2]. The developed AppTestMigrator can be used to transfer tests for comparable functions in different Android apps. It is not related to BPMN. LEHNERT ET AL. have created a rule-based approach to determine the impact across multiple artifacts [11]. In this process, UML diagrams,

Java source code, and JUnit tests are merged into a meta-model to map the dependencies. The migration of tests is not part of the concept.

4 Overview of the Solution

This section is intended to provide an understanding of the overall solution. Our concept can be subdivided, as illustrated in Fig. 3, into four phases: First, the end-to-end process paths (called *flow* in the sequel) of the new version of the process model must be identified and, for comparability, assigned to the flows of the old version. In phase two, modifications on each flow can be determined and their impact on the previously created tests elaborated. The third phase is threefold: Depending on the modification impact, the tests will either be migrated automatically, or a manual release step is placed before the automatic migration, or a manual intervention is required, if a test case becomes incomplete or invalid by the current change. Finally, in the fourth phase, the corresponding test code is generated.

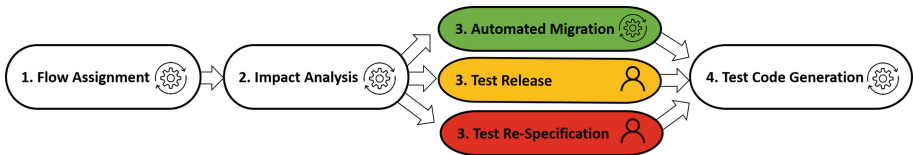


Fig. 3. Semi-automated four-stage test migration process at a glance.

The first phase is to establish comparability between the old and new versions, as this is the only way to determine modifications on each flow. For this purpose, all flows of the two versions, including the data fields and value proposals, are elaborated analogously to the test template creation process described in Sect. 2. After that, the flows of the new version are assigned to the flows of the old version based on a similarity matching. This assignment is an automated step, but can still be corrected by the process analyst.

Modifications can be determined in the second phase based on the assignment of the flows between the two versions. Furthermore, the impact of the modifications on existing tests can be analyzed. For this purpose, all process elements within a pair of matched flows are compared one by one. When a modification is detected, its impact on already specified tests has to be classified. We developed a set of rules in order to categorize each type of modification according to a traffic light system:

- *Green*: Modification allows an automated migration and does not require any further intervention.
- *Yellow*: Modification allows an automated migration. However, checking the impact manually is recommended.

- *Red*: Modification requires manual intervention due to new mandatory data input fields or violating constraints with existing test data.

Once all modifications within a flow have been checked and classified individually, an aggregated flag for the flow can be derived from the maximum impact flag of all flow elements. The flow classification forms the basis for classifying the existing test cases: As each test case belongs to one of the flows, it gets at minimum the aggregated flag of the corresponding flow. However, if some formerly specified test data violates (new) constraints and has to be rewritten, the individual test case might get a higher classification than the underlying flow.

The final migration of already existing tests and the subsequent test code generation take place in the third phase. Green flagged test specifications are migrated directly into the new scheme, whereby yellow flagged tests require a manual release step. Invalid or incomplete test specifications (red flag) can be re-specified wizard-based, supported by value suggestions for each required user input. As soon as a test case has been completed, the implementation code for a specific process engine is generated in the final phase.

The following sections provide a detailed explanation of the flow matching algorithm (Sect. 5), the impact analysis (Sect. 6), and the migration and code generation phase (Sect. 7).

5 Flow Assignment

Given an old process schema p_1 and a new schema p_2 , the intention of this phase is to find flows in p_2 again, despite any changes made to the old schema p_1 . This problem can be formalized as finding a matching function between the two sets of flows. For this purpose, we compute the similarity between all flows of the two schema versions.

A flow represents a possible execution run of a process model and forms the basis of a test template (see Sect. 2). Formally¹, we can define a flow $f = (e_1, e_2, \dots, e_n)$ over process model p as a tuple, where e_1 is start event, e_n is end event, and all pairs (e_i, e_{i+1}) are process elements connected by a sequence flow in p (where $1 \leq i < n$).

The similarity between two flows $a = (a_1, a_2, \dots, a_n)$ over p_1 and $b = (b_1, b_2, \dots, b_m)$ over p_2 can be measured by counting the twin elements, i.e. those with same unique id (provided by the modeling tool) or same label (which is weaker) but also same neighbors. Formally, we define a function $twin_{ab} : \{1, 2, \dots, n\} \rightarrow \{0, 1\}$ as

$$twin_{ab}(i) = \begin{cases} 1 & \text{if } \exists b_j \in b \text{ with } (a_i.id = b_j.id) \vee \\ & (a_i.label = b_j.label \wedge \\ & a_{i-1}.id = b_{j-1}.id \wedge \\ & a_{i+1}.id = b_{j+1}.id) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

¹ For brevity, we omit parallel sections of a flow in the definitions presented here.

which returns 1 for every element in a which has got a twin in b , and aggregate the outcome as average value over all elements in a as

$$similarity(a, b) = \sum_{i=1}^n \frac{twin_{ab}(i)}{n} \quad (2)$$

A simple approach for computing the similarity could compare every element of a to every element of b leading to a complexity of $O(n \cdot m)$. If necessary, this could be improved by sorting. In our previous work [17], we demonstrated how to compute the set of all relevant flows for a given process model. Hence, let F_1 and F_2 be these sets for p_1 and p_2 , respectively. Then, we can compute a function $matching : F_1 \rightarrow F_2$ in $O(|F_1| \cdot |F_2| \cdot |p_1| \cdot |p_2|)$ time which assigns to every flow $a \in F_1$ the most similar flow $b \in F_2$ satisfying:

$$similarity(a, b) = \underset{\beta \in F_2}{Max}(similarity(a, \beta)) \quad (3)$$

However, a minimum match rate can be set to prevent false matching (by default at 60%). If a flow of the new version is not matched above the minimum rate, it is classified as a new flow. Test cases can be specified for previously non-existent flows as explained in Sect. 2. If a flow from the old version has no match above the minimum, it will be considered as no longer existing, and the related test template and test cases are deleted. The flow matching is suggested to the process analyst for manual correction if necessary.

6 Test Impact Analysis

After the assignment between flows of the old and new PDA versions has been established, the impact analysis can be conducted. The analysis is twofold: First, the types of modifications that may occur within a PDA need to be elicited. These are then analyzed for their potential impact on tests, resulting in a classification system that distinguishes between tests that can be migrated automatically and those requiring manual intervention. A test cannot be migrated automatically if the previously specified test data is incomplete (e.g., due to new mandatory user input fields) or invalid (e.g., due to new constraints). In a second step, this information can be used to identify and categorize modifications between two PDA versions. This analysis is mandatory for the intended semi-automated test migration. In this section, we first look deeper into the impact analysis at the flow level, while the second part deals in more detail with the analysis on the test level.

6.1 Impact Analysis at Flow Level

WEBER ET AL. identified 18 modification patterns for PDAs [22]. We have generalized the possible modifications within a flow as follows: Flow elements can be added, removed, or modified. Furthermore, the element sequence within the

flow can be changed. In the following paragraphs, we survey the four different types of changes. A simplified overview of the classification rules according to the traffic light system described in Sect. 4 is depicted in Fig. 4. The flags resulting from the classification are then persisted by an extension to the DSL we proposed in [17].

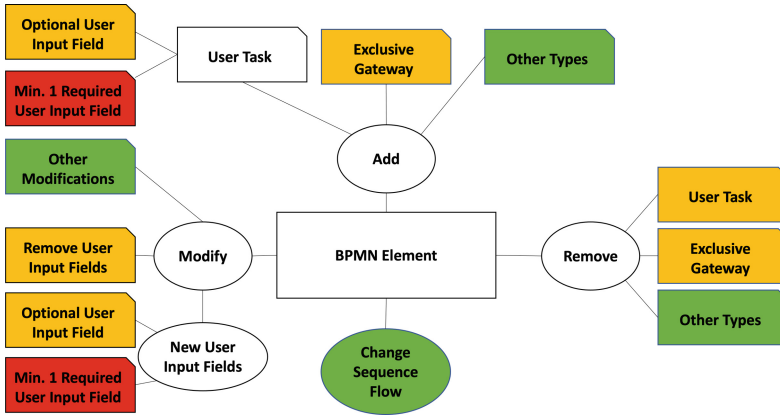


Fig. 4. Simplified representation of flow-level classification rules. (Color figure online)

Change of the Flow Sequence: A frequently used change for process optimization is swapping the sequence within a flow [23]. The aim is to achieve faster processing times or fewer correction loops. Concerning the tests, the impact is uncritical and allows an automated migration. The previously specified test data would be, in this scenario, still complete and valid. In our process example, we have reversed the order of the two user tasks *Accept Privacy Policy* and *Provide Medical History* (cf. Fig. 1). Thus, regardless of the element type, such a modification can be automatically migrated and is always flagged green.

Removing a Flow Element: A process step may be removed from a flow since it is no longer required (at this point). Looking at the previous release, deleting an element on a flow does not require additional test data for tests considering that flow. We have removed the task *Inform Health Authority* from the upper flow in our running example. However, the specified test data for the considered flow would still be complete and valid compared to the previous version. Therefore, this type of modification allows automated test migration and is classified green.

Nevertheless, we have identified two constellations constituting an exception to this regulation. If a process step is removed requiring user inputs (typically a user task) and thus test data, we consider the modification more stringently. Although automatic migration is still possible, previously specified test data for

this step would be deleted. Since removing these test data may lead to unintended business behavior, we recommend checking the migrated tests manually by the process analyst. Hence, we would flag this modification constellation yellow, requiring a manual release step. The effects of removing an exclusive gateway are also more substantial. These gateways typically have two or more outgoing sequences with different business behaviors, which are removed this way. Therefore, this type of modification is also flagged yellow.

Adding a New Flow Element: A more precise distinction must be applied when adding a new element to a flow. The classification depends primarily on the required user interaction in this process step. BPMN activities such as service, script, or business rule tasks do not require user interaction and are executed automatically during regression tests. Adding these types of activities keeps the test definition complete and enables automated migration, which is why it is classified green accordingly.

The addition of parallel gateways is also classified with a green flag. As pointed out in [17], the execution order within parallel flow sections cannot be statically determined. We assume that the order of processing within the parallel branch is irrelevant, since we can eliminate race conditions based on static analysis approaches [16]. The parallel gateway is passed during the test and does not require further test specifications. The situation is similar for exclusive gateways, where the element can be migrated automatically. However, it should be noted that two or more independent paths are subsequently created. This is not considered here but was already analyzed during the flow matching algorithm (see Sect. 5), whereby new outgoing paths from the gateway are defined as new flows. The yellow flag is applied for the previously existing flow as a manual check is recommended in this situation.

A further distinction is made in the classification for added activities with required user input, as is generally the case for user tasks. User input fields within a user task activity can be optional or mandatory. If the newly added task on the flow contains only optional user input fields, this modification would technically be automatically migratable. Since we cannot technically classify the business significance of the user input, we recommend a manual release according to the yellow flag classification.

```

1  <userTask id="SignLetterOfAcceptance">
2      <camunda:formField id="signFlag" type="boolean">
3          <camunda:constraint name="required" />
4      </camunda:formField>
5      <camunda:formField id="signDate" type="date">
6          <camunda:constraint name="required" />
7      </camunda:formField>
8  </userTask>

```

Listing 3. Excerpt of the User Task Definition *Sign Letter Of Acceptance* with two mandatory Fields.

A red impact classification is assigned to an added task if it contains at least one mandatory input field. The test specification would be incomplete, and the generated test code would be suspended during execution or fail. Here, manual intervention by the process analyst is necessary. We have such a case in our running example, whereby the user task *Sign Letter of Acceptance* is added. The new activity provides two new mandatory user-input fields, as demonstrated in the XML representation of the task in Listing 3.

Modification of an Existing Flow Element: Besides the previously mentioned possibilities of adapting a flow, there is also the option of modifying an already existing process element. BPMN activities without user interaction and thus without influence on the required test data are classified green. In our process example, the element type changes for *Warn Contacts* to the type service task, which can be migrated automatically since no additional test data is required.

The classification of modifications on existing elements with the change of at least one user input field is distinguished between optional and mandatory input fields. The treatment is quite analogous to the previous cases: If only optional user inputs are added or removed, this modification is classified yellow, as it is the case for *Provide Medical History* in our running example. In the case of adding at least one mandatory user input field, tests created for the flow cannot be migrated automatically, resulting in a red flag classification. The process analyst must specify the missing user input in this case for the previously created tests.

6.2 Aggregation at Flow Level and Derivation at Test Level

Once all flow elements have been classified in terms of their test impact, an aggregated classification for the flow can be derived as the maximum rating of all contained flow elements. The information about the new flows, including the flags, is persisted in a platform-independent DSL.

```

1 Flow Flow_1 for process with flag: RED
2 with elements : [ VaccinationRequest, ... ,
   ↪ SignLetterOfAcceptance: RED, ..., Vaccinated ]
3 without elements : [ ... ] ;
4
5 Test Vaccinated for Flow_1 with flag: RED
6 with variables for VaccinationRequest : [ Name = "John_Smith"
   ↪ , DateOfBirth = "01.05.1990" ], ...,
7 with variables for SignLetterOfAcceptance : [ ];

```

Listing 4. Excerpt of a Test-Template and Test Definition after the Impact Analysis.

The upper half of Listing 4 demonstrates this for the middle flow again, whose flag is red, because the new user task *SignLetterOfAcceptance* has mandatory input fields and requires manual intervention by the process analyst. The impact classification of individual test cases is preliminarily copied from the underlying

flow (see lower half of Listing 4). However, that classification is increased, if formerly specified test data violate (new) constraints, such as a new number range for a user input field. In that case, the specified user input is cleared, and both the element at hand and, as a consequence, the entire test case receive the red impact flag.

After all modifications have been identified and classified according to the traffic light system, the semi-automated migration can be performed in the next phase.

7 Migration and Test Code Generation

The impact analysis described above is a prerequisite for the intended test migration. Based on the assigned classification per test according to the traffic light system, the tests can be migrated automatically directly (green flag), after a release step (yellow flag), or after adaptation of the test specification (red flag). Analogously to the creation of the test templates (see Sect. 2), the new flow schemes are persisted in a DSL, as are the tests themselves. Based on this platform-independent test specification, process-engine specific test code can be generated automatically.

After the impact analysis has been completed, the tests need to be transferred to the new flow scheme as far as possible. All tests are flagged by their modification impact, and test data is removed, if elements are omitted or constraints are violated. Similarly, test templates and linked test cases are removed, if their underlying flow does no longer exist. The tests marked with a green flag are released for code generation without any approval step. This way, the corresponding new test code is immediately available after the impact analysis.

A yellow flagged test has to be inspected and released manually via the user interface, changing the flag to green and conducting the automatic code generation. Alternatively, the process analyst has the option to modify the test. In this case, value suggestions support the specification of required user inputs. In this context, the test data is validated during the wizard-based re-specification. If the test data are valid, the test is flagged green, and the test code generation is performed. Otherwise, the tests will be flagged red, and the incorrect entries will be underlined. A test flagged red during impact analysis must be edited manually in any case, as the test data is invalid or incomplete. The test code to be generated would be incomplete and would result in an incorrect test code. The re-specification of the test takes place in the same way as in the previously described scenario of yellow flagged tests.

Overall, the migration process for tests has a high degree of automation. Programming knowledge is not necessary due to the underlying no-code approach. If manual intervention by the process analyst is required, the task is performed wizard-based. In addition, the process analyst is supported by value suggestions for the corresponding flow for each required user input. The test templates and test cases stored in a DSL trigger a test code generator after adaptation, generating the desired platform-specific test code.

8 Evaluation

The presented concept for semi-automated migration is platform-independent and can be implemented for any process engine. We developed a prototype² for the Camunda BPM platform.

After adapting PDA artifacts, the analyst starts the migration process using the user interface realized as a client-sided web application. A Spring-based backend parses the artifacts and performs the process flow matching algorithm (cf. Sect. 5), which can be confirmed or adjusted by the analyst. Then, the backend performs the impact analysis and migration (cf. Sect. 6, 7). Finally, the analyst can re-specify test cases that must be adapted via the web client. The test case specification is stored in a DSL developed with Xtext, triggering the test code generator using the Camunda BPM assert library after each modification.

Our evaluation is twofold: In an AB/BA crossover design [20] experiment, 14 participants completed a migration process in two rounds, once using the tool and once manually. In the experiment, we measured the time required for migration and assessed the test code quality. Furthermore, 21 participants used the tool in various case studies and completed a questionnaire structured according to the Likert scale afterward. The questionnaire focuses on the usability and the functional benefits of the tool.

8.1 Experiment Setting

We evaluated our approach with a practical AB/BA experiment. A total of 14 volunteers with an average programming experience of over six years participated in the experiment. First, the participants received a brief introduction to the manual development of JUnit tests for PDAs and to our tool. Subsequently, the group was randomly divided into two groups of equal size. For each of the two rounds, different modified PDAs were prepared, requiring the existing test code to be adapted. The process models consist of five flows with 18 (round 1) or 20 (round 2) activities. For each flow one test has been implemented before. The PDAs were subject to six modifications. Of these changes, three cases were flagged red using the tool and had to be manually adjusted. One test case was categorized yellow, leading to a manual release step, and one test could be migrated automatically. The groups differ as follows: Group I used the tool during the first round only, and group II during the second round only. Participants had to commit their implementation to Git repositories provided before the round started and after the migration was completed. The timestamps of the commits can be used to measure the time required for the migration. The commit triggers a small CI pipeline validating the migrated tests.

8.2 Results and Discussion

The metrics gained from the experiment and the questionnaire results show increased productivity, a higher test-code quality, and a good usability. Since

² Available at <https://git.fh-muenster.de/winfo/code-pro/pda-testing-framework..>

the automatic migration was run through within a few seconds, even for larger models, there is no separate performance analysis. The complete results of the experiment and the questionnaire are available in a public repository³ and are presented in excerpts in the following.

Table 1. Required time and rate of valid migrations.

Approach	Time required (in minutes)	Valid test rate
Manual	36:47	82%
Tool	6:13	100%

Feasibility of Our Tool: The modifications in the case studies were correctly recognized by the tool, and the categorization according to the traffic light system was performed as desired (cf. objective no. 1 & 2). The data gathered from the experiment allows conclusions to be drawn about the efficiency of the migration process and the quality of the test code. The migration process was completed six times faster with the tool than the manual adaptation of the test code. While the tool-based migration took, on average, around six minutes in the two rounds, the manual adjustment required over 36 min (cf. Table 1). The strengths of our concept could also be proven concerning the test code quality. While all tests migrated via the tool were correct, only 82% of the manually edited tests were valid. The invalid tests result from implementation errors that were probably not uncovered without running the test framework (JUnit). The implemented prototype is limited to the extended core elements of BPMN, DMN models, and linked views (cf. objective no. 4). However, this limitation is sufficient to demonstrate the feasibility of the concept.

Usability: We evaluated the tool’s usability using the questionnaire structured according to the Likert scale. Therefore, 21 participants performed test migrations using various case studies after a brief introduction to the prototype. Migrating tests with the tool was considered to be intuitive by more than 85 % of the participants. The tests could be migrated and re-specified without the need for programming skills (cf. objective no. 3). Over 85 % of respondents (fully) agreed that it was obvious which elements of an existing test case need to be re-specified or checked. Only a few questions were asked about the use of the tool. However, participants also suggested improvements, such as highlighting the buttons.

Limitations: Our approach for semi-automated migration of regression tests for PDA was convincing during the evaluation conducted. However, there are also limitations. The concept currently does not consider modifications in referenced source code or other third-party systems. This analysis extension is necessary to consider the PDA in its entirety. For industrial use, the prototype should cover a

³ Available at <https://fh-muenster.sciebo.de/s/6ZEmMeo6Quh6vaH>.

broader range of elements than the extended core of BPMN. The prioritization or selection of test cases can also be derived based on the modification impact analysis presented in this paper, which will be considered in the future.

9 Summary and Outlook

Automated regression testing is becoming increasingly important in software development, not at least due to the widespread practice of continuous software engineering. Applications are subject to constant adaptations to optimize them or cover new requirements. Not only the software artifacts themselves have to be changed, but also the affected test code. The migration of existing regression tests is a time-consuming and error-prone process.

We have developed a semi-automated test migration approach for a specific software domain, called BPMN-based Process-Driven Applications. The approach is based on a previous no-code concept for creating test code, eliminating the need for programming knowledge during test migration. The concept can be divided into the following steps. First, the flows of the new version must be determined and, for comparability, matched with the flows of the previous PDA version. For the intended impact analysis, we have analyzed what kind of changes can occur and what impact these changes may have on the existing test cases. We have derived a set of rules that classifies the changes according to a traffic light system, distinguishing the impact between (fully) automated migration and required manual intervention. The automated and manually modified test cases are persisted in a platform-independent DSL. Out of the DSL, process-engine specific test code can be generated automatically. This way, programming knowledge is not required. Based on our approach, a prototype for the Camunda BPM platform has been developed and evaluated in a AB/BA crossover design experiment. Our concept convinced with great time savings and an increased test code quality. In future work, the concept will be extended by a test prioritization approach, allowing to find bugs as early as possible. Furthermore, we plan to extend our analysis by examining referenced source code and related external systems.

References

1. Activiti: Open source business automation (2022). <https://www.activiti.org>
2. Behrang, F., Orso, A.: Automated test migration for mobile apps. In: International Conference on Software Engineering: Companion Proceedings, pp. 384–385 (2018)
3. Böhmer, K., Rinderle-Ma, S.: Automatic business process test case selection: coverage metrics, algorithms, and performance optimizations. *Int. J. Cooperative Inf. Syst.* **25**(04) (2016)
4. Camunda: Workflow and decision automation platform—Camunda BPM (2022). <https://camunda.com>
5. Elbaum, S., Rothermel, G., Penix, J.: Techniques for improving regression testing in continuous integration development environments. In: International Symposium on Foundations of Software Engineering, pp. 235–245 (2014)

6. Graham, D., Fewster, M.: Experiences of Test Automation: Case Studies of Software Test Automation. Addison-Wesley Professional (2012)
7. Harmon, P., Wolf, C.: Business Process Modeling Survey. BPTrends Report (2011)
8. ISO: ISO/IEC 19510:2013 - business process model and notation (2013). <https://www.iso.org/standard/62652.html>
9. Kherbouche, O.M., Ahmad, A., Bouneffa, M., Basson, H.: Analyzing the ripple effects of change in business process models. In: Inmic, pp. 31–36. IEEE (2013)
10. Küster, J.M., Gerth, C., Förster, A., Engels, G.: Detecting and resolving process model differences in the absence of a change log. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 244–260. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85758-7_19
11. Lehnert, S., Riebisch, M., et al.: Rule-based impact analysis for heterogeneous software artifacts. In: European Conference on Software Maintenance and Reengineering, pp. 209–218. IEEE (2013)
12. Mendling, J.: Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness, vol. 6. Springer, Heidelberg (2008)
13. Mens, T., Wermelinger, M., Ducasse, S., Demeyer, S., Hirschfeld, R., Jazayeri, M.: Challenges in software evolution. In: International Workshop on Principles of Software Evolution, pp. 13–22 (2005)
14. de Moura, J.L., Charão, A.S., Lima, J.C.D., de Oliveira Stein, B.: Test case generation from BPMN models for automated testing of web-based bpm applications. In: International Conference on Computational Science and Its Applications, pp. 1–7 (2017)
15. Recker, J.: Opportunities and constraints: the current struggle with BPMN. *Bus. Process Manag. J.* **16**, 181–201 (2010)
16. Schneid, K., Kuchen, H., Thöne, S., Di Bernardo, S.: Uncovering data-flow anomalies in BPMN-based process-driven applications. In: Symposium on Applied Computing, p. 1504–1512. ACM (2021)
17. Schneid, K., Stapper, L., Thöne, S., Kuchen, H.: Automated regression tests: a no-code approach for BPMN-based process-driven applications. In: International Enterprise Distributed Object Computing Conference, pp. 31–40. IEEE (2021)
18. Schneid, K., Usener, C.A., Thöne, S., Kuchen, H., Tophinke, C.: Static analysis of BPMN-based process-driven applications. In: Symposium on Applied Computing, pp. 66–74. ACM (2019)
19. Tippapharat, P., Suwannasart, T.: Test case impact analysis for BPMN input changes. In: Asia Service Sciences and Software Engineering Conference, pp. 70–74 (2020)
20. Vegas, S., Apa, C., Juristo, N.: Crossover designs in software engineering experiments: benefits and perils. *IEEE Trans. Softw. Eng.* **42**(2), 120–135 (2016)
21. Wang, H., Xing, J., Yang, Q., Han, D., Zhang, X.: Modification impact analysis based test case prioritization for regression testing of service-oriented workflow applications. In: Annual Computer Software and Applications Conference, vol. 2, pp. 288–297 (2015)
22. Weber, B., Reichert, M., Rinderle-Ma, S.: Change patterns and change support features - enhancing flexibility in process-aware information systems. *Data Knowl. Eng.* **66**(3), 438–466 (2008)
23. Whitfield, R.I., Duffy, A.H.B., Coates, G., Hills, W.: Efficient process optimization. *Concurr. Eng.* **11**(2), 83–92 (2003)

24. Yotyawilai, P., Suwannasart, T.: Design of a tool for generating test cases from BPMN. In: 2014 International Conference on Data and Software Engineering, pp. 1–6 (2014)
25. Zakaria, Z., Atan, R., Ghani, A.A.A., Sani, N.F.M.: Unit testing approaches for BPEL: a systematic review. In: Asia-Pacific Software Engineering Conference, pp. 316–322. IEEE (2009)



Splitting Quantum-Classical Scripts for the Generation of Quantum Workflows

Daniel Vietz^(✉) , Johanna Barzen , Frank Leymann ,
and Benjamin Weder 

Institute of Architecture of Applications Systems, University of Stuttgart,
Universitätsstraße 38, 70569 Stuttgart, Germany
{vietz,barzen,leymann,weder}@iaas.uni-stuttgart.de

Abstract. Quantum applications play an essential role in exploiting the full potential of quantum computers. However, quantum applications are typically hybrid, i.e., they consist of classical and quantum processing steps that must be orchestrated. Workflow technologies enable such orchestration and offer several advantages such as robustness, reusability, and scalability. Nowadays, many quantum applications are only available as script-based implementations. Thus, to benefit from the advantages of workflow technologies, a transformation is necessary; one driving question thereby is how to split hybrid quantum applications into their quantum and classical parts. To address this, we propose an approach that automatically generates equivalent workflow models from existing script-based quantum implementations. Based on syntactical analysis, it first splits the script-based implementations into their quantum and classical parts. These parts are then orchestrated by a generated workflow model that resembles the original execution order of the script and ensures the correct data flow between them. Furthermore, we generate deployment models for each part to enable automated deployment. To validate the practical feasibility of our approach, we present a prototype and a case study examining a concrete quantum algorithm implementation.

Keywords: Workflow technologies · Workflow generation · Quantum computing · Hybrid quantum algorithms · Quantum-classical split

1 Introduction

Quantum computing introduces a new computing paradigm, which promises to solve many problems faster, more precisely, or with less energy-consumption than classical computing [4, 12, 28]. One prerequisite to take advantage of these benefits is the implementation of quantum algorithms, which is often done in scripts targeting single use cases. With the continuous advancement of quantum computers, real-world use cases become more feasible, and the need for quantum software in industry is growing just as steadily [15, 28]. To meet this need, quantum software engineering has to involve aspects such as quality, reusability, integration, and maintainability [34, 45, 52].

The execution of a quantum algorithm is hybrid, i.e., it requires additional classical steps for pre- and post-processing [21, 23]. Thereby, the integration of

quantum and classical steps is a major challenge [44]. For example, this is due to the heterogeneous implementations (caused by the multitude of different technologies, e.g., programming languages) and the need for expertise from various areas [45]. The usage of workflow technologies offers several advantages over manual integration, such as reusability, scalability, and robustness. However, modeling quantum applications with workflows is complex, time-consuming, and requires a lot of expertise. To overcome this issue, QuantME [50] was developed as a modeling extension for imperative workflow languages. Nevertheless, quantum applications are often not available as workflows but as monolithic script implementations. A workflow-based re-implementation of such script implementations is time-consuming and manual conversion is cumbersome and not trivial, raising the challenge of how quantum script implementations can be automatically split into its quantum and classical parts to be orchestrated by a workflow. Thus, the main research question of this work can be formulated as follows:

How can quantum workflows be generated from existing script-based quantum implementations in an automated manner to benefit from the advantages of workflow technologies?

To address this research question, it is necessary to automatically analyze quantum implementations on the one hand. On the other hand, analyzed quantum implementations need to be broken up into multiple parts, and an executable workflow model needs to be generated. Thus, we subdivide the research question into the following questions tackling the two main challenges:

How can script-based quantum implementations automatically be analyzed to detect their quantum and classical parts?

How can different parts of a script-based implementation be extracted, packaged, automatically deployed, and orchestrated by a workflow model?

We address the research questions by presenting an approach that (i) analyses and splits script implementations into its quantum and classical parts in an automated manner, (ii) orchestrates the resulting script parts in a standards-based workflow model retaining the execution order of the script parts in the original implementation, and (iii) generates appropriate deployment models. Thus, the approach is able to generate quantum workflows from a script-based quantum implementation as input. All script parts can be automatically deployed and reused by other applications. Furthermore, we demonstrate the practical feasibility of our approach by a case study in which we examine our prototype and apply our method to split a script-based implementation of Shor’s algorithm [38].

After having covered the basic fundamentals in Sect. 2, Sect. 3 introduces the script splitting and workflow generation method, Sect. 4 presents our prototype, Sect. 5 evaluates our approach in a case study, and Sect. 6 discusses the limitations of our approach. Finally, Sect. 7 positions related work, and Sect. 8 provides a conclusion to the paper.

2 Fundamentals

In this section, we introduce fundamentals about hybrid quantum algorithms and their implementation. We also discuss quantum workflows and the automation of deployment using deployment models.

2.1 Hybrid Quantum Algorithms

Most quantum algorithms are hybrid [21], even the famous Shor algorithm [38] is hybrid [3]. Furthermore, due to the limitations of quantum computers [20, 35], today's focus is on variational algorithms that are inherently hybrid. Common examples are the *Variational Quantum Eigensolver (VQE)* for approximating eigenvalues of a matrix [33], and the *Quantum Approximation Optimization Algorithm (QAOA)* for approximating the solution of optimization problems [11]. They both use a parameterized quantum circuit and optimize the measurement results classically in multiple iterations by adjusting the input parameters of that circuit. Such algorithms are a subcategory of hybrid algorithms.

The implementation of (hybrid) quantum algorithms is enabled by *Software Development Kits (SDKs)* which are provided by IBM (Qiskit [1]), Rigetti (PyQuil [16]), and Microsoft (QDK [26]), Xanadu (PennyLane [5]), or CQL (t|ket) [39]). The SDKs are often available as libraries for existing classical programming languages. Important features are, e.g., the construction and manipulation of so-called *quantum circuits*, and establishing a connection to *quantum cloud services* for execution. Furthermore, these SDKs often not only provide quantum-specific functionality but classical functionality, too. One example are classical optimizers which are often used by variational quantum algorithms. Thus, hybrid quantum algorithms can be implemented via classical programming languages containing both quantum and classical parts in one single file.

2.2 Quantum Workflows

But even a hybrid quantum algorithm is by far not an application. For example, a quantum support vector machine intended to classify shopping carts requires this data in numerical format; thus, textual data needs to be transformed, embedded, feature reduced, etc., before the quantum computer can be used. The plethora of classical processing steps and the few quantum steps require orchestration [21, 48]. Trying to achieve this by using traditional programs, e.g., a Python script, suffers from different disadvantages, such as missing robustness or the possibility to interrupt its execution. In contrast, *workflow technologies* are an orchestration approach that has already been applied successfully in different application domains, such as *e-science* [25] or *business process management* [24]. Thereby, a *workflow model* defines a set of activities and their partial order, as well as the data flow between them [19]. The workflow model can be uploaded to a workflow engine for execution. Workflow engines provide an interruptible, scalable, persistent, and recoverable execution environment [24]. Furthermore, by utilizing a standardized workflow language, such as the *Business*

Process Model and Notation (BPMN) [31] or the *Business Process Execution Language (BPEL)* [29], the portability of the workflow models can be achieved. Thus, workflows should also be used to orchestrate the quantum and classical programs comprising a hybrid quantum algorithm [21, 48].

However, the modeling of quantum workflows is complex, time-consuming, and requires expertise from various fields [44]. To overcome this issue, the *Quantum Modeling Extension (QuantME)* [50] for imperative workflow languages, such as BPMN or BPEL, was introduced. QuantME provides explicit modeling constructs for various frequently occurring tasks when orchestrating hybrid quantum algorithms. Thus, it eases the modeling, increases the understandability of workflow models, and the reuse of implementations for these tasks. However, existing implementations of hybrid quantum algorithms are mostly script-based, through libraries of certain providers, such as *Qiskit* and *Cirq*, or platforms for sharing quantum software, such as the *PlanQK platform* [22]. This means the existing implementations have to be transformed into quantum workflows to benefit from the advantages of workflow technologies. However, due to the large amount and the steady increase of available implementations, a manual transformation is impractical which is why automation can make an important contribution here.

2.3 Deployment Models

As required by [21], our approach not only addresses the splitting of a script implementation into several parts but also their automated deployment. This involves provisioning a suitable execution environment and installing the necessary dependencies. Since a manual deployment is difficult and error-prone, different technologies to automate the deployment of applications (e.g., [9, 13]) have been developed [7]. These technologies process a deployment model which contains all the required information to instantiate the modeled application. Hence, deployment models ensure automation, traceability, and reusability.

In our approach, we rely on *declarative deployment models* [10], which describe the desired structure and state of the application that shall be deployed. In contrast, *imperative deployment models* specify a process to be executed, i.e., they define all tasks the deployment engine has to complete, their execution order, and the data flow between them. More precisely, we use the *Topology and Orchestration Specification for Cloud Applications (TOSCA)* [30]. In TOSCA, components are represented as *Node Templates* in a *Topology Template*, whereas each Node Template is of a certain *Node Type* defining its semantics. The relations between them are specified by *Relationship Templates*, which are also defined by *Relationship Types*, e.g., *connectTo*-relations and *hostedOn*-relations. Furthermore, Node Types can be annotated with *Requirement Definitions* and *Capability Definitions*. For example, a Requirement Type might be defined describing the need for a specific runtime environment. During modeling time, it is also possible for node templates to have open requirements that are not fulfilled by any other node template in the topology template. However, for successful deployment, all requirements must be satisfied by appropriate capabilities.

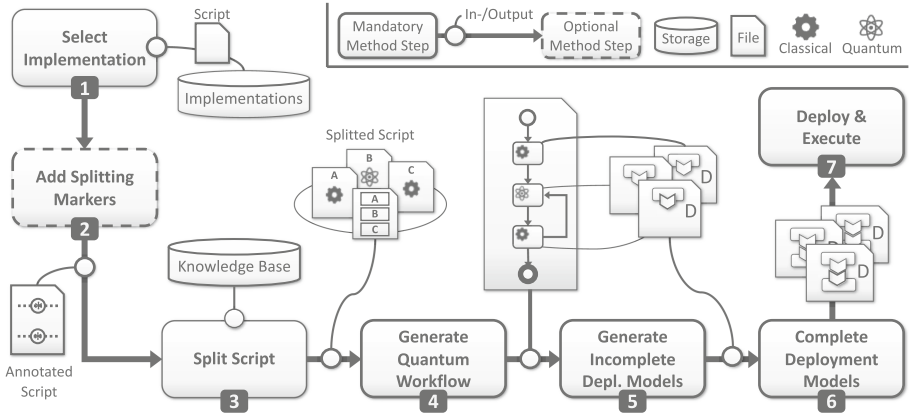


Fig. 1. Script Splitting and Workflow Generation Method

3 Script Splitting and Workflow Generation

In this section, we introduce the script splitting and workflow generation method. As depicted in Fig. 1, it consists of seven steps, which are explained in more detail in the following subsections: (i) select a script implementation, (ii) optionally annotate the script with custom splitting markers, (iii) split the annotated script into multiple parts using information from a knowledge base, (iv) generate a workflow model from the script parts retaining the control flow of the original script, (v) generate an incomplete deployment model for each script part and assign it to the respective task in the workflow, (vi) autocomplete the deployment models by adding appropriate nodes, and (vii) deploy the script parts, bind them to the workflow model, and execute it.

3.1 Select Implementation

The first step of the method is to select a quantum implementation. This step can be implemented in different ways, e.g., as a manual search on a collection of existing implementations or by an automated selection based on custom requirements. One prerequisite is that a quantum algorithm and a corresponding algorithm implementation exists for the given problem [48]. In addition, the selection step is also interested in whether benefits can be expected for an existing quantum-specific implementation. This can refer to different aspects, such as price, computing power, speed, or accuracy of the results [37].

In any case, databases and code repositories provide an important basis for this step. Application logic is often already divided into separate files, thus, several interrelated files might be selected in this step. Although the following steps can be applied to a set of files, for the sake of comprehensibility, we limit the explanation to single independent script files.

3.2 Add Splitting Markers

This optional step allows to influence the splitting of the selected quantum implementation by specific splitting markers that either prevent or explicitly force splitting at a certain point in the source code. For example, a force split can be useful if a user is aware of a computationally intensive part in the source code that should be separated, e.g., to deploy it separately and being able to parallelize it. In contrast, preventing a split can be helpful, e.g., to avoid distribution and the introduced communication between the script parts that comes with it. There are different ways to implement such splitting markers, e.g., via annotations, comments, or defined in a separate file pointing to specific code lines. Listing 1 shows how the splitting markers can be set in our prototype (see Sect. 4). The splitting markers are implemented as special line-comments that will be recognized by the script analysis in the next step. To force a split at a specific position in the source code, a user can insert a single-line comment between two existing lines of code. The comment in Listing 1 at line 2, thus, explicitly separates line 1 from lines 4 to 5. To prevent a split, a combination of two single-line comments can be used—one for starting a split-protected code block (line 3) and one for ending it (line 6). Note that force split markers must not occur within a split-protected code block.

```

1 a()
2 # ---force split---
3 # ---start prevent split---
4 b()
5 c()
6 # ---end prevent split---
```

Listing 1: Example usage of splitting markers

3.3 Split Script

This step splits the selected and annotated quantum implementation into multiple files so that quantum source code is separated from classical source code¹. As shown in Fig. 2, the splitting mechanism is implemented in four substeps:

First, the input script (possibly annotated with splitting markers of the last step) is analyzed to identify its quantum and classical parts. This is done by a syntactical analysis that goes through all lines in the source code and checks whether they use quantum or classical logic. The rules for this decision are defined in a knowledge base consisting of a list of quantum imports and a list of non-quantum imports. Within the former list, libraries (or parts of them) can be listed that are quantum-related. The purpose of the non-quantum imports list is to exclude (parts of) specific libraries. Listing 2 shows an excerpt of such

¹ Note that the resulting script parts are not necessarily pure quantum or pure classical due to user-defined splitting markers and a configurable splitting algorithm.

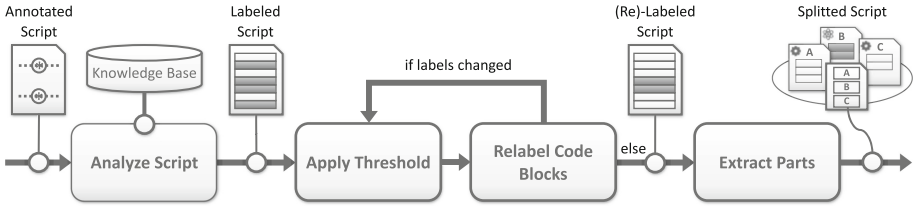


Fig. 2. Detailed view on the script splitting

a knowledge base in JSON. It defines that all imports from *qiskit* are labeled as “quantum” and all lines of code which use elements from that import are quantum as well. However, some libraries can also contain classical functionalities which can be specifically excluded using the non-quantum imports list. In the example in Listing 2 the sub-package *visualization* of *qiskit* is excluded resulting in it to be labeled as “classical”.

```

1 {
2   "quantum_imports": ["qiskit"],
3   "non-quantum_imports": ["qiskit.visualization"]
4 }

```

Listing 2: Excerpt of the knowledge base in JSON

Without any restriction, the splitting algorithm would also extract single lines in the source code into separate parts. However, such a fine-grained split is not always intended. Therefore, the second sub-step relabels the script based on a configurable threshold. This threshold specifies the number of classical lines of code that are allowed before or after a quantum part without splitting it off from the quantum part. If there are multiple quantum parts, the threshold is also applied to the classical lines of code in between them.

The third sub-step analyzes all code blocks (i.e., if-else-blocks, while-loops, etc.) in the source code recursively and relabels them. The goal is that only the control flow between hybrid code blocks becomes visible in the workflow model. Thus, each part of a hybrid code block is represented by a task and the loops and conditions are represented by appropriate sequence flows and gateways. In contrast, code blocks that are pure quantum or pure classical are mapped to single tasks. As can be seen in Fig. 2, if any code blocks have been relabeled, we apply the threshold again and repeat both steps until the relabeling converges.

Finally, the relabeled script is splitted into multiple parts based on the splitting labels (see Fig. 2). For this, groups of equal labels are identified and extracted to functions in separate files. In the original script, the code parts are replaced by function calls to the separate file, i.e., the script is still executable. For the extracted code parts to be self-contained and executable on their own, all referenced variables need to be passed to the functions properly. Thus, the extracted

parts are analyzed to identify all external variables needed. These variables are added as arguments to the created functions. Analogously, return variables are computed since the results of one code part might be needed in another one. In the later steps, we wrap the code parts by services, and passing variables between the distributed code parts becomes more difficult. While primitive variables might be handed over easily, more complex variable types require appropriate serialization and deserialization by the service wrappers. This can be achieved, e.g., by using templates for different data types.

3.4 Generate Quantum Workflow

After splitting the script into individual parts, a service task is first created in the workflow for each part. To determine the execution order of the service tasks, they must be connected with corresponding sequence flows in the workflow model. The execution order of the tasks in the workflow model corresponds to the position of the parts in the original script, which also applies to the data flow, where the output of the one part can be used in the subsequent parts. For this reason, additional parameter and result lists are created for each extracted code part. Once a computation is complete, the result is returned and passed on to the next task using data flow.

A quantum implementation typically includes not only sequential expressions, but also loops. As mentioned in the previous step, hybrid loops will be represented in the workflow model, which can be done using exclusive gateways and conditional sequence flows. The workflow model must therefore be supplemented by suitable elements. In addition, loops typically have loop variables that are incremented and linked to a loop condition. These have already been analyzed in the previous step and are now initialized by an additional task. The prototype (see Sect. 4) which contains the details for the workflow generation is open-source and available on Github [41, 42].

3.5 Generate Incomplete Deployment Models

So far, the script was split into several parts, and a workflow model that orchestrates these parts was generated. The individual tasks in the workflow model are service tasks, each invoking a service corresponding to a script part at a certain endpoint. However, these services are not running yet and must be deployed before the workflow execution. In order to deploy the script parts as services, we use a service wrapper and inject the script part into it. The service wrapper and its integration can be implemented differently. Either asynchronously, e.g., via topics where the service queries these topics, or synchronously, e.g., with the service defining specific endpoints for the workflow engine to call.

For an automated deployment of the service wrapper and script parts, there are various deployment technologies that can be used (see Sect. 2.3). However, most are limited in their interoperability and portability. TOSCA [30] addresses this by providing a standardized specification for deployment models, which is used in this step. Thus, we create a TOSCA-based deployment model containing

node templates representing the polling agent and service wrapper. We also add a requirement to these node templates stating how they can be hosted. These incomplete deployment models then serve as input for the next step. At this point, the deployment model is still independent of a concrete infrastructure or target environment, i.e., it is highly portable.

3.6 Complete Deployment Models

In the next step, the incomplete deployment models are completed to obtain executable deployment models. Based on the SePaDe approach [17], our approach uses a repository of pre-existing deployment models. This repository is scanned for service templates that contain appropriate node types for the script parts. The script parts are then injected into the deployment models as artifacts of the respective nodes leading to complete deployment models. Instead of replacing the incomplete deployment models with pre-existing deployment models, the incomplete deployment models could also be supplemented by injecting deployment model fragments [40]. Hereby, the definition of open requirements in TOSCA can be used. These open requirements are satisfied by deployment model fragments that offer corresponding capabilities [14, 36].

3.7 Deploy and Execute

Currently, the tasks in the workflow model are connected to the deployment models, more precisely, the tasks link to *Cloud Service Archive (CSAR)* files. As defined in TOSCA [30], CSARs are self-contained zip files with a well-defined structure and content. For example, they contain the deployment model as well as all needed artifacts (i.e., the script parts, service wrapper, and other artifacts such as Dockerfiles). Therefore, we deploy an instance of each generated deployment model in the last step. These instances can be automatically bound with the workflow [49]. This involves removing links to the CSAR for each task and adding the configuration to invoke the deployed service. The result is an executable workflow, and since it orchestrates the script parts in the execution order of the original implementation, the overall logic is equivalent.

4 Prototype

Our system extends the *MODULO framework* [47] to model, transform, deploy, and execute quantum workflows with additional components. Thereby, the overall system architecture is shown in Fig. 3. In the *QuantME Transformation Framework* [42], we extended the user interface of the *QuantME Modeler* to upload script-files for processing. The script splitting mechanism is implemented in the *Script Handler* component which runs separately providing appropriate endpoints. The *Script Handler* [41] is integrated with the *QuantMe Transformation Framework* by the *Script Handler Connector* which handles the interaction with the Script Handler, i.e., it handles the (long-running) requests and

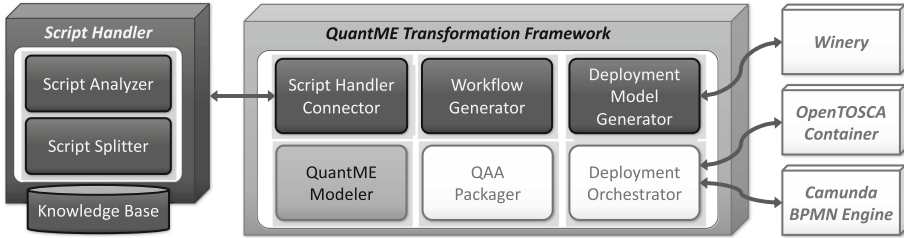


Fig. 3. Extended system architecture of the MODULO framework (light: existing components, medium gray: extended components, dark: new components)

responses. The Script Handler consists of two main components: the *Script Analyzer* which find suitable splitting positions depending on the used programming language and knowledge base, and the *Script Splitter* which extracts different parts into separate file and identifies the correct input/output variables. Both components use the *Knowledge Base* to provide their functionality. Another added component is the *Workflow Generator* which takes results from the Script Handler to generate a corresponding workflow model. The *Deployment Model Generator* generates deployment models for each extracted script part using the topology completion functionality of *Winery* [18]. *Winery* is thereby used for deployment model generation, as well as repository for deployment models related to services of a workflow. Finally, the MODULO framework contains two unmodified components. The *Deployment Orchestrator* performs the deployment of the service instances for a workflow using the *OpenTOSCA Container* [6], binds the service instance with the workflow, and uploads the workflow to the *Camunda BPMN Engine*. The *QAA Packager* packages all required information for a quantum workflow in a so-called *Quantum Application Archive (QAA)* [21, 49].

5 Case Study: Shor's Algorithm

For validating our approach, we consider the quantum algorithm of Shor [38] for factoring integers. The goal is to find a prime factor z that is a proper divisor of N . Applied iteratively, Shor's algorithm can thus be used to compute a set of prime factors whose product is equal to N . For the use case study, we use an exemplary implementation of Shor's algorithm which is a Python script using the Qiskit SDK. The implementation is open source and provided in our GitHub repository [43]. A user can directly execute the algorithm implementation in Python. However, the script combines both quantum and classical tasks and reusing (parts of) the code in other projects is difficult and requires tedious re-implementation. Therefore, we want to apply our method to this script to split it into its quantum and classical parts on the one hand and generate a workflow that orchestrates the script parts on the other hand.

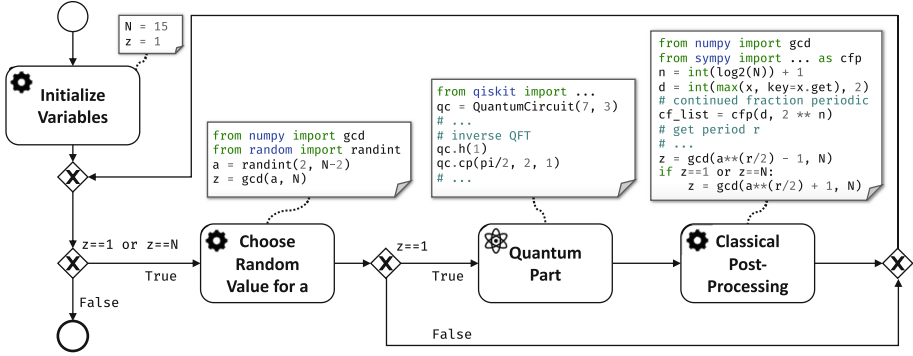


Fig. 4. Generated workflow for an implementation of Shor’s algorithm

Figure 4 shows the result of the workflow generation. For presentation and comprehensibility, we have already manually named the tasks in a meaningful way and included the corresponding script parts in the figure. The workflow consists of three classical tasks, one quantum task, and several exclusive gateways. Using the workflow model, the main building blocks of Shor’s algorithm can now be easily identified: The first task initializes variables needed for the subsequent tasks. Afterwards, the workflow contains two gateways which form a loop containing all remaining tasks. These tasks are repeated until a non-trivial factor z of N has been found, i.e., if $z \neq 1$ and $z \neq N$. In this case, the algorithm is completed and the workflow terminates. If the loop condition still holds and no valid factor has been found, we first choose a random integer value a between 2 and $N - 2$, compute the *greatest common divisor* (*gcd*) of a and N , and assign the result to z . If z is not equal to 1, we have found a non-trivial divisor of N and skip the remaining tasks. Since the loop condition is no longer satisfied, the workflow terminates. If a , on the other hand, does not divide N , then the key part of the algorithm is executed, which consists of two tasks. The first task represents the *Quantum Part* of Shor’s algorithm; it constructs a quantum circuit based on a and applies an inverse *quantum fourier transform* (*QFT*) to it. The result of this task is then used in the rightmost task in Fig. 4. It performs the continued fractions algorithm for $\frac{x}{2^n}$ and uses the resulting continued fraction sequence to find the period r (see [3]) of $a^x \bmod n$. Finally, z is computed by $\gcd(a^{\frac{r}{2}} - 1, N)$ or $\gcd(a^{\frac{r}{2}} + 1, N)$ if the former does not yield a valid result.

6 Discussion

In the following, we discuss limitations of our method regarding the detection of suitable splits, the handling of input and output parameters, as well as the peculiarities of scripts implementing variational quantum algorithms.

The presented method relies on the availability of an up-to-date knowledge base comprising rules to detect suitable splits. These rules are based on libraries

used by quantum implementations. Since quantum computing is a very active area, new libraries or SDKs appear regularly and the knowledge base must be updated periodically by experts. Automated investigation of underlying libraries, e.g., based on heuristics, is also conceivable. We want to point out that our method to split script-based quantum implementations can also be applied to do a split in other application areas, e.g., to separate machine learning parts from other parts. To achieve correct analysis and splitting, the knowledge base needs to be adapted.

The method extracts code blocks into separate tasks in the workflow model whenever both quantum and classical parts are contained. Thus, the optimization loop of variational algorithms would be modeled via control flows and XOR-gates in the workflow model. However, the distributed execution of variational algorithms often suffers from performance issues due to the time waiting in queues. Therefore, quantum service providers have recently introduced hybrid runtimes to speed up execution. To make use of this, the workflow models generated by our method, can automatically converted into hybrid implementations [46]. Furthermore, since loops and if-else-blocks in workflows are explicitly represented by sequence flows and gateways, nested loops with hybrid code blocks can lead to results with many branches, hence, they can be difficult to read. One way to address this is to increase the splitting threshold as described in Sect. 3.3, reducing the number of tasks in the workflow model and, consequently, the number of branches.

It can be argued that providing all script parts as services, at least for small script parts, is inefficient, e.g., due to the additional data transfer between the workflow tasks. For example, in the case study, one small task was generated which only initialize variables that will be needed it subsequent tasks. These variables are serialized by the external service task and sent over the network back to the workflow engine. Instead, such tasks could also be realized as script tasks that run directly in the workflow engine. However, this would raise other challenges. For example, the workflow engine does not necessarily understand the programming language of the original script, i.e., the script parts would have to be automatically translated into a programming language supported by the workflow engine. Furthermore, it can be argued that the benefits of workflows do not always justify the increased effort in development and execution, i.e., there are projects where implementing and executing simple quantum scripts might be sufficient.

7 Related Work

The recovery of business knowledge from existing software is a challenge frequently addressed in literature. For example, Pérez-Castillo et al. [32] use reverse engineering techniques to extract models using the *Knowledge Discovery Meta-model* from existing software and further use a pattern-based approach transforming these models to BPMN. Cai et al. [8] propose an approach for recovering business processes based on static and dynamic source code analysis. Zou et

al. [53] introduce a framework which utilizes static code analysis combined with heuristic rules to map source code entities to business process entities.

Also, the execution of quantum algorithms using workflows is covered by different research works and commercial products. For example, Zapata *Orchestra* [51] provides a YAML-based workflow language to define workflows including quantum and classical computations. Thus, Orchestra is not based on widely used standards, such as BPMN [31], and does not seem to be based on any of the established workflow systems. Although Orchestra incorporates a software platform automatically conducting these workflows, they do not allow the generation of workflows by splitting a given script. Similarly, *Covalent* [2] is a Python-based so-called workflow orchestration platform which uses annotations in the source code to encapsulate functions as tasks in the workflow. In contrast to the workflow language we use in our approach, these approaches do, e.g., not allow human tasks and the execution cannot be paused or rolled back.

Separating source code into individual, maintainable modules has always been a good practice in application development. Automatic approaches to modularization also already exist. Mitchell et al. [27] generate a graph from source code whereby all classes and function calls are represented. They further discuss a clustering approach, which applied on the graph results in modularization of the source code. However, to the best of our knowledge, there are no publications combining the automatic splitting of script-based implementations (into its quantum and classical parts) with the generation of workflow models.

8 Summary and Future Work

Workflow technologies are an orchestration approach that provides robustness and reusability. However, many quantum implementations are often available as scripts, e.g., as programs written in Python. Thereby, the orchestration of the control flow between classical and quantum parts is implemented within the source code. A manual conversion of these script implementations to workflow models is time-consuming and complicated. Our approach can be used to automatically generate a workflow model from given script-based quantum implementations. The quantum and classical parts are represented as tasks and the control flow is modeled explicitly. This allows to control the execution of single parts of an algorithm or reuse them in other scenarios. We validated our approach and prototype by a use case study showing that the building blocks of Shor's Algorithm can easily be detected.

In future work, we plan to evaluate the practical applicability for use cases of different complexity. We also want to identify criteria for deciding which parts of an application to implement classically or as quantum implementations.

Acknowledgement. This work was partially funded by the BMWK projects *PlanQK* (01MK20005N) and the project *SEQUOIA* funded by the Baden-Württemberg Ministry of Economy, Labour and Housing.

References

1. Abraham, H., et al.: Qiskit: An Open-source Framework for Quantum Computing (2019). <https://doi.org/10.5281/zenodo.2562110>
2. Agnostiq: Covalent (2022). <https://agnostiq.ai/covalent/>
3. Barzen, J., Leymann, F.: Continued fractions and probability estimations in Shor's algorithm: a detailed and self-contained treatise. [arXiv:2007.07047](https://arxiv.org/abs/2007.07047) (2022)
4. Barzen, J., Leymann, F., Falkenthal, M., Vietz, D., Weder, B., Wild, K.: Relevance of near-term quantum computing in the cloud: a humanities perspective. *Cloud Comput. Serv. Sci.* **1399**, 25–58 (2021)
5. Bergholm, V., et al.: PennyLane: automatic differentiation of hybrid quantum-classical computations (2020). [arXiv preprint arXiv:1811.04968](https://arxiv.org/abs/1811.04968)
6. Binz, T., et al.: OpenTOSCA – a runtime for TOSCA-based cloud applications. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) *ICSOC 2013*. LNCS, vol. 8274, pp. 692–695. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45005-1_62
7. Breitenbücher, U., Binz, T., Képes, K., Kopp, O., Leymann, F., Wettinger, J.: Combining declarative and imperative cloud application provisioning based on TOSCA. In: *International Conference on Cloud Engineering (IC2E 2014)*, pp. 87–96. IEEE (2014)
8. Cai, Z., Yang, X., Wang, X.: Business process recovery for system maintenance - an empirical approach. In: *2009 IEEE International Conference on Software Maintenance*, pp. 399–402 (2009)
9. CNCF: Kubernetes (2021). <https://kubernetes.io>
10. Endres, C., Breitenbücher, U., Falkenthal, M., Kopp, O., Leymann, F., Wettinger, J.: Declarative vs. imperative: two modeling patterns for the automated deployment of applications. In: *Proceedings of the 9th International Conference on Pervasive Patterns and Applications (PATTERNS 2017)*, pp. 22–27. Xpert Publishing Services (2017)
11. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. [arXiv:1411.4028](https://arxiv.org/abs/1411.4028) (2014)
12. Gabor, T., et al.: The holy grail of quantum artificial intelligence: major challenges in accelerating the machine learning pipeline. [arXiv:2004.14035](https://arxiv.org/abs/2004.14035) (2020)
13. HashiCorp: Terraform (2021). <https://www.terraform.io>
14. Hirmer, P., Breitenbücher, U., Binz, T., Leymann, F.: Automatic topology completion of TOSCA-based cloud applications. In: *Proceedings des CloudCycle14 Workshops auf der 44. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*. LNI, vol. 232, pp. 247–258. Gesellschaft für Informatik e.V. (GI) (2014)
15. Johansson, M.P., Krishnasamy, E., Meyer, N., Piechurski, C.: *Quantum Computing - A European Perspective* (2021). <https://doi.org/10.5281/zenodo.5547408>
16. Karalekas, P.J., et al.: PyQuil: Quantum programming in Python (2020). <https://doi.org/10.5281/zenodo.3631770>
17. Képes, K., Breitenbücher, U., Leymann, F.: The SePaDe system: packaging entire XaaS layers for automatically deploying and managing applications. In: *Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER 2017)*, pp. 626–635. SciTePress (2017)
18. Kopp, O., Binz, T., Breitenbücher, U., Leymann, F.: Winery – a modeling tool for TOSCA-based cloud applications. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) *ICSOC 2013*. LNCS, vol. 8274, pp. 700–704. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45005-1_64

19. Leymann, F., Altenhuber, W.: Managing business processes as an information resource. *IBM Syst. J.* **33**(2), 326–348 (1994)
20. Leymann, F., Barzen, J.: The bitter truth about gate-based quantum algorithms in the NISQ era. *Quantum Sci. Technol.* 1–28 (2020)
21. Leymann, F., Barzen, J.: Hybrid quantum applications need two orchestrations in superposition: a software architecture perspective. [arXiv:2103.04320](https://arxiv.org/abs/2103.04320) (2021)
22. Leymann, F., Barzen, J., Falkenthal, M.: Towards a platform for sharing quantum software. In: *Proceedings of the 13th Advanced Summer School on Service Oriented Computing (2019)*, pp. 70–74. IBM Technical Report (RC25685), IBM Research Division (2019)
23. Leymann, F., Barzen, J., Falkenthal, M., Vietz, D., Weder, B., Wild, K.: Quantum in the cloud: application potentials and research opportunities. In: *Proceedings of the 10th International Conference on Cloud Computing and Services Science (CLOSER 2020)*, pp. 9–24. SciTePress (2020)
24. Leymann, F., Roller, D.: *Production Workflow: Concepts and Techniques*. Prentice Hall PTR (2000)
25. Liu, J., Pacitti, E., Valduriez, P., Mattoso, M.: A survey of data-intensive scientific workflow management. *J. Grid Comput.* **13**(4), 457–493 (2015). <https://doi.org/10.1007/s10723-015-9329-8>
26. Microsoft: Quantum Development Kit (2020). <https://microsoft.com/en-us/quantum/development-kit>
27. Mitchell, B., Mancoridis, S.: On the automatic modularization of software systems using the bunch tool. *IEEE Trans. Softw. Eng.* **32**(3), 193–208 (2006)
28. National Academies of Sciences, Engineering, and Medicine: *Quantum Computing: Progress and Prospects*. National Academies Press (2019)
29. OASIS: *Web Services Business Process Execution Language (WS-BPEL) Version 2.0*. Organization for the Advancement of Structured Information Standards (OASIS) (2007)
30. OASIS: *Topology and Orchestration Specification for Cloud Applications (TOSCA) Version 1.0*. Organization for the Advancement of Structured Information Standards (OASIS) (2013)
31. OMG: *Business Process Model and Notation (BPMN) Version 2.0*. Object Management Group (OMG) (2011)
32. Pérez-Castillo, R., García-Rodríguez de Guzmán, I., Piattini, M.: Implementing business process recovery patterns through QVT transformations. In: Tratt, L., Gogolla, M. (eds.) *ICMT 2010*. LNCS, vol. 6142, pp. 168–183. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13688-7_12
33. Peruzzo, A., et al.: A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**(1), 4213 (2014)
34. Piattini, M., et al.: The Talavera manifesto for quantum software engineering and programming. In: *Proceedings of the 1st International Workshop on the Quantum Software Engineering & Programming (QANSWER 2020)*, pp. 1–5. CEUR Workshop Proceedings (2020)
35. Preskill, J.: Quantum Computing in the NISQ era and beyond. *Quantum* **2** (2018)
36. Saatkamp, K., Breitenbücher, U., Kopp, O., Leymann, F.: Topology splitting and matching for multi-cloud deployments. In: *Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER 2017)*, pp. 247–258. SciTePress (2017)
37. Salm, M., Barzen, J., Breitenbücher, U., Leymann, F., Weder, B., Wild, K.: The NISQ analyzer: automating the selection of quantum computers for quantum

- algorithms. In: Dustdar, S. (ed.) SummerSOC 2020. CCIS, vol. 1310, pp. 66–85. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64846-6_5
38. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26**(5), 1484–1509 (1997)
 39. Sivarajah, S., Dilkes, S., Cowtan, A., Simmons, W., Edgington, A., Duncan, R.: `t|ket`: a retargetable compiler for NISQ devices. *Quantum Sci. Technol.* **6**(1), 014003 (2020)
 40. Soldani, J., Binz, T., Breitenbücher, U., Leymann, F., Brogi, A.: ToscaMart: a method for adapting and reusing cloud applications. *J. Syst. Softw.* **113**, 395–406 (2016)
 41. University of Stuttgart: qscript splitter (2022). <https://github.com/UST-QuAntiL/qscript-splitter>
 42. University of Stuttgart: QuantME Modeling and Transformation Framework (2022). <https://github.com/UST-QuAntiL/QuantME-TransformationFramework>
 43. University of Stuttgart: QuantME Use Cases (2022). <https://github.com/UST-QuAntiL/QuantME-UseCases>
 44. Vietz, D., Barzen, J., Leymann, F., Weder, B., Yussupov, V.: An exploratory study on the challenges of engineering quantum applications in the cloud. In: Proceedings of the 2nd Quantum Software Engineering and Technology Workshop (Q-SET 2021), pp. 1–12. CEUR Workshop Proceedings (2021)
 45. Vietz, D., Barzen, J., Leymann, F., Wild, K.: On decision support for quantum application developers: categorization, comparison, and analysis of existing technologies. In: Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloat, P.M.A. (eds.) ICCS 2021. LNCS, vol. 12747, pp. 127–141. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77980-1_10
 46. Weder, B., Barzen, J., Beisel, M., Leymann, F.: Analysis and rewrite of quantum workflows: improving the execution of hybrid quantum algorithms. In: Proceedings of the 12th International Conference on Cloud Computing and Services Science (CLOSER 2022), pp. 38–50. SciTePress (2022)
 47. Weder, B., Barzen, J., Leymann, F.: MODULO: modeling, transformation, and deployment of quantum workflows. In: Proceedings of the 25th IEEE International Enterprise Distributed Object Computing Workshop (EDOCW 2021), pp. 341–344. IEEE Computer Society (2021)
 48. Weder, B., Barzen, J., Leymann, F., Vietz, D.: Quantum software development lifecycle. [arXiv:2106.09323](https://arxiv.org/abs/2106.09323) (2022)
 49. Weder, B., Breitenbücher, U., Képes, K., Leymann, F., Zimmermann, M.: Deployable self-contained workflow models. In: Brogi, A., Zimmermann, W., Kritikos, K. (eds.) ESOC 2020. LNCS, vol. 12054, pp. 85–96. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44769-4_7
 50. Weder, B., Breitenbücher, U., Leymann, F., Wild, K.: Integrating quantum computing into workflow modeling and execution. In: Proceedings of the 13th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2020), pp. 279–291. IEEE Computer Society (2020)
 51. Zapata: Orquestra (2022). <https://www.zapatacomputing.com/orquestra>
 52. Zhao, J.: Quantum software engineering: landscapes and horizons. [arXiv:2007.07047](https://arxiv.org/abs/2007.07047) (2020)
 53. Zou, Y., Lau, T.C., Kontogiannis, K., Tong, T., McKegney, R.: Model-driven business process recovery. In: 11th Working Conference on Reverse Engineering, pp. 224–233 (2004)

Author Index

- Adach, Malina 36
Ali, Syed Juned 57
- Barzen, Johanna 255
Beerepoot, Iris 165
Bellan, Patrizio 182
Bork, Dominik 57
- Cappelli, Claudia 91
Chesani, Federico 217
Costa, Marcus Vinicius 91
- da Silva, José Lutiano Costa 91
de Kinderen, Sybren 19
Di Francescomarino, Chiara 217
Dragoni, Mauro 182
Drews, Paul 74
- Gallik, Florian 111
Ghidini, Chiara 182, 217
Glaser, Philipp-Lorenz 57
Grundler, Giulia 217
- Hacks, Simon 19
Hänninen, Kaj 36
Huang, Tsung-Hao 200
- Kaczmarek-Heß, Monika 19
Kanin, Oleg 74
Kirikkayis, Yusuf 111
Koorn, Jelmer J. 165
Kuchen, Herbert 237
- Leymann, Frank 255
Li, Chiao-Yun 128
Lomidze, Giorgi 128
- Loreti, Daniela 217
Lundqvist, Kristina 36
- Maggi, Fabrizio Maria 217
Mangat, Amolkirat Singh 145
Mello, Paola 217
Milosevic, Zoran 3
Montali, Marco 217
- Nunes, Vanessa 91
- Pyefinch, Frank 3
- Reichert, Manfred 111
Reijers, Hajo A. 165
Rinderle-Ma, Stefanie 145
- Sallinger, Emanuel 57
Schneid, Konrad 237
Schuster, Daniel 128
Sommer, Felipe 91
- Tessarar, Sergio 217
Thöne, Sebastian 237
Töpel, Daniel 19
- van der Aalst, Wil M. P. 200
van Zelst, Sebastiaan J. 128
Vietz, Daniel 255
- Weber, Barbara 165
Weder, Benjamin 255
- Xisto, Adriana 91
- Zerbato, Francesca 165